Lab 2
Scan Chain Insertion and ATPG Using DFTADVISOR and FASTSCAN

Pro: Chia-Tso Chao
TA: Szu-Pang Mu
2016/5/23
Outline

- Introduction
- Dftadvisor
- Fastscan
- Mix Flow
- Lab
Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mix Flow
- Lab
Introduction

- This lab focuses on ATPG results from different tools: Mentor Graphic and Synopsys.
- Dftadvisor is used to insert scan chain (basically replacing FF with scan FF).
- Fastscan is used to perform ATPG and fault simulation.
Insert Scan and ATPG Flow

**Tool Flow**

- **RTL Coding**
  - RTL Design

- **SYNTHESIS**
  - Scan Insertion
    - DFTAdvisor
  - Gate Level Netlist
  - Scan Inserted Netlist

- **ATPG**
  - FastScan
  - Test Patterns

---

**Non-scan**

**Scan**

**DIAGRAM**
Input/Output Files

- Simulation Library
  adk.atpg

- Dftadvisor
- Scanned Circuit
  design_scan.v

- Circuit Netlist
  design.v

- Scan Chain Information
  design_scan.dofile
  design_scan.testproc

- Fastscan

- ATE Vectors
- Simulation Testbenches
- Fault Reports
Invoke DFTADVISOR

- Read in verilog source file and assign library file.
  
  ```
  $ cd lab2
  $ dftadvisor pre_norm_noscan.v -verilog -lib l90sprvt.atpg -nogui
  ```
Specify Clock

Clocks are primary input signals that asynchronously change the state of sequential logic elements.

- `SETUP> add clock 0 clk`

  specify which state cannot affect output
Setup Test Logic Configuration

- Set scan methodology.
  - Mux_scan : mux-DFF
  - Lssd : level sensitive DFF
  - Clocked_scan : clocked-signal
    - SETUP> set scan type m

- Test logic options -- make clock lines controllable to get a scannable design.
  - SETUP> set test logic -clock on -reset on

Non-scannable

Scannable
DRC

- Entering DRC mode.
  - SETUP> set system mode dft

- Perform DRC.
  - DFT> run

- Report some information.
  - DFT> report statistics
DRC Rules

- General Rules
- Procedure Rules
- Scan Chain Trace Rules
- Scan Cell Data Rules
- Clock Rules
- Ram Rules
- BIST Rules
- EDT Rules
- Timing Rules
Insert Scan Chain and View Report

☐ Set # of scan-chains to insert and do so.
   - DFT> insert test logic -number 10

☐ Report scan-chain information.
   - DFT> report scan chain
   - DFT> report test logic
Output Scanned Design for ATPG

- Write out files and exit DFTADVISOR
  - `DFT> write netlist pre_norm_scan.v -verilog -replace`
  - `DFT> write atpg setup pre_norm_scan -replace`
  - `DFT> exit`

`.dofile : setup information
.testproc : procedure file`
Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mix Flow
- Lab
Invoke Fastscan

- specify scanned verilog file and library file.

$ fastscan pre_norm_scan.v -verilog -lib l90sprvt.atpg -nogui
Read Setup Information

- Read setup information from Dftadvisor.
  - SETUP> dofile pre_norm_scan.dofile

- Entering atpg mode.
  - SETUP> set system mode atpg

- Setup fault type: stuck, iddq, toggle, transition.
  - ATPG> set fault type stuck
Generate Patterns

- Uses '-auto' option to allow Fastscan to analyze design and suggest the best settings possible to generate the most compact patterns with the highest coverage with the lowest time.

  ATPG> create patterns -auto
Generate Patterns

- Without ‘auto’ option, you can specify your own configurations using these commands:

  - set atpg limits -Cpu_seconds [integer] -Test_coverage [real] -Pattern_count [integer]
  - set atpg compression on -Abort_limit [integer]
  - identify redundant faults

  ATPG> create patterns
View Report

- Report simulation result and faults.
  - ATPG> report statistics
  - ATPG> report faults -all

<table>
<thead>
<tr>
<th>Fault code</th>
<th>Fault site</th>
<th>Fault value: Either 0 (for stuck-at-0) or 1 (for stuck-at-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>/I$7$/OUT</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>/I$7$/IN</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>/I$1$/en</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>/I$7$/OUT</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>/I$7$/IN</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>/I$1$/en</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>/I$4$/l1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>/I$20$/en</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>/I$20$/en</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>/I$2$/en</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>/I$2$/en</td>
<td></td>
</tr>
</tbody>
</table>
Simulation performed for #gates = 27550  #faults = 110115

system mode = ATPG  pattern source = internal patterns

------------------------------------------------------------------------
#patterns test #faults #faults # eff. # test process RE/AU/abort
simulated coverage in list detected patterns patterns CPU time
deterministic ATPG invoked with comb/seq abort limit = 300/100

<table>
<thead>
<tr>
<th>#</th>
<th>#patterns</th>
<th>test</th>
<th>#faults</th>
<th>#faults</th>
<th>%</th>
<th>#test</th>
<th>process</th>
<th>RE/AU/abort</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>82.90%</td>
<td>20861</td>
<td>89030</td>
<td>32</td>
<td>32</td>
<td>0.10 sec</td>
<td>224/0/0</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>91.30%</td>
<td>10501</td>
<td>9161</td>
<td>32</td>
<td>64</td>
<td>0.26 sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>95.67%</td>
<td>5187</td>
<td>4395</td>
<td>32</td>
<td>96</td>
<td>0.41 sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>98.43%</td>
<td>1887</td>
<td>2450</td>
<td>32</td>
<td>128</td>
<td>0.60 sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>99.22%</td>
<td>954</td>
<td>516</td>
<td>10</td>
<td>138</td>
<td>0.68 sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>99.69%</td>
<td>357</td>
<td>597</td>
<td>32</td>
<td>170</td>
<td>0.69 sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>99.98%</td>
<td>24</td>
<td>333</td>
<td>32</td>
<td>202</td>
<td>0.70 sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>100.00%</td>
<td>0</td>
<td>24</td>
<td>3</td>
<td>205</td>
<td>0.70 sec</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Result

### Statistics report

<table>
<thead>
<tr>
<th>Fault Classes</th>
<th>#faults (total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU (full)</td>
<td>6078</td>
</tr>
<tr>
<td>UO (unobserved)</td>
<td>9 (0.15%)</td>
</tr>
<tr>
<td>DS (det_simulation)</td>
<td>5393 (88.73%)</td>
</tr>
<tr>
<td>DI (det_implication)</td>
<td>540 (8.88%)</td>
</tr>
<tr>
<td>UU (unused)</td>
<td>16 (0.26%)</td>
</tr>
<tr>
<td>RE (redundant)</td>
<td>118 (1.94%)</td>
</tr>
<tr>
<td>AU (atpg_untestable)</td>
<td>2 (0.03%)</td>
</tr>
</tbody>
</table>

### Coverage

- test_coverage: 99.81%
- fault_coverage: 97.61%
- atpg_effectiveness: 99.85%

### Additional metrics

- #test_patterns: 180
- #simulated_patterns: 224
- CPU_time (secs): 3.7
Fault Types (1/3)

- **AU**: Atpg_untestable
  - Due to pin constraint or insufficient sequential depth cause testable faults become atpg_untestable
- **FU**: Full
- **TE**: Testable
- **DT**: Detected
Fault Types(2/3)

- **UT**: Untestable
  - Faults which no pattern can exist to either detect or possible-detect them, such as unused pins

- **UD**: Undetected
  - Faults cannot be proven untestable or ATPG_untestable

- **RE**: Redundant
Fault Types (3/3)

- **UU**: usused
  - All faults unconnected to any circuit observation point
- **BL**: blocked
  - Faults which logic blocks all paths to an observation point
- **TI**: tied
  - Point of the fault value is always same (and-gate with complementary inputs)
Test Coverage Formula Comparison

- **Tmax**
  \[
  \text{test\_coverage} = \frac{DT + (PT \times \text{posdet\_credit})}{all\_faults - (UD + AU \times \text{au\_credit})} \times 100
  \]
  
  - possible detected
  - default 50%
  - default 0

- **Fastscan**
  \[
  \text{test\_coverage} = \frac{DT + (PT \times \text{posdet\_credit})}{\text{testable}} \times 100
  \]
  \[
  \text{fault\_coverage} = \frac{DT + (PT \times \text{posdet\_credit})}{\text{full}} \times 100
  \]
  \[
  \text{ATPG\_effectiveness} = \frac{DT + UT + AU + PU + (PT \times \text{posdet\_credit})}{\text{full}} \times 100
  \]

Testable = DT + PT + AU + UD

Untestable = UU + TI + BL + RE
Save Patterns

- Save patterns just generated. The response of ATE saved in s38584_seq_ate.pat
- Various format including binwgl, ctl2005, stil2005, stil999, verilog, vhdl, wgl, zycad, tstl2, utic.
  - ATPG> save patterns pre_norm_scan.pat -verilog -proc -replace
  - ATPG> save patterns pre_norm_scan_tstl2.pat -TSTL2 -rep
  - ATPG> exit

Toshiba Standard Tester Interface Language 2
Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mix Flow
- Lab
Testing Flow

- Synopsys Design Compiler is better in mapping from RTL code to gate-level circuit.
- Some cases in industry field uses Design Compiler to synthesis RTL code and uses Fastscan to perform ATPG and fault simulation.
Input/Output Files

- Simulation Library
  tsm18.v

- DC

- Scanned Circuit
  design_scan.v

- Scan Chain Information
  design_scan.stil

- Fastscan
  stil2mgc

- Circuit Netlist
  design.v

- libcomp

- Simulation Testbenches
- ATE Vectors
- Fault Reports
Input Files Required from Design Compiler’s Flow

- Library file need to be converted.
  - From .v to .atpg
- The detail information about scanned circuit need to be converted.
  - From .stil to .dofile
- Scanned code.
Convert STIL File

- Using command 'stil2mgc' to convert STIL file into dofile and test procedure file for Fastscan.
  - $ stil2mgc pre_norm_scan.stil
    - It will generate pre_norm_scan.stil.do and pre_norm_scan.stil.proc
Performing ATPG using FASTSCAN

- Read scanned circuit and library from design compiler to perform ATPG.
  - `$ fastscan pre_norm_scan.v -verilog -lib l90sprvt.atpg -nogui`
  - SETUP> dofile pre_norm_scan.stil.do
  - SETUP> set sys mode atpg
  - ATPG> create patterns -auto
  - ATPG> report statistics
Outline

- Introduction
- DFTADVISOR
- FASTSCAN
- Mix Flow
- Lab
Lab Goal

- Compare test coverage and # of patterns and run time during ATPG using methods in DC + TMAX and DFTA + FS and DC + FS.

- You need to run in circuits pre_norm.v, and show the results like next slide.
## Result

<table>
<thead>
<tr>
<th></th>
<th>Total Faults</th>
<th>Test Coverage</th>
<th># of Patterns</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC + TMAX</td>
<td>71298</td>
<td>100%</td>
<td>138</td>
<td>0.83s</td>
</tr>
<tr>
<td>DFTA + FS</td>
<td>12207</td>
<td>100%</td>
<td>201</td>
<td>0.66s</td>
</tr>
<tr>
<td>DC + FS</td>
<td>75208</td>
<td>100%</td>
<td>203</td>
<td>0.51s</td>
</tr>
</tbody>
</table>
References

- Mentor Graphic User Guide
- Synopsys TetraMax User Guide