Critical Area Weighted Pattern Optimization

Ron Press
Director, Technology Enablement
Tessent Test Solutions

February 2021
**Tessent TestKompress**

defect-oriented/automotive-grade ATPG

- New defect-oriented fault models detect defects missed by traditional methods
- New critical area based optimization uses defect likelihood & reduces pattern size by up to 40%

We have shown with test system results from a 14nm FinFET chip, that CAT patterns detect a huge amount of 4300 DPPM which are otherwise not detected with traditional SA and TDF patterns. We have shown on 150 SLT rejects that all passed traditional SA and TDF tests, but failed both SLT and CAT patterns. This resulted in the decision at Intel to make CAT patterns POR (Plan of Record)

*W. Howell (Intel), et.al, ITC 2018*

We have found that the bridge component alone reduces scan-related DPPMs by more than 700 compared even to cell-aware patterns

*P. Maxwell (ON Semiconductor), et.al, ETS 2017*
Executive summary

• New defect-oriented fault models show significant DPM impact
• Traditional ATPG metrics use faults treated as equally likely
• Critical area weighted pattern selection provides a consistent metric based on likelihood of detecting physical defects:
  – Choose the most effective patterns from multiple pattern sets
  – Order patterns by most effective
  – Pick the best sample of new pattern types to try in production
  – Create a smaller pattern set for multiple fault models in one ATPG run
• Note: patterns such as timing-aware might be separately needed to target marginalities
Agenda

• Choosing targets for pattern types and coverage
  – How do you know a pattern is going to be effective?

• Effectiveness of new pattern types
  – How choose which to supplement existing pattern sets with?

• Grading pattern value by likelihood to detect physical defects

• Automatic pattern sorting and selection
NEW PATTERN TYPES WITH SIGNIFICANT SI RESULTS
Tessent TestKompress defect-oriented/automotive-grade ATPG

Creates fault models and test patterns that detect defects at the transistor and interconnect levels missed by traditional methods.

We have shown with test system results from a 14nm FinFET chip, that CAT patterns detect a huge amount of 4300 DPPM which are otherwise not detected with traditional SA and TDF patterns. We have shown on 150 SLT rejects that all passed traditional SA and TDF tests, but failed both SLT and CAT patterns. This resulted in the decision at Intel to make CAT patterns POR (Plan of Record).

W. Howell (Intel), et.al, ITC 2018

We have found that the bridge component alone reduces scan-related DPPMs by more than 700 compared even to cell-aware patterns.

P. Maxwell (ON Semiconductor), et.al, ETS 2017
Timing-aware with cell-internal/cell-aware test

W. Howell (Intel), et.al, ITC 2018
Timing-aware cell-aware test usage

- Documented in “Scan and ATPG user’s Manual”
  - Defect-Oriented Test/Timing-Aware Pattern Generation for cell-internal Defect
  - Example script to generate patterns for small delay defects with topoff patterns for remaining delay defects:

```bash
set_fault_type udfm -delay_fault
read_fault_sites <tech>.udfm -filter {impact >= 50}
read_sdf <design>.sdf.gz
set_atpg_timing on
set_atpg_timing -clock_waveform <design settings>
add_faults -all
create_patterns
write_patterns CAT_timing.stil -stil -replace
set_atpg_timing off
delete_fault_sites -all
read_fault_sites <tech>.udfm
add_faults -all
read_patterns CAT_timing.stil
simulate_patterns -source external -store_pattern none
create_patterns
write_patterns CAT_delay.stil -stil -replace
```
Getting started with automotive-grade ATPG

• Tessent software tree contains two testcases for automotive-grade ATPG

1. Create an empty run directory in which you want to do the run. Then enter below command to check if all needed tools are accessible:
   ```bash
cellmodelgen -check_tools
   ```

2. Extract a few Unix C-shell scripts to run the tool fully automated on the CellModelGen 24nm library:
   ```bash
cellmodelgen -get_script all
   ```
USING CRITICAL AREA FOR RELATIVE LIKELIHOOD
Overview critical area calculation

Dimensions distance \( d \), length \( l \) or width \( w \) are expressed in units of “technology length” \( (tl) \):
- 1 \( tl \) equates to the width of metal1 in a technology
- Critical areas are expressed in units of “technology squares” \( (ts) \)

Critical area calculation considers:
- the spot size
- the distance between adjacent objects (for bridges)
- the width of the net object (for opens and vias)
- the length of the adjacency

Lower distance and larger the bridging length equals higher probability of a bridge

Larger distance and smaller net width equals higher probability of an open
TOTAL CRITICAL AREA BASED PATTERN SORTING
Automotive-grade ATPG and diagnosis flow

- GDS2, Verilog, ATPG, ...
- Rule deck
- GDS2, Verilog, ATPG, ...
- Rule deck

CellModelGen
- Cell-internal model creation
- Once per library

CellModelGen
- Inter-cell model creation
- Once per design block

Tessent TestKompress/Diagnosis*
- Layout Database (LDB) creation
- Interconnect bridge extraction
- Interconnect open extraction
- Cell pair extraction

LDB

Fault models

Tessent TestKompress
- ATPG

Test Patterns

Fail Data

Layout Database (LDB) creation

Inter-cell data

Cell data

Inter-cell data

Cell data

* Either using Tessent Diagnosis, or Tessent TestKompress w/Automotive-Grade option
Complete DOT – Putting it all together
Shortest test time & highest quality

- Just two ATPG runs will result in **shortest test time** with **highest pattern quality**
- The final production test patterns can now **first time** be **optimized** based on TCA

<table>
<thead>
<tr>
<th></th>
<th>#pat</th>
<th>#faults</th>
<th>TCA [ts]</th>
<th>FC [%]</th>
<th>TCA [%]</th>
<th>RE [%]</th>
<th>TC [%]</th>
<th>UC+AU [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>7405</td>
<td>25.9M</td>
<td>278M</td>
<td>88.58</td>
<td>91.10</td>
<td>2.44</td>
<td>91.01</td>
<td>8.99</td>
</tr>
<tr>
<td>Static</td>
<td>4782</td>
<td>22.9M</td>
<td>203M</td>
<td>96.27</td>
<td>96.61</td>
<td>2.76</td>
<td>99.16</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Source: F. Hapke, P. Maxwell, Mentor/ON Critical Area Based Test collaboration 2018
Defect-based pattern sorting

What
Ability to sort patterns based on Total Critical Area (TCA) values specified for UDFM faults

How
New option enables sorting of internal pattern sets by critical area

Value
- Easy way to determine the most effective patterns when truncating
- Faster defective part detection by ordering by likelihood of detecting silicon defects
- Any pattern set can be reordered based on critical area

Impact
Requires an Automotive-Grade ATPG license

---

```
order_patterns -critical_area
```

```
read_fault_sites cells.udfm
cells.udfm
read_fault_sites bridge.udfm
read_patterns stuck_at.patdb
read_patterns static_bridge.patdb -append
set_critical_area_options -reporting ON
add_faults -all
simulate_patterns -source external -store_patterns all
order_patterns -critical_area
```
Usage models

1. Pick a small sample of patterns for new fault model to try
2. Mix in new patterns with existing pattern set (shown in previous example)
3. Create patterns for all static or dynamic fault models in one ATPG run (recommended)
   1. Results in smaller pattern set than separate ATPG runs
   2. Method – read UDFMs and create patterns

• **NOTE:** static (1-cycle) and dynamic pattern generation and sorting need to be performed as separate runs
Guidelines and example testcase

• Automotive-grade ATPG is included in the automotive reference flow in the Tessent Shell User’s Manual
  – Includes an open source testcase in the software tree
• Critical area weighted pattern optimization is included in the same automotive reference flow and testcase
Summary

• Critical area weighted pattern selection results in a major change in your ability to have a consistent metric to select or optimize patterns for the best DPM in the shortest test time

• The suggested process for generating high quality defect-oriented/automotive-grade tests and critical area pattern sorting is included in the Tessent Shell User’s Manual
  – The documented process has a corresponding open source testcase in the Tessent software
THANK YOU