Outline

- Motivation for modular testing of SOCs
- Wrapper design
  - IEEE 1500 Standard
  - Optimization
- Test access mechanism design and optimization
- Test scheduling
- Exploiting port scalability to test embedded cores at multiple data rates
  - Virtual TAMs
  - Matching ATE data rates to scan frequencies of embedded cores
- Conclusions
Motivation

- System-on-chip (SOC) integrated circuits based on embedded intellectual property (IP) cores are now commonplace
  - SOCs include processors, memories, peripheral devices, IP cores, analog cores
  - Low cost, fast time-to-market, high performance, low power
  - Manufacturing test needed to detect manufacturing defects

System-on-Chip (SOC)

- Test access is limited
- Test sets must be transported to embedded logic

Philips Nexperia™ PNX8550 SOC: 338,839 flip-flops, 274 embedded cores, 10M logic gates, 40M logic transistors!
Shorten production cycles, and increasing complexity of modern electronic systems has forced designers to employ reuse based designs approaches.

System-on-Chip (SOC) is an example of such reuse based design approach where pre-designed, pre-verified cores are integrated into a system.

**System-on-Chip (SOC)**

- Shorten production cycles, and increasing complexity of modern electronic systems has forced designers to employ reuse based designs approaches.
- System-on-Chip (SOC) is an example of such reuse based design approach where pre-designed, pre-verified cores are integrated into a system.
Test access mechanism (TAM)

- An ATE is used to transport the test stimuli to the SOC. The produced responses are transported back to the ATE where they are compared with the expected responses.
- Memory cores are usually tested using a built-in self-test.

Modular Testing

- Test embedded cores using patterns provided by core vendor (test reuse)
- Test access mechanisms (TAMs) needed for test data transport: TAMs impact test time and test cost
- Test wrappers translate test data supplied by TAMs
- TAM optimization and test scheduling are critical
  - ITRS 05: Test data volume and testing time in 2010 will 30X that for today’s chips
Test Access Problem

1. How to isolate cores?
2. How to get patterns to cores?

Test Scheduling

- Test scheduling determines sequence of core tests on the TAMs
- Avoid test resource conflicts
- Minimize testing time
- Ineffective scheduling can increase tester data volume: Idle bits
IEEE 1500 Core Test Standard

- **Goals**
  - Define test interface between core and SOC
  - Core isolation
  - Plug-and-play protocols
- **Scope**
  - Standardize core isolation protocols and test modes
  - TAM design
  - Type of test to be applied
  - Test scheduling
IEEE 1500 Wrapper

Wrapper Modes: (1) Normal; (2) Serial Test; (3) 1-N Test; (4) Bypass; (5) Isolation; (6) Exttest


Wrapper Boundary Cells

wrapper boundary input cell

wrapper boundary output cell
Wrapper Usage

Wrapped Embedded Cores
Wrapper Operation Modes (I)

Normal Mode

Serial Bypass Mode

Wrapper Operation Modes (II)

Serial Internal Test Mode

Serial External Test Mode
Wrapper Operation Modes (III)

Parallel Internal Test Mode

Parallel External

Test Wrapper Optimization

Priority 1: Balanced Wrapper Scan Chains

Minimize length of longest wrapper scan in/out chain
Reducing TAM Width

Priority 2: Minimize wrapper scan chains created

Scan chain – 32 FF

I  I  I  8 FF  O
I  I  8 FF  O
I  8 FF

4 Wrapper scan chains

2 Wrapper scan chains

Scan chain – 32 FF

I  I  I  I  8 FF  8 FF  8 FF  O  O

Two-Priority Wrapper Design Algorithm

1. Minimize length of longest wrapper scan in/out chain
2. Minimize number of wrapper scan chains

Design_wrapper algorithm uses the BFD heuristic for Bin Design
Test Access Mechanisms

Types of TAMs

- Multiplexed access [Immaneni, ITC’90]
- Reuse system bus [Harrod, ITC’99]
- Transparent paths [Ghosh, DAC’98]
- Isolation rings [Whetsel, ITC’97]
- Test Bus [Varma, ITC’98]
- Test Rail [Marinissen, ITC’98]

TAM Design

1. Partial isolation rings
2. Multiplexing
TAM Design

3. Core Transparency

- Core A
- Core B

Test Bus Architecture

- Architecture:
  - A
  - B
  - C
  - D
  - E
  - F

- Schedule: Serial
  - TAM width:
    - A
    - B
    - C
    - D
    - E
    - F

- Combination of multiplexing and distribution
- Supports only serial schedule
- Core-external testing is cumbersome or impossible
TestRail Architecture

- Combination of Daisychain and Distribution architectures
- Cores connected to a TestRail can be tested simultaneously as well as sequentially
- Multiple wrappers can be activated simultaneously for Extest
- TestRails can be either fixed-width or flexible-width

**Fixed-width TestRails**

**Flexible-width TestRails**

---

Step-by-Step Approach to Wrapper/TAM Co-optimization

1. $P_W$: Wrapper design
2. $P_{AW}$: Core assignment + $P_W$
3. $P_{PAW}$: TAM width partitioning + $P_{AW}$
4. $P_{NPAW}$: Number of TAMs + $P_{PAW}$
Mathematical Programming Model for TAM Partitioning

- Variable $x_{ij} = 1$, if core $i$ assigned to TAM $j$
- Testing time of core $i$ on TAM width $w_j = T_i(w_j)$
- Testing time on TAM $j = \sum_i T_i(w_j) x_{ij}$
- **Objective:** Minimize $T = \max_j \sum_i T_i(w_j) x_{ij}$
- **Constraints**
  1. $\sum_i x_{ij} = 1$, every core connected to exactly one TAM
  2. $\sum_i w_j = W$, total TAM width is $W$
  3. $w_j \leq w_{\text{max}}$, maximum width of any TAM is $w_{\text{max}}$

TAM Design and Test Scheduling

- Given the test set parameters for the cores and the total TAM width $W$
- Assign a part of $W$ to each core, design a wrapper for each core, and determine the test schedule,
- Such that
  - $W$ is not exceeded at any time and
  - Testing time is minimized
Architectures Determine Schedules

Slide provided by Erik Jan Marinissen, Philips Research Labs

Rectangle Model for Test Buses

Three test buses
Each core on same bus gets equal, fixed TAM width
**Rectangle Representation**

- Testing time $T_i(w_j)$ for Core $i$ and TAM width $j$
- Rectangle $R_{ij}$
- Set of rectangles $R_i$ for each core
- Collection of rectangles $R$ for SOC

**Rectangle Packing Problem**

- Given collection $R$ of rectangle sets for the SOC cores,
- Select one rectangle $R_{ij}$ for each Core $i$
- Pack the selected rectangles into a bin of fixed height,
- Such that bin width is minimized
Packed Bin = TAM Design + Test Schedule

Bin height = total TAM width
Bin width = SOC testing time

Rectangle area = tester memory for core test
Empty space = wasted tester memory

Core 1
Core 2
Core 3
Core 4
Core 5
Core 6
Core 7
Core 8

Preferred TAM Widths

- Only Pareto-optimal TAM widths are considered
- Procedure: Tests are scheduled at current time in decreasing order of preferred TAM width until no TAM width remains
Non-Preferred Rectangles:

Fill Idle Time

Core 1-P

Core 2-P

Core 3-P

Core 1

Core 2

Core 3

Next_time

Increasing Current TAM Widths

• Modify current rectangle that will benefit the most from an increase in TAM width

If idle time is inevitable, advance Current_time and repeat procedure from the start
1. Break up a long test to fill in idle time gaps in schedule
2. Break up a test to avoid a potential conflict

Power Constraints

1. Excessive test concurrency can burn the SOC
2. Cores that do not operate together may be tested together
Precedence Relationships

1. Memory BIST before external test
2. Tests that detect more faults applied first
3. Shorter tests first

Example: Core 4 must complete before Core 6 begins

---

ITC 2002 SOC Test Benchmarks
Initiative led by Philips Research Labs and Duke University

http://www.hitech-projects.com/itc02socbenchm/

- Freely downloadable through Internet
- Currently 12 SOC benchmarks
  - 5 from academic contributors
  - 7 from industrial contributors

Awareness, Interest, Usage

- E-mail reflector with 43 subscribers
- Awareness panel session at TECS’02: ‘How Useful Are The ITC’02 SOC Test Benchmarks?’
- Special session at ITC’02 + papers in other sessions
- Numerous papers using the benchmarks in journals and conferences
- Extensions added by others (layout data, power consumption data, functional interconnects, etc.)
Current-Generation ATEs

- Port scalability features
- Digital speeds of up to 2.5 Gbps
- Application flexibility

Every port of a tester, consisting of multiple channels, can be configured at a desired data rate.

Virtual TAMs

- Embedded core test frequency is limited by scan frequency
  - Scan frequencies are low to meet power, routing, and clock skew constraints
- Virtual TAMs allow use of high frequency ATE pins
- How can we match fast ATE data rates to slow scan frequencies?
Bandwidth Matching

ATE pins: $W_{ATE} = 4$
ATE frequency factor: $n = 4$
High frequency pins $U = 2$

$$U \times f_{ATE} + (W_{TAM} - U) \times f_{TAM} = U \times n \times f_{TAM} + (W_{ATE} - U) \times f_{TAM}$$

Implementation of Bandwidth Matching
Selection of $U$ and $n$

- Testing of SOC is often dominated by the testing time of bottleneck cores
- Testing time of SOCs containing bottleneck cores does not decrease for TAM widths greater than $W^*$
- The lower bound on test time in such SOCs is $T^*$ corresponding to TAM width $W^*$

### SOCs with Bottleneck Cores

<table>
<thead>
<tr>
<th>SOC</th>
<th>$W^*$ (bits)</th>
<th>$T^*$ (clock cycles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>u226</td>
<td>48</td>
<td>5333</td>
</tr>
<tr>
<td>d281</td>
<td>48</td>
<td>3926</td>
</tr>
<tr>
<td>g1023</td>
<td>40</td>
<td>14794</td>
</tr>
<tr>
<td>p34392</td>
<td>36</td>
<td>544579</td>
</tr>
<tr>
<td>t512505</td>
<td>36</td>
<td>5228420</td>
</tr>
<tr>
<td>h953</td>
<td>16</td>
<td>119357</td>
</tr>
<tr>
<td>f2126</td>
<td>16</td>
<td>335334</td>
</tr>
<tr>
<td>q12710</td>
<td>16</td>
<td>2222349</td>
</tr>
</tbody>
</table>
Relationship of $U$, $n$ and $W^*$

- $U$ and $n$ should be chosen such that total virtual TAM width $W$ does not exceed $W^*$

\[ W \leq W^* \]

\[ W = nU + (W_{ATE} - U) \]

\[ U(n - 1) \leq W^* - W_{ATE} \]

---

Variation of $U$ with $n$
Multiple-Speed TAM Architectures

- Exploit port-scalability of ATEs
- Facilitate efficient use of high data-rate tester channels
- Unlike virtual TAMs, avoid on-chip hardware overhead
- Reduce testing time of bottleneck cores
Problem Formulation

• Dual-speed optimization problem

Given:

Determine the wrapper design, TAM width and test data rate for each core, and the SOC test schedule such that:

• the total number of TAM wires utilized at any moment does not exceed $W$
• the number of TAM wires driven at the high data rate does not exceed $V$
• the SOC testing time is minimized

Selection of Data Rate for a Core

Core 5 in SOC p93791

<table>
<thead>
<tr>
<th>$f$</th>
<th>$V$</th>
<th>$T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10</td>
<td>14026.9 $\mu$s</td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>11398.9 $\mu$s</td>
</tr>
</tbody>
</table>
Matching Core Scan Frequencies to ATE Data Rates

Baseline Case 1

Baseline Case 2

Test time

Test time
Problem Statement

Given

- Test data parameters for $N$ embedded cores
- Maximum scan frequency $f_i$ for each core $i$
- SOC-level TAM width $W$

Determine

- The number of TAM partitions $B$
- Width $w_j$ and scan frequency $f_j$ of each TAM partition $j$
- Assignment of cores to TAM partitions

Such that

- TAM frequency does not exceed the maximum scan frequency of any core assigned to that TAM partition
- The overall test time is minimized
- The sum of the widths of all the TAM partitions does not exceed $W$
Solution Techniques

- Lower bound on test time based on geometric arguments (rectangle packing)
- Integer linear programming
  - Exact optimization method, limited to small problem instances
- Fast heuristic method
  - Scalable, close to optimal results

Comparison with Baseline

<table>
<thead>
<tr>
<th>TAM Width</th>
<th>Test time (μs)</th>
<th>p22810</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(5 frequencies: 10 to 50 MHz)</td>
</tr>
<tr>
<td></td>
<td>LB</td>
<td>baseline</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>24</td>
<td>32</td>
<td>40</td>
</tr>
<tr>
<td>32</td>
<td>48</td>
<td>56</td>
</tr>
</tbody>
</table>

37% improvement
### Conclusions

- Test reuse and test time minimization are necessary to reduce test cost for SOCs.
- Wrapper/TAM optimization and test scheduling can reduce test time for core-based SOCs.
- Virtual TAMs offer several advantages for SOC testing:
  - On-chip TAM wires are not limited by the number of available pins on the SOC.
  - Better utilization of high-speed ATE channels reduces testing times.
- TAM architectures can be designed to match port-scalable ATE channels to different scan frequencies of embedded cores.