Self-Testable, Self-Adaptable and Error-Resilient System Design -
coping with increasing variability and reliability concerns

Tim Cheng
Electrical and Computer Engineering
Drivers of System-on-a-Chip/System-in-a-Package Design & Test Technologies

- Decreasing design window
- Less tolerance for design revisions

- Exponentially more transistors
- Increasing complexity in system context

- Coupling cap
- Signal integrity
- Inductance
- Leakage

- Greater diversity of on-chip elements: processors, SW, RF, memory, analog, high-speed bus
Challenges Facing the Next Decade in Integrated System Design

- Managing and exploiting design partitioning and trade-offs for heterogeneous systems
  - HW, SW, analog, RF, MEMS, optical, etc..
- Power and energy
- Verification and test
- Reliability and robustness
- Implementation fabrics beyond silicon
Increasing Failure Sources and Failure Rates

- Design errors
- Soft errors
- Random defects
- Parametric variations

![Failed Chips](image)

![On-Die Temperature variations](image)

- SEU
- On-Die Temperature variations

![Temperature (C)](image)

- On-Die Temperature variations
- SEU
- n+ n+
- p-substrate
- STI
- \( Q_{CRIT} \)

- Defects
- Parametric variations

- SEU
- Random defects
- Parametric variations
Harder to Design Reliable System-Chips

• First-silicon success rate has been dropping
  - <30% for complex ASIC/SoC@.13m
  - Pre-silicon logic bugs have been increasing at 3X-4X/generation for Intel’s processors
• Yield has been dropping for volume production
  - IBM’s 8-core Cell-Processor chips: ~10-20% yield
• “Better than worst-case” design resulting in failures w/o defects
  - Increase in variation of process parameters with scaling
  - Worst-case design getting way too conservative
• One-time-factory production testing will be too costly and insufficient for failure screening
New Design and Test Paradigm: Reliable Systems With Unreliable Components

- Systems must be designed to cope with failures
- Efficient silicon debug is a must
  - Design for debugging would become necessary
- Must have embedded self-test for error detection
  - For both testing in manufacturing line and in-the-field
  - Both on-line and off-line testing
- Must be re-configurable and adaptable for error recovery
  - Using spares to replace defective parts
  - Using redundancy to mask errors
  - Using self-tuning to compensate variations
Some of Our Research Results

• Embedded Software-Based Self-Test for SoC
  - Reuse of embedded processors and on-chip resources for self-test and diagnosis

• Test, Characterization and Diagnosis for High Speed Serial IO Interfaces

• Silicon Debug for Timing Failures

• Formal Equivalence Checking between System Specification and RTL Code
Embedded-Software-Based Self-Test For Programmable Systems

- Reuse of on-chip programmable components for test
- Processor/DSP/FPGA cores for on-chip test generation, measurement, response analysis and even diagnosis
  - Self-test a processor using its instruction set for high fault coverage
  - Use the tested processor/DSP to test buses, interfaces and other components, including analog and mixed-signal components

Test and diagnosis are applications of a programmable SOC!!
Embedded SW-Based Self-Testing for Programmable System Chips

- Low-cost tester
- High-quality at-speed test
- Low test overhead
- Non-intrusive

Test in normal operational mode
- No violation of power consumption
- More accurate speed-binning

Loading test program at low speed
Self-test at operational speed
Unloading response signature at low speed
DSP-Based Self-Test for Analog/Mixed-Signal Components

- DSP-based embedded self-test - on-chip tester
  - Relieve need of expensive ATE
  - Reduce external noise
- Practical issues
  - Test quality limited by DAC/ADC
  - DAC/ADC are not always available, and must be tested first
Self-Test for Analog/MS Components

- A self-test architecture using the delta-sigma modulation principle for signal generation and for waveform acquisition

[Diagram showing the self-test architecture involving ATE and SOC components, including test stimuli, software ΔΣ modulator, programmable core + memory, on-chip DAC/low. res. DAC, on-chip ADC/1-bit ΔΣ modulator, and response analysis.]
Bit-Error-Rate (BER) Estimation for High-Speed Serial Interface

- Most applications require $10^{-12}$ or even lower BER
  - Measuring BER impractical for production testing
- BER can be estimated using parameters with strong correlation with BER:
  - Spectral information of jitter
    - Frequencies and amplitudes of Periodic Jitter (PJ)
    - Rms value of Random Jitter (RJ)
  - Jitter transfer characteristics of a CDR circuit
    - Magnitude response (low pass filter characteristic)
    - Phase response (timing response in clock recovery)
- Channel characteristics
BER Estimation - HW Validation

1. Measure magnitude response of a CDR circuit

2. Measure phase response of a CDR circuit

3. Measure BER

Maxim's 2.5 Gbps CDR circuit (MAX 3873A)

Synthesys Research's BERTScope

Clock with jitter

2.5 Gbps Data with jitter

Recovered clock
Results - Clk-like and PRBS Data

- Measured and estimated BER match very well
- The difference is larger at low (~$10^{-12}$) level, due to the bounded nature of RJ in practice

*Clock-like pattern w/ 0.5T PJ*

*Clock-like pattern w/ 0.45T PJ*

*PRBS pattern w/ 0.5T PJ*
Testable/Debuggable Design for Adaptive Equalizer in High Speed Serial-Link

- A design-for-testability (DfT) solution for adaptive equalizer (EQ) in high-speed IO
- Applicable to various EQ architectures
- Addressing RX’s observability & controllability problems
- Lower test cost and higher fault coverage than conventional eye-diagram-based method
Testable Design for Adaptive Equalizer

- Minor HW modification of EQ: a FF chain (storing tap-coefficients), a pattern generator, and some switches
  - Extra circuits are all digital
- Addressing both characterization and production testing

General Architecture for DFE/FFE

Testable Decision-Feedback Equalizer (DFE) or Feed-Forward Equalizer (FFE):

- Minor HW modification of EQ: a FF chain (storing tap-coefficients), a pattern generator, and some switches
- Extra circuits are all digital
- Addressing both characterization and production testing
Experiments: Testable Equalizers

- Applying digital tests and examining tap-coefficients for fault detection and diagnosis
- Demonstrating much higher fault coverage than eye-diagram-based approach
From Test to Recovery/Reconfiguration - Examples

• Memory
  - “BIST → BISD → BISR” a common practice

• Analog/RF/High-speed IO components
  - Digitally assisted self-calibration: fine-tuning performance; more robust to process, temperature and voltage variations...

• Dynamic circuits
  - Using programmable keeper and on-chip leakage sensors for tuning performance and robustness
Dynamic Circuit Using Static Keeper

- Keeper upsizing degrades average performance
Pessimistic Design Hurts Performance

- Substantial variation in leakage across dies
- 4-5X variation between nominal and worst-case leakage
- Performance determined at nominal leakage
- Robustness determined at worst-case leakage
Programmable Keeper for Dynamic Ckts

3-bit programmable keeper

Ref: C. Kim and K Roy, Purdue

Opportunistic speedup via keeper downsizing
On-Die Leakage Sensor

- High leakage sensing gain
- Compact analog design sharing bias generators

<table>
<thead>
<tr>
<th>Technology</th>
<th>90nm dual Vt CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DD}$</td>
<td>1.2V</td>
</tr>
<tr>
<td>Resolution</td>
<td>7 levels</td>
</tr>
<tr>
<td>Power consumption</td>
<td>0.66 mW @80°C</td>
</tr>
<tr>
<td>Dimensions</td>
<td>83 X 73 µm²</td>
</tr>
</tbody>
</table>

C. Kim et al., VLSI Circuits Symp. '04
Leakage Binning Results

Output codes from leakage sensor
Test & Tuning Process for Self-Tunable Design

Fab

wafer test

Process detection

Leakage measurement

On-die leakage sensor

Program using fuses

Customer

Package test

Burn in

Assembly
Digital Logic - Exploring Redundancy and Reconfiguration Tradeoffs

- Redundancy
  - Suitable for soft/transient/marginality errors
  - Different forms:
    - Hardware redundancy
    - Time redundancy
    - Information redundancy

- Reconfiguration
  - Suitable for hard errors (e.g. defects)

- Design methodology and tools for area, power, performance and reliability tradeoffs?
Summary

- Quality can't be added, it has to be designed in.
- Cost-effective embedded self-test will replace existing manufacturing test methodologies for heterogeneous SoC/SiP
- Post-silicon tuning/calibration/reconfiguration is becoming promising, and necessary, for Si nano systems