EECS 579: Built-in Self-Test 3

Outline

• Implementing BIST by regularization
  Adder
  ALU
  RAM
• Commercial BIST approaches
  LOCSD
  STUMPS
  CSTP
• Case Study
  Bosch AE11 microcontroller

Regular Circuits

• Circuits composed of (nearly) identical cells with (nearly) uniform interconnections
• Structured as $n$-dimensional iterative logic arrays or trees
• Regular circuits tend to be easy to test

Examples

• Random-access memories (RAMs and ROMs)
• Arithmetic circuits: adders, multipliers, etc.
• Data-transfer circuits: (de)multiplexers, decoders, etc.
• Nearly regular circuits can often be made regular for testing purposes (regularization)
**Regular Circuits and BIST**

(Nearly) identical test patterns implying small, easily generated test sets

(Nearly) identical responses allowing use of equality checkers as response monitors

---

**Example: Ripple-Carry Adder**

- All CF faults in an $N$-bit RC adder can be detected by a constant number of test patterns for any $N$, implying that it is C-testable.
**Example: Ripple-Carry Adder**

- For BIST, an $N$-bit RC adder can be tested using $2N + 6$ patterns that produce identical responses from all cells, implying that it is *I*-testable.
- Faults can be detected by comparing the cell outputs

![Adder Diagram]

**BIST via Regularization**

- Add logic as necessary to allow temporary creation of a regular array for self-testing
- Modify the cells as necessary to make the regularized array C- and/or I-testable

![Regularization Diagram]
Nearly Regular ALU: 74X381

Control logic  
Four-bit datapath logic (bit-sliced)

Regularized Version of 74X381

From $C_X$  
Added control lines

Added gates (shaded)
BIST Implementation of 74X381

Regularized 1-bit ALU modules

Test pattern generator based on NLFSR

Duplicate copy of \( C_X \)

Response shift register and equality checker

Error latch

ERROR1 ERROR2

C\(_X\) C\(_3\) C\(_2\) C\(_1\) C\(_0\)

TEST CLOCK

Regularizing a Tree Circuit: 74X154

Original circuit: 1-out-of-16 decoder/demultiplexer

Requires \( O(2^n) = 32 \) tests to detect all SSL faults

Test

Regularized circuit

Requires \( O(n) = 11 \) tests to detect all SSL faults
**BIST for RAMs**

**Testing Problems**
- High component count and density
- Complex fault types, e.g., pattern sensitivity
- Long testing times to achieve high fault coverage: \( O(N^k) \) for an \( N \)-bit RAM, \( 1 \leq k \leq 2 \).
- Large overhead or limited fault coverage for self-testing via conventional techniques

**BIST Approaches**
- EC/ED code circuits (concurrent)
- Special non-concurrent test logic that exploits the RAM’s inherent regularity

---

**RAM Layout**

![RAM Layout Diagram]
**RAM as Nearly-Regular Array**

- **Address**
- **Data**
- **Control**

**Self-Testing RAM** [You and Hayes 88]

- Modified standard RAM with on-chip TG and RM logic
- Tests are derived from standard RAM tests and are highly regular
- Array cells are normal storage subarrays with modified peripheral circuitry
- RAM behaves like a shift register during testing
- Many cells are tested in parallel to reduce testing time

**Test Derivation**

- Complete test subsequences are derived for all expected RAM faults
  1. Apply read/write excitation to set of cells in all storage arrays
  2. Read test cells and background pattern
  3. Modify background pattern for next test step
- Individual test subsequences are overlapped to form a composite C/I-style test for an entire storage subarray
Self-Testing RAM

- The RAM is made fully self-testing for all recognized failure modes
- Testing time is $O(R^{0.5})$ where $R$ is the number of cells in a storage subarray $C$
- Area overhead due to BIST is a few percent of total area for storage capacities in the multimegabit range
- Two or three extra I/O pins needed to initiate self-testing and observe test responses
LOCST (LSSD On-Chip Self-Test)

- Centralized and separate BIST
- (LSSD boundary) scan paths around the CUT
- Serial, LFSR-based test pattern generation and response compression

STUMPS

- Centralized and separate BIST
- Multiple scan paths without boundary scan
- Designed to have low overall testing time
CSTP (Circular Self-Test Path)

- Mixes conventional and special self-test registers
- Self-test registers have three modes: normal, scan, and test. In the test mode, the system data is XORed with scan data.

![Self-test cell diagram]

- All I/O lines are linked in a circular (boundary) scan path.

CSTP (contd.)

**Features**
- Initializable registers not scanned
- TG and response compression done in scan path

**Advantages:**
- Low area overhead
- Easy design
- Simple test control

**Drawbacks:**
- Low fault coverage
**Case Study: Bosch AE11 Microcontroller**

- Single-chip, self-testing microcontroller designed to detect hardware faults rapidly under all operating conditions
- Intended for safety-critical applications like automotive control
- Compatible with Intel 8051 8-bit ISA; 4-KB RAM and I/O modules

**Self-testing Features:**
- Parity checking throughout the system
- Parity checking and ALU parity prediction in the CPU datapath
- Program control-flow checking via signature monitoring
- Self-checking address decoding logic in the RAM
- Programmable watchdog timer
- Pseudorandom test and I\text{DDQ} testing of peripheral modules
- Power supply and temperature monitoring
- Test control employing boundary scan and a TAP controller

**Bosch AE11 Microcontroller (contd.)**

- Test controller
- Test access port (TAP)
- Analog-digital converter (ADC)
- Serial I/O
- Watchdog timer
- Misc. I/O modules
- System bus
Bosch AE11 Microcontroller (contd.)

CPU Testing

• Datapath uses parity prediction, which has low cost and is compatible with the AE11’s overall use of parity codes.
• Control unit applies parity checking to control words
• Software control-flow checking is supported: Signatures are computed during compilation and are inserted automatically into programs. Special AE11 instructions monitor these signatures.

RAM Testing

• The AE11’s fault latency requirements rule out conventional RAM BIST methods which are slow and destroy stored data
• Parity check bit are added to address and data buses
• RAM parity checkers are self-checking
• Special BIST logic detects word line and address decoder faults
• Special circuits detect bridging faults, including resistive shorts

I/O Testing

• The I/O subsystem also uses parity checking
• Peripheral modules are tested on-line via BIST logic and I\textsubscript{DDQ} testing that employs on-chip current monitors.

Test Control

• Testing functions are handled by a test controller that conforms to the IEEE 1149.1 boundary scan standard
• The special BIST logic uses pseudorandom test and signature generation implemented by LFSRs in the CPU and MISRs in the I/O modules
• The pseudorandom test patterns are believed to provide protection against unknown or “non-targeted” hardware faults, as well as standard SSL faults.
**Bosch AE11 Microcontroller (contd.)**

**Test Management**
- Start-up tests check that the major subsystems, CPU, RAM and peripheral modules, are operational
- In normal operation, concurrent checking circuits flag the errors when they occur
- CPU is interrupted periodically to execute test procedures such as the I_DDQ tests and various functional tests

**Performance and Cost**
- Estimated to achieve more than 99.7% coverage of modeled faults and errors
- Less than 35% chip area overhead
- Less than 15% performance loss