FPGA Characteristics

- Configuration memory
  - 32Kbits – 79Mbits
- Array of Programmable Logic Blocks (PLBs)
  - 100 – 25,920 PLBs per FPGA
    - 1–8 4-input LUTs and 1–8 flip-flops per PLB
- Programmable interconnect network
  - Wire segments
    - 45 – 406 per PLB
  - Programmable switches
    - 140 – 4,100 per PLB
- Programmable I/O cells
  - Bi-direction buffer with flip-flops/latches
    - 62 – 1,200 per FPGA
Important Trends in FPGAs

- Dynamic partial reconfiguration
- Incorporating specialized cores
  - RAMs - single-port, dual-port, FIFO, ECC
    - 128 - 18K bits per RAM
    - 4 - 576 per FPGA
  - DSPs including multipliers, accumulators, etc.
    - Up to 512 per FPGA
  - Embedded processor cores
    - Up to 2 hard cores per FPGA
    - Also support soft processor cores synthesized in FPGA
  - Internal access to configuration memory
    - Write and read access by embedded processor core
- FPGAs becoming more like SoCs
- ASICs & SoCs now incorporate FPGA cores
FPGA Testing Challenges

- Programmability
  - Must test all modes of operation
- Architectures designed for applications
  - Testing issues/problems left to product/test engineers
- CAD tools designed for high-level synthesis
  - Do not support control of proper test conditions
- Constantly growing sizes
  - Reconfiguration dominates test time
- Constantly changing architectures
  - Architectural features/limitations directly affect testability and test development
- Incorporation of many new/different cores
CAD Tool Features vs. Testability

- Controlling test conditions with CAD tools
  - Oriented for design
  - Oriented for synthesis
  - For testing we need to:
    - Control unselected inputs to logic multiplexers
      - Test for stuck-at faults
    - Control opposite logic values on at least one unselected input for MUX PIPs
      - Test for PIP stuck-on faults
      - # test configurations = # MUX inputs
  - DRC complaints about antennas & stubs
    - Delete signals for test conditions
FPGA Testing

- Typically partitioned for logic and routing
  - But both resources needed to test each other

- External testing
  - Good for manufacture testing only
  - Tests applied via I/O pins
    - Package dependent and limited by I/O pins
  - Boundary Scan (only with INTEST)
    - Extremely long test time

- Internal Testing (BIST)
  - Good for manufacturing & system-level test
  - Good for embedded FPGA cores
FPGA Testing

- Application independent testing
  - Test all resources in FPGA
    - Good for manufacturing testing
  - Requires many test configurations
    - Long test time - downloads dominate test time
  - No area/performance penalty in system

- Application specific testing
  - Test only resources used by system function
  - Requires fewer configurations
    - But requires new tests for new applications
      - Good for system-level testing only
  - Area/performance penalty for test circuitry
System-Level FPGA Testing

- System-level test of FPGA-based designs
  - Diagnostic software for test in system mode
  - Many months of diagnostic code development
  - Good diagnostic resolution difficult to achieve
- DFT/BIST in FPGA (for system-level test)
  - Area penalty typically 10-30%
  - Performance penalty typically 2-3 gate delays
  - Less logic for system function
  - May require larger or more FPGAs
  - Longer design time
BIST for FPGAs

Basic idea: reprogram FPGA to test itself
- BIST logic disappears after test
- No area overhead or performance penalties

Applicable to all levels of testing
- Application independent testing
- A generic test for a generic component
- Good diagnostic resolution
  - To faulty PLB or wire segment/switch within FPGA
  - No diagnostic code development or DFT design

Cost:
- Memory to store BIST configurations
  - Goal: minimize number of configurations
- Download time to execute BIST configurations
  - Goal: minimize downloads
FPGA Architectures

- **Early FPGAs**
  - $N \times N$ array of unit cells
    - Unit cell = CLB + routing
    - Special routing along center axes
  - I/O cells around perimeter

- **Next Generation FPGAs**
  - $M \times N$ array of unit cells
  - Added small block RAMs at edges

- **More Recent FPGAs**
  - Added larger block RAMs in array
  - Added multipliers
  - Added Processor Cores (PC)

- **Latest FPGAs**
  - Added DSP cores w/multipliers
  - I/O cells along columns for BGA
Look-up Tables

- Using multiplexer example
- Configuration memory holds truth table
- Input signals connect to select inputs of multiplexers to select output value of truth table for any given input value
Look-up Table Based RAMs

- Normal LUT mode performs read operations
- Address decoder with write enable generates load signals to latches for write operations
- Small RAMs but can be combined for larger RAMs
Test Configurations for a Simple PLB

- Two 3-input LUTs
  - Can implement any 4-input combinational logic function
- 1 flip-flop
  - Programmable:
    - Active levels
    - Clock edge
    - Set/reset
- 22 configuration memory bits
  - 8 per LUT
    - C0-7
    - S0-7
  - 6 controls
    - CB0-7

<table>
<thead>
<tr>
<th>Config Bits</th>
<th>Configuration #1</th>
<th>Configuration #2</th>
<th>Configuration #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUT C (C7-C0)</td>
<td>XNOR (01101001)</td>
<td>XOR (10010110)</td>
<td>XOR (10010110)</td>
</tr>
<tr>
<td>LUT S (S7-S0)</td>
<td>XOR (10010110)</td>
<td>XNOR (01101001)</td>
<td>XNOR (01101001)</td>
</tr>
<tr>
<td>CB0-CB5</td>
<td>000010</td>
<td>111110</td>
<td>000001</td>
</tr>
<tr>
<td>Individual FC</td>
<td>149/174 = 85.6%</td>
<td>149/174 = 85.6%</td>
<td>108/174 = 62.1%</td>
</tr>
<tr>
<td>Cumulative FC</td>
<td>85.6%</td>
<td>97.7%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Diagram showing the configuration bits and their distribution.
Input/Output Cells

- Bi-directional buffers
  - Programmable for input or output signals
  - Tri-state control for bi-directional operation
  - Flip-flops/latches for improved timing
    - Set-up and hold times
    - Clock-to-output delay
  - Pull-up/down resistors

- Routing resources
  - Connections to core of array

- Programmable I/O voltage & current levels
Interconnect Network

- Wire segments of varying length
  - $xN = N$ PLBs in length
  - Typical values of $N = 1, 2, 4, 6, 8$
- Long lines
  - $xH = \text{half the array in length}$
  - $xL = \text{full array in length}$

- Programmable Interconnect Points (PIPs)
  - Transmission gate connects to 2 wire segments
  - Controlled by configuration memory bit
  - Four basic types of PIPs

Diagram:
- Wire A
- Wire B
- Transmission gate connects to 2 wire segments
- Controlled by configuration memory bit
- Four basic types of PIPs
Programmable Interconnect Points

- **Break-point PIP**
  - Connect or isolate 2 wire segments

- **Cross-point PIP**
  - 2 nets straight through
  - 1 net turns corner and/or fans out

- **Compound cross-point PIP**
  - Collection of 6 break-point PIPs
    - Can route 2 isolated signal nets

- **Multiplexer PIP**
  - Directional and buffered
  - Main routing resource in recent FPGAs
  - Select 1-of-\(N\) inputs for output
    - Decoded MUX PIP – \(N\) configuration bits select from \(2^N\) inputs
    - Non-decoded MUX PIP – 1 configuration bit per input
On-line BIST, Diagnosis & FT

- **Roving Self Testing AREas (STARs)**
  - Test programmable logic & interconnect in FPGA
  - Horizontal STAR (roves up and down FPGA)
    - Tests horizontal routing resources
  - Vertical STAR (roves across FPGA)
    - Tests logic and vertical routing resources

\[ \text{FPGA} + \text{V-STAR} + \text{H-STAR} = \text{System Function} = \text{Self-Testing} \]
On-Line BIST, Diagnosis, & FT

- Exploits dynamic partial reconfiguration
- STARs rove across FPGA performing BIST
- Diagnosis when faults are detected
- Reconfiguration of system function to avoid faults when STAR moves to new position
## FPGA BIST Configurations

<table>
<thead>
<tr>
<th>FPGA</th>
<th>Logic</th>
<th>Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORCA</td>
<td>2C</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>2CA</td>
<td>14</td>
</tr>
<tr>
<td>Atmel</td>
<td>AT94K/40K</td>
<td>4</td>
</tr>
<tr>
<td>Cypress</td>
<td>39K</td>
<td>20</td>
</tr>
<tr>
<td>Xilinx</td>
<td>4000E/Spartan</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>4000XL/XLA</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Virtex-I/Spartan-II</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Virtex-4</td>
<td>15</td>
</tr>
</tbody>
</table>

Notes: Logic BIST configurations typically applied 2 times
Configurations for embedded cores not included
First Logic BIST Approach

- Schematic entry difficult
  - Manual placement needed to test all PLBs
- Routing difficult with larger \( N \times N \) arrays
  - Routing complexity = \( O(N^2) \)
- Global routing resources heavily used

Diagram:
- BIST start
- TPG
- BUT
- ORA
- LUT
- FF
- C+1
- pass/fail

...
Second Logic BIST (Iterative Logic Array)

- **Advantages:**
  - Linear routing complexity
  - Easily scaleable
  - Algorithmic PLB placement & routing with NCL

- **Disadvantages:**
  - 3 test sessions
  - Difficult to propagate test patterns through BUTs
    - Particularly for sequential logic functions

---

**Diagram:**
- TPG
- BUT
- Helper
- ILA cell
- ORA

- Global routing
- Local routing
- From other ILA

---

**Advantages:**
- Linear routing complexity
- Easily scaleable
- Algorithmic PLB placement & routing with NCL

**Disadvantages:**
- 3 test sessions
- Difficult to propagate test patterns through BUTs
  - Particularly for sequential logic functions
Third Logic BIST (Hybrid)

- Two test sessions
  - Row or column orientation
- Good balance of global & local routing
- Algorithmic placement & routing
  - Good for dynamic partial reconfiguration
  - Easily scalable with NCL
Output Response Analyzers

- Comparison-based
  - XOR with OR feedback from flip-flop
    ✓ Latches mismatches observed due to faults

- Results retrieval
  - ORA with shift register
    ✓ Requires additional logic
  - Configuration memory readback
    ✓ Read contents of ORA flip-flops
      ✓ Good with partial configuration memory readback capabilities

![Diagram of Output Response Analyzers](image)

Pass/Fail shift data shift mode

Pass/Fail

Read contents of ORA flip-flops

Pass/Fail

Good with partial configuration memory readback capabilities
Pathological Case

To escape detection **all** of the following must be true:
- X & Y have same position in both TPGs in row 1
- V & Z have same position in both TPGs in row 8
- X & Y have equivalent faults
- V & Z have equivalent faults
- X & Y cause TPGs in row 1 to skip patterns that detect V & Z
- V & Z cause TPGs in row 8 to skip patterns that detect X & Y

But rotating test sessions will detect these faults!!
Diagnosis Based on BIST Results

Step 1: Record ORA results
Step 2: Mark BUTs good between consecutive ORAs with 0s
Step 3: Mark BUTs good for every two adjacent 0s followed by empty cell
Step 4: Mark BUTs bad for every consecutive 0 and 1 followed by empty cell
Step 5: Inconsistencies mean fault in ORA or in routing resources
Step 6: Unique diagnosis if all BUTs marked faulty or fault-free

Note:
Row 4: BUTs 1 & 2 have equivalent faults

Ambiguities:
Row 2: BUT 6 may be faulty or fault-free
Row 6: BUT 6 may be faulty or fault-free
Row 3: BUT 5 and/or BUT 6 is faulty
Row 5: BUTs 1 & 2 may be fault-free or faulty (with equivalent faults)

rotate BIST 90° to remove ambiguities
Circular-Comparison BIST

- Circular comparison of BUTs
  - Better diagnostic resolution
  - Possibly better fault detection

- Need TPGs
  - Embedded processor
  - Other cores
    - DSP
    - Embedded RAM
      - DSP counter reads
      - RAM (ROM) with test patterns

- Need sufficient routing resources
  - Available in many newer FPGAs
Circular Comparison Diagnosis

Step 1: Record ORA results
Step 2: Mark all CUTs associated with two or more consecutive ORAs with 0s (0=fault-free)
Step 3: Recursively mark CUTs with 1 (1=faulty) for every consecutive 0 and 1 followed by empty cell
Step 4: Inconsistencies mean fault in CUT-to-ORA routing resources or in ORAs if they have not been tested and known to be fault-free
Step 5: Unique diagnosis if all CUTs marked faulty or fault-free

Notes:
- No loss of diagnostic resolution at edge of array - *there are no edges*
- C3 and C4 have equivalent faults

<table>
<thead>
<tr>
<th>O_91</th>
<th>C_1</th>
<th>O_12</th>
<th>C_2</th>
<th>O_23</th>
<th>C_3</th>
<th>O_34</th>
<th>C_4</th>
<th>O_45</th>
<th>C_5</th>
<th>O_56</th>
<th>C_6</th>
<th>O_67</th>
<th>C_7</th>
<th>O_78</th>
<th>C_8</th>
<th>O_89</th>
<th>C_9</th>
<th>O_91</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

CUT=Circuit Under Test (CLBs, DSPs, RAMs, etc.)
Logic BIST for Large FPGAs

- Need to manage loading on TPGs
  - Signals degrade completely after 200 PIPs
- Quad BIST structures in large arrays
- Small number of rows with BIST structure across all columns
  - Repeat to fill array
Virtex-4 Logic BIST

- TPGs constructed from DSPs
  - Accumulates constant 0x691
    - Produces pseudo-exhaustive patterns
  - Two TPGs per 4 rows of CLBs
  - Each TPG drives alternating columns of BUTs
- ORAs in alternate columns
  - 2 test sessions needed
- BUTs
  - Logic slices need 10 configs
  - Memory slices need 12 configs
    - Not counting LUT RAMs
    - Includes 2 for testing Shift Registers
  - All slices test concurrently
Reducing Test Time

- Orient BIST architecture to configuration memory
  - Keep routing constant between configurations
- Downloading BIST configurations
  - Partial reconfiguration
    - Reduce # frames written between configurations
      - Keep routing constant between configurations
      - Optimize ordering of BIST configurations
- Retrieving BIST results
  - Partial configuration memory readback
    - Eliminates ORA logic for scan chain
      - Allows concurrent testing of more resources
    - Reduce # frames read
  - Dynamic partial reconfiguration
    - Read BIST results after a series of BIST configurations
      - Slight loss of diagnostic resolution
Reducing Test Time

Optimized Partial Reconfig
- End Partial Mem RB
  - 3x test time speed-up
  - 8x test time speed-up
  - 7x test time speed-up

Partial Reconfig
- Partial Mem RB
  - 3x test time speed-up
  - 8x test time speed-up

Full Config
- End Shift Reg
  - 2x test time speed-up
- Shift Reg
  - Initial Virtex-4 Results:
    - 7x test time speed-up
    - Partial Reconfig
- Full Mem RB
  - 1x test time speed-up

Download Technique

Virtex I Logic BIST Test Time Speed-up/Memory Reduction

Memory Reduction
- 0
- 1
- 2
- 3
- 4
- 5
- 6

Test Time Speed-up

Initial Virtex-4 Results:
- 7x test time speed-up
- Partial Reconfig
- 8x test time speed-up
- Optimized Partial Reconfig

3 sets of BIST

ELECTRICAL AND COMPUTER ENGINEERING
AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING
Programmable Routing Network

- Wire segments of varying length
  - $xN = N$ PLBs in length
    - $N = 1, 2, 4, 6$ are common
  - $xH = \frac{1}{2}$ the array in length
  - $xL = \text{length of full array}$

- Programmable Interconnect Points (PIPs)
  - Also known as Configurable Interconnect Points (CIPs)
  - Transmission gate connects to 2 wire segments
  - Controlled by configuration memory bit
    - $0 = \text{wires disconnected}$
    - $1 = \text{wires connected}$
Programmable Interconnect Points

- **Break-point PIP**
  - Connect or isolate 2 wire segments

- **Cross-point PIP**
  - 2 nets straight through
  - 1 net turns corner and/or fans out

- **Compound cross-point PIP**
  - Collection of 6 break-point PIPs
    - Can route to two isolated signal nets
  - Significant resource in 4000 series

- **Multiplexer PIP**
  - Directional and buffered
  - Main routing resource in Virtex FPGAs
  - Select 1-of-$N$ inputs for output
    - Decoded MUX PIP – $N$ config bits select from $2^N$ inputs
    - Non-decoded MUX PIP – 1 config bit per input
      - Largest $N=37$ in Virtex-4
Routing BIST

- Program PLBs as TPGs and ORAs
  - Like in logic BIST
- Program groups of wires under test
  - Wire segments
  - Programmable Interconnect Points
- Tests partitioned for local and global routing resources
  - Must route through PLBs for local routing
- Fault models
  - Bridging faults and opens in wire segments
  - Line stuck-at faults
    - Shorts to Vdd and Vss
  - PIPs stuck-on and stuck-off
- Test conditions
  - Opposite logic values on wires/PIPs
  - Monitor both logic values
First Routing BIST Approach

- Original thinking - logic BIST will test routing resources
  - Not true (only 55% in ORCA)

- Comparison-based
  - ORAs compare two groups of WUTs
    - Similar to logic BIST

- Try to test as much routing as possible at one time
  - Poor diagnostic resolution
  - Difficult to develop configurations
Second Routing BIST

- Developed during on-line BIST project
  - Testing restricted to routing resources for 2 rows or columns of PLBs
  - Small Self-Test AReas (STARs)
  - Comparison-based BIST
- Applied to off-line BIST
  - Fill FPGA with STARs
  - Tests run concurrently
  - Diagnostic resolution to STAR
- Easier BIST development
  - But more BIST configurations
    27 vs. 48 for ORCA 2C
Other Routing BIST Approaches

- **Parity-based (Sun and Chan)**
  - Xilinx 4000
  - Parity bit routed over fault-free resources
    - What is fault-free until you’ve tested it?

- **Harris and Tessier**
  - Used comparison-based approach
    - Pointed out 2-testing requirement

- **Renovell and Zorian**
  - Minimum test configurations for switch boxes

- **Modified parity-based approach**
Newer Routing BIST

- Comparison-based BIST
  - No good for small PLBs and difficult to route

- Modified parity-based approach
  - N-bit up-counter with even parity, and
  - N-bit down-counter with odd parity
    - Gives opposite logic values for
      - Stuck-on PIPs & bridging faults
  - Parity used as test pattern
    - N+1 wires under test
    - Good for small PLBs

- Make STARs as small as possible

- Latest: cross-coupled parity
Comparing FPGAs

Routing BIST

- Routing resources per PLB
  - 4000XL/XLA has 25% more than ORCA 2C/2CA
  - ORCA 2C/2CA has 48% more than 4000E/Spartan

- Routing BIST configurations
  - 206 for 4000XL/XLA
  - 48 for ORCA 2C/2CA
  - 128 for 4000E/Spartan

Number and size of multiplexer PIPs

- N=5 for ORCA 2C multiplexer PIPs
- N=35 for 4000XL/XLA multiplexer PIPs

**Bad News:** more & larger MUX PIPs in new FPGAs
- Even more routing BIST configurations
Comparing Routing Architectures

- PLB input/output access to busses
  - More difficulty routing to/from wires = more configs
- Shared vs. dedicated busses to each PLB
  - Routing conflicts from TPGs to ORAs = more configs
Routing Diagnostic Configurations

- **Partition into smaller STARs**
  - Identify faulty region of WUT

- **Add ORAs & change directions**
  - Identify fault region of WUT

- **Re-route portions of net**
  - Identify faulty wire segment or PIP

Diagram:

- TPG
- ORA
- Single wire

Red crosses indicate failed configurations.
Results for Actual Faulty FPGAs

ORCA 2C15A that fails manufacturing tests

Diagnosis of Chip 1

- Single fault
- Location: row 10 column 8
- Short to Vdd

![Diagram showing fault location](image)
Results for Actual Faulty FPGAs

Diagnosis of Chip 2

- **Fault #1**
  - Location: row 5, columns 6-8
  - Short between 3 wires in 4-wire bus

- **Fault #2**
  - Location: row 1, column 12
  - Short to Vdd
Virtex-4 Routing BIST

- **6-bit parity-based BIST architecture**
  - count-up/even parity (3-bits)
  - count-down/odd parity (3-bits)

- **Opposite logic values**
  - Bridging faults
  - PIPs stuck-on

- **BIST logic**
  - Algorithmic in XDL
  - Initial development
    - For local routing

- **Wires under test**
  - Develop router
    - Similar to prior work

**Count-up/Even** | **Count-down/Odd**
<table>
<thead>
<tr>
<th>Par C1 C0</th>
<th>Par C1 C0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>1 1 1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>0 1 1</td>
<td>1 0 0</td>
</tr>
</tbody>
</table>