



Exploiting FPGA's Configurability for Testing and Tolerance

Erik Chmelar
Center for Reliable Computing
Stanford University
July 22, 2002

Outline

- Introduction
- **FPGA Architecture**
- Testing
- Defect Tolerance
- Fault Tolerance
- Conclusion

Outline

- **Introduction**
- FPGA Architecture
- Testing
- Defect Tolerance
- Fault Tolerance
- Conclusion

FPGA Architecture

- Basic Building Blocks
 - Configurable Logic Blocks (CLBs)
 - Input/Output Blocks (IOBs)
 - Programmable Switch Matrices (PSMs)
- Routing Resources: > 50% of die area
 - Wires
 - Programmable Interconnect Points (PIPs)

Introduction Why FPGAs?

FPGA

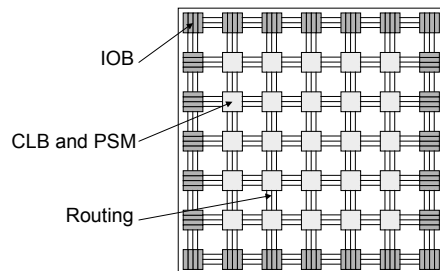
- Field (re)programmable
- Highly flexible
- No *NRE* costs
- Short development time

Standard Cell ASIC

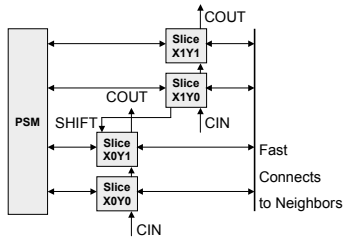
- Application specific
- Very inflexible
- High *NRE* costs
- Long development time

[Schrek 99]

Basic Building Blocks



Configurable Logic Block (1) Xilinx Virtex-II CLB



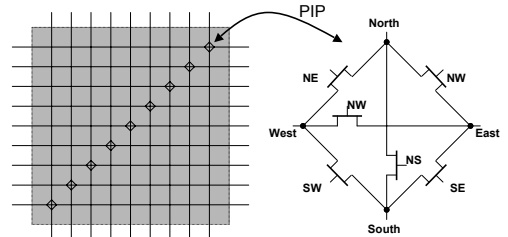
[Xilinx 01]

07/22/02

(c) 2002 Center for Reliable Computing

7

Programmable Switch Matrix



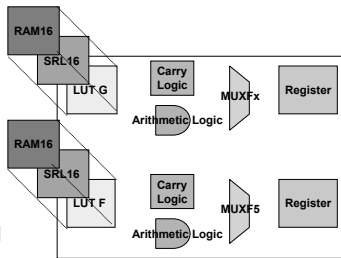
Six possible connections, NS, NE, NW, SW, SE, WE

07/22/02

(c) 2002 Center for Reliable Computing

10

Configurable Logic Block (2) Xilinx Virtex-II Slice



[Xilinx 01]

07/22/02

(c) 2002 Center for Reliable Computing

8

Wires

Xilinx Virtex-II Wires

Horizontal Long Lines	Span entire height/width of device	
Vertical Long Lines		
Horizontal Hex Lines	Connects third or sixth CLB in all four directions	
Vertical Hex Lines		
Horizontal Double Lines	Connects first or second CLB in all four directions	
Vertical Double Lines		
Direct Connects	Connect neighboring vertical, horizontal, and diagonal CLBs	
Fast Connects	Internal to a CLB	

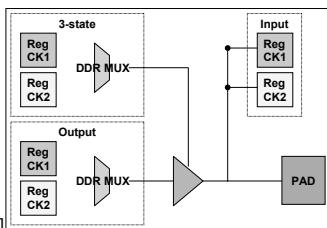
[Xilinx 01]

07/22/02

(c) 2002 Center for Reliable Computing

11

Input/Output Block Xilinx Virtex-II IOB



[Xilinx 01]

07/22/02

(c) 2002 Center for Reliable Computing

9

Programmable Interconnect Point

- Cross-point PIP – multiple pass transistors
– Connects wires in disjoint routing planes
- Break-point PIP – single pass transistor
– Connects wires in same routing plane
- Multiplexer PIP – group of cross-point PIPs
– Connects one of many possible wires

07/22/02

(c) 2002 Center for Reliable Computing

12

Outline

- Introduction
- FPGA Architecture
- **Testing**
- Defect Tolerance
- Fault Tolerance
- Conclusion

General Test Methodology

Good Qualities

- Scalable, independent of array size
- Re-usable, thus automatable
- Small amounts of tester memory
- Few programming phases
- Simple test pattern generation
- High fault coverage for logic and routing

[Toutouchi et al. 02] [Huang et al. 98] [Huang et al. 96]

Testing (1)

- General Test Methodology
- Testing Logic Blocks
- Testing Routing Resources

Standard Testing Practices (1)

Test procedure	Test overhead
Configure FPGA with some design	Time consuming, Configuration simulation ATPG, ATE, Patterns not applied at-speed Fault simulation
Apply test vectors	
Analyze response	

Iterate

[Stroud 96]

Testing (2)

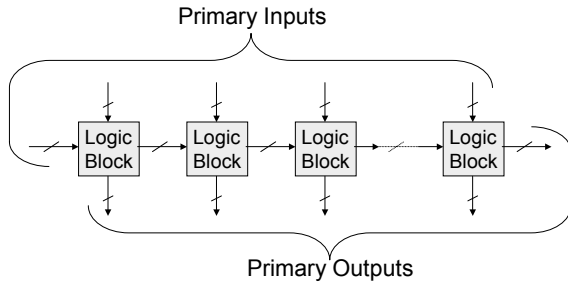
- Testing Logic Blocks
 - Standard Testing Practices
 - Iterative Logic Arrays
 - Built-in Self Test
- Testing Routing Resources
 - Bus-based Testing
 - Delay Fault Testing

Standard Testing Practices (2)

Assessment

- | | |
|--|----|
| • Scalable, independent of array size | No |
| • Re-usable, thus automatable | No |
| • Small amounts of tester memory | No |
| • Few programming phases | No |
| • Simple test pattern generation | No |
| • High fault coverage, logic and routing | No |

Iterative Logic Arrays (1)



07/22/02

(c) 2002 Center for Reliable Computing

19

Built-in Self Test (1)

Procedure

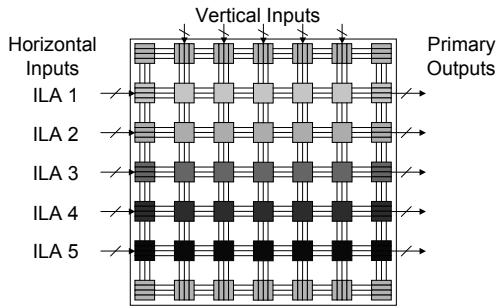
- Configuration
 - Block under test (BUT)
 - Test pattern generator (TPG)
 - Output response analyzer (ORA)
 - Test pattern generation – TPG
 - Test pattern application – TPG, BUT
 - Response analysis – BUT, ORA
- [Stroud 96]

07/22/02

(c) 2002 Center for Reliable Computing

22

Iterative Logic Arrays (2)

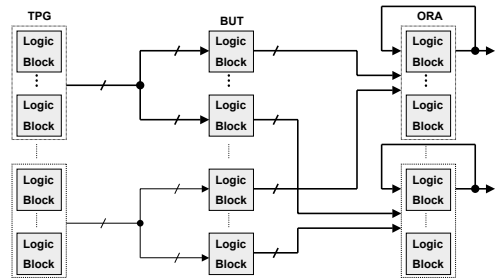


07/22/02

(c) 2002 Center for Reliable Computing

20

Built-in Self Test (2)



07/22/02

(c) 2002 Center for Reliable Computing

23

Iterative Logic Arrays (3)

Assessment

- | | |
|--|-----------|
| • Scalable, independent of array size | Yes |
| • Re-usable, thus automatable | Yes |
| • Small amounts of tester memory | Yes, BIST |
| • Few programming phases | No |
| • Simple test pattern generation | Neutral |
| • High fault coverage, logic and routing | Neutral |

07/22/02

(c) 2002 Center for Reliable Computing

21

Built-in Self Test (3)

Assessment

- | | |
|--|------------|
| • Scalable, independent of array size | Yes |
| • Re-usable, thus automatable | Yes |
| • Small amounts of tester memory | Yes |
| • Few programming phases | Yes |
| • Simple test pattern generation | Yes |
| • High fault coverage, logic and routing | Yes, logic |

07/22/02

(c) 2002 Center for Reliable Computing

24

Bus-based Testing (1)

Routing Resources: PSM

- FPGA Configuration
 - Three PSM configurations needed
- Test pattern application
- Response analysis

[Sun 00]

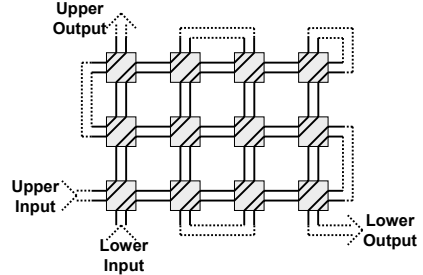
07/22/02

(c) 2002 Center for Reliable Computing

25

Bus-based Testing (4)

Example 2: Left-diagonal



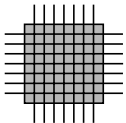
07/22/02

(c) 2002 Center for Reliable Computing

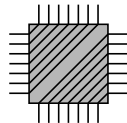
28

Bus-based Testing (2)

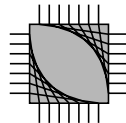
PSM Configurations



Orthogonal
NS, WE



Left-diagonal
NW, SE



Right-diagonal
NE, SW

07/22/02

(c) 2002 Center for Reliable Computing

26

Bus-based Testing (5)

Assessment

- | | |
|--|-----------|
| • Scalable, independent of array size | Yes |
| • Re-usable, thus automatable | Yes |
| • Small amounts of tester memory | Yes, BIST |
| • Few programming phases | Yes |
| • Simple test pattern generation | Yes |
| • High fault coverage, logic and routing | Yes, PSM |

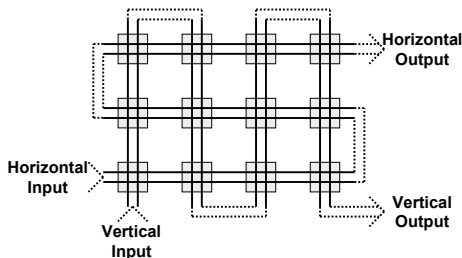
07/22/02

(c) 2002 Center for Reliable Computing

29

Bus-based Testing (3)

Example 1: Orthogonal



07/22/02

(c) 2002 Center for Reliable Computing

27

Delay Fault Testing (1)

Routing Resources: Wires

- Targets resistive opens
 - Discontinuity in circuit path
 - 58% returned parts suspect [Needham 98]
- (Circuit) Delay
 - Proportional resistance and capacitance
- Delay ratio $\frac{\text{defective circuit delay}}{\text{equivalent defect-free circuit delay}}$

[Tahoori 02]

07/22/02

(c) 2002 Center for Reliable Computing

30

Delay Fault Testing (2) Procedure

- FPGA Configuration
 - Disjoint linear paths
 - Only transparent logic – flip flops, inverters
- Path loading
 - Add fanout branches to paths
- Test pattern application
- Response analysis

07/22/02

(c) 2002 Center for Reliable Computing

31

Delay Fault Testing (5) Assessment

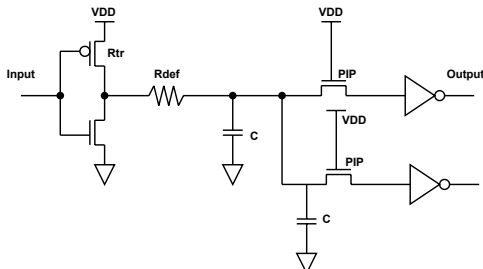
- | | |
|--|------------|
| • Scalable, independent of array size | No |
| • Re-usable, thus automatable | Yes |
| • Small amounts of tester memory | Yes |
| • Few programming phases | No |
| • Simple test pattern generation | Yes |
| • High fault coverage, logic and routing | Yes, wires |

07/22/02

(c) 2002 Center for Reliable Computing

34

Delay Fault Testing (3) Delay ~ $[R_{tr}(V_{DD}) + R_{def}] \times C_{total}$



07/22/02

(c) 2002 Center for Reliable Computing

32

Outline

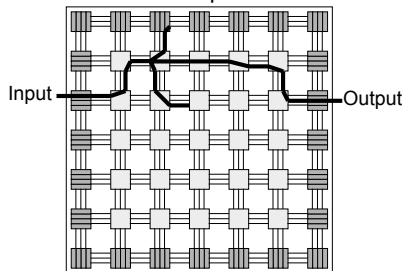
- Introduction
- FPGA Architecture
- Testing
- **Defect Tolerance**
- Fault Tolerance
- Conclusion

07/22/02

(c) 2002 Center for Reliable Computing

35

Delay Fault Testing (4) Example



07/22/02

(c) 2002 Center for Reliable Computing

33

Defect Tolerance

- Application-specific Testing

07/22/02

(c) 2002 Center for Reliable Computing

36

Application-specific Testing (1)

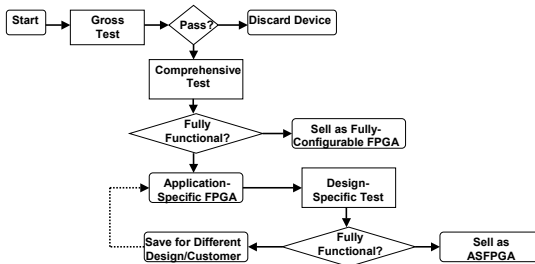
Overview

- Use known faulty FPGA
 - Avoid faulty resources
- Develop tests based on customer design
- Test FPGA functionality with special test
- Sell FPGA to customer at reduced price
- Effectively increases yield

Fault Tolerance

- Roving STARs
- Torus Configuration Data Shifting

Application-specific Testing (2)



Roving STARs (1)

Overview

- Find faulty block(s), reconfigure FPGA online
 - Self-testing area (STAR)
 - portion being tested
 - Working area
 - portion implementing normal function
- FPGA remains online

[Stroud 01]

Outline

- Introduction
- FPGA Architecture
- Testing
- Defect Tolerance
- **Fault Tolerance**
- Conclusion

Roving STARs (2)

Requirements

- Runtime reconfigurable (RTR) FPGA
- Spare resources
- Memory to store faulty/not faulty information
- Memory to store precompiled configurations
- Test and reconfigurability controller (TREC)

Roving STARs (3) Procedure

- Two STARs
 - V-STAR – FPGA column
 - H-STAR – FPGA row
- STARs rove about the FPGA
 - Move to equally-sized areas
 - Configure area for BIST
 - Test the area

07/22/02

(c) 2002 Center for Reliable Computing

43

Torus Configuration Data Shifting (1) Overview

- Reconfigure FPGA around faulty block(s)
 - Identical unit elements (UEs)
 - Shifts configuration data of UEs
 - north, south, east, or west

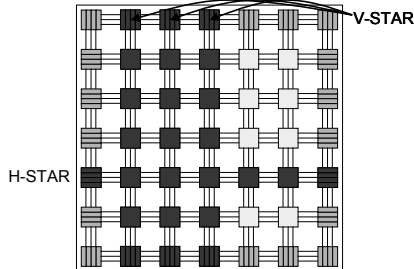
[Doumar 01]

07/22/02

(c) 2002 Center for Reliable Computing

46

Roving STARs (4) Example



07/22/02

(c) 2002 Center for Reliable Computing

44

Torus Configuration Data Shifting (2) Requirements

- Modification to standard FPGA structure
 - One 4-input MUX for each memory cell
- Spare UEs
- Memory to store faulty/not faulty information
- Test and reconfigurability controller (TREC)

07/22/02

(c) 2002 Center for Reliable Computing

47

Roving STARs (5) Issues

- System stalls 3-4 seconds before STAR moves
- Takes long time for STARs to completely rove FPGA
 - 6 minutes to rove 20x20 ORCA 2C15A FPGA

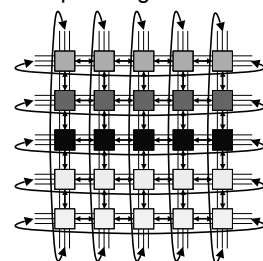
[Emmert 00]

07/22/02

(c) 2002 Center for Reliable Computing

45

Torus Configuration Data Shifting (3) Example: Single Shift North



07/22/02

(c) 2002 Center for Reliable Computing

48

Outline

- Introduction
- FPGA Architecture
- Testing
- Defect Tolerance
- Fault Tolerance
- **Conclusion**

07/22/02

(c) 2002 Center for Reliable Computing

49

References (2)

- [Needham 98] Needham, Wayne C. and Prunty, E. H. Yeow. "High Volume Microprocessor Test Escapes, an Analysis of Defects out Tests are Missing," International Test Conference Proceedings, 25-34, 1998.
- [Schrek 99] Schrek, Rob. "New ASIC Estimator—For Cost Modeling," Xcell Journal, Q1 1999.
- [Stroud 96] Stroud, Charles, et al. "Built-in Self-test of Logic Blocks in FPGAs," Fourteenth VLSI Test Symposium, 1996.
- [Stroud 01] Stroud, Charles et al. "Online BIST and Diagnosis of FPGA Interconnect Using Roving STARS," Seventh International On-line Testing Workshop Proceedings, 27-33, 2001.

07/22/02

(c) 2002 Center for Reliable Computing

52

Conclusion

- Exploit FPGA reconfigurability
 - For testing
 - Reconfigure sparingly – BIST ILAs
 - For defect tolerance
 - Reconfigure until defect is avoided
 - For fault tolerance
 - Reconfigure until fault is avoided

07/22/02

(c) 2002 Center for Reliable Computing

50

References (3)

- [Sun 00] Sun, Xiaoling, et al. "Novel Technique for Built-in Self-test of FPGA Interconnects," International Test Conference Proceedings, 795-803, 2000.
- [Tahoori 02] Tahoori, Mehdi Baradaran and McCluskey, E. J. "Testing for Resistive Opens in FPGA." Unpublished as of May 2002, 2002.
- [Toutouchi 02] Toutouchi, Shahin and Andrew Lai. "FPGA Test and Coverage," Unpublished as of April 2002, Xilinx, Inc., 2002.
- [Xilinx 01] Xilinx, Inc., San Jose, Ca. "Virtex-II Platform FPGA Handbook," December 2001.

07/22/02

(c) 2002 Center for Reliable Computing

53

References (1)

- [Doumar 01] Doumar, Abderrahim and Ito, Hideo. "FPGAs and Fault Tolerance," Thirteenth International Conference on Microelectronics, 222-225, 2001.
- [Emmert 00] Emmert, J. et al. "Dynamic Fault Tolerance in FPGAs via Partial Reconfiguration," Proceedings of the IEEE Field-Programmable Custom Computing Machines, 165-174, 2000.
- [Huang 96] Huang, W. K. and Lombardi, F. "An approach for Testing Programmable/Configurable Field Programmable Gate Arrays," Fourteenth VLSI Test Symposium, 1996.
- [Huang 98] Huang, W. K. and Lombardi, F. "Testing Configurable LUT-based FPGAs," IEEE Transactions on VLSI Systems, 6(2):276-283, June 1998.

07/22/02

(c) 2002 Center for Reliable Computing

51