

Minimizing the Number of Test Configurations for FPGAs*

Erik Chmelar
LSI Logic
November 10, 2004

*Work done at the Center for Reliable Computing at Stanford University.

11/10/04 1

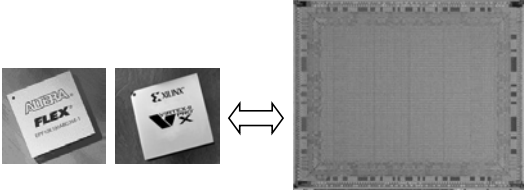
Outline

- **FPGA Architecture**
- Background
- Test Configuration Minimization
- Conclusion

11/10/04 2

Field-programmable Gate Array

- (Re)configurable integrated circuit
 - Implement arbitrary logic design
- Low non-recurring engineering costs
- Fast time to market



11/10/04 3

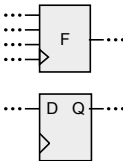
Building Blocks

- Logic Block (LB)
 - Combinational and sequential logic
- Input/Output Block (IOB)
 - Chip input and output
- Switch Matrix (SM)
 - Conditionally joins interconnects
 - Programmable Interconnect Points (PIPs)

11/10/04 4

Logic Block

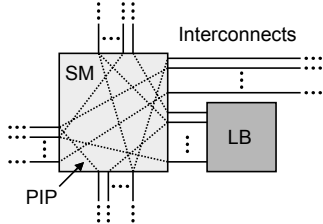
- Look-up Tables (LUTs)
 - Boolean function truth table
- Bistables
 - Flip-flop or latch



11/10/04 5

Switch Matrix

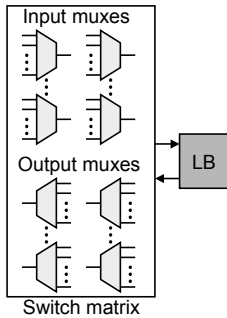
- Joins interconnects to
 - Logic Blocks
 - Other interconnects



11/10/04 6

Switch Matrix

- Collection of muxes
 - Input muxes
 - Output muxes
- Various sizes
 - Virtex-II, Spartan-3
 - Largest is 32:1



11/10/04

7

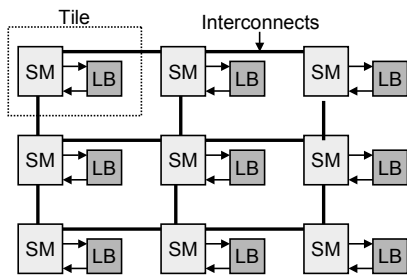
Routing Network

- Wire segments
 - Several lengths
 - Grouped into busses
- Programmable Interconnect Points
 - Pass transistor
 - Controlled by SRAM cell
- Buffers
- Vias

11/10/04

8

Tiled Layout



11/10/04

9

Outline

- FPGA Architecture
- **Background**
- Test Configuration Minimization
- Conclusion

11/10/04

10

FPGA Test

- Billions of possible configurations
 - How to test them all?
 - Create special test configurations
 - Hundreds used in practice
- Test process

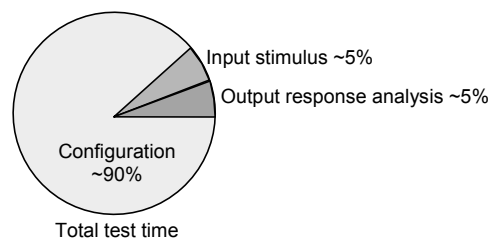

```
do {
  program FPGA with test configuration
  apply test vectors
  compare output to expected value
} until all test configurations used
```

11/10/04

11

Test Bottleneck

- Configuration time
 - Dominates total test time



11/10/04

12

Previous Work (1)

- Minimize number of test configurations
 - Represent FPGA as graph
 - Traverse vertices and edges of graph
 - Maximum flow
- Want 100% fault coverage
 - Stuck-at 0/1, PIP stuck-on/off

11/10/04

13

Previous Work (2)

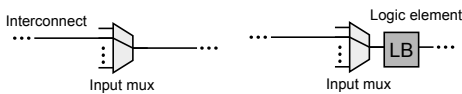
- Graph traversal algorithms
 - Virtex architecture
- [Tahoori and Mitra VTS 2003]
 - 8 configurations
 - Not all PIPs considered
 - Input and output muxes not considered
- [Fernandes and Harris ITC 2003]
 - 59 configurations
 - Considers larger subset of routing resources

11/10/04

14

Mux Perspective

- To test all
 - Interconnects
 - Logic element inputs and outputs
 - Signal must traverse input mux
- Necessary and sufficient to test all
 - Input mux inputs and outputs

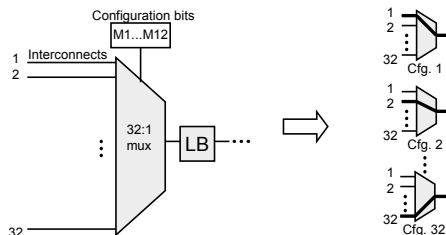


11/10/04

15

Limitation

- At most one path through mux
 - Minimum N configurations for N -input mux
 - 32:1 mux \rightarrow 32 configurations



11/10/04

16

Outline

- FPGA Architecture
- Background
- **Test Configuration Minimization**
- Conclusion

11/10/04

17

Solution

- Test
 - Multiple paths through mux per configuration
 - Intentionally short mux inputs
- Reduce number of configurations
 - Less than N
- Configuration generation addressed at two levels
 - Mux-level
 - Chip-level

11/10/04

18

Mux-level Configuration

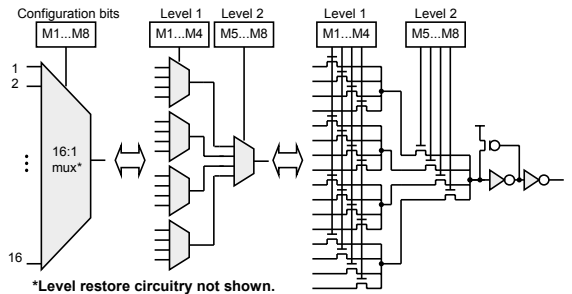
- Detect all stuck-at 0/1, PIP stuck-on/off faults
- Configure mux
 - Short several mux inputs
- Drive shorted inputs
 - Mux is NMOS pass transistors
 - Strong logic-0
 - Weak logic-1

11/10/04

19

Mux Implementation

- Two-level NMOS pass transistors



11/10/04

Logic element 20

Minimum Configurations

- How to detect stuck-at 0/1 and PIP stuck-on/off?
 - 1 logic-0 input overpowers 4 logic-1 inputs
 - SPICE simulations (90 nm, 1.2 V)
 - Verified on FPGA
- Condition mux inputs
 - Create pull-down path
 - Output transition or absence of transition

11/10/04

21

Stuck-at 0 Detection

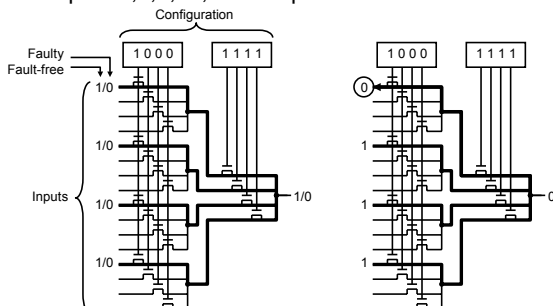
- Short 4 inputs
 - Drive each to logic-1
 - Stuck-at 0 overpowers other logic-1 inputs
 - Pulls mux output to logic-0

11/10/04

22

Stuck-at 0 Example

- Inputs 1,5,9,13, and output tested



11/10/04

23

Stuck-at 1 Detection

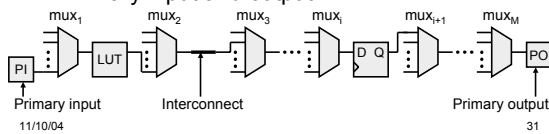
- Short 4 inputs
 - Drive 3 to logic-1, 1 to logic-0
 - Stuck-at 1 on logic-0 input
 - Mux output remains at logic-1

11/10/04

24

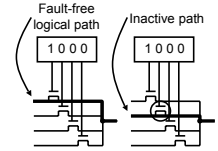
Chip-level Configuration

- Control and observe all mux inputs and outputs
- Join muxes to form Iterative Logic Arrays
 - Interconnects
 - Bistables
 - Look-up Tables (LUTs)
- ILA input and output
 - Primary input and output



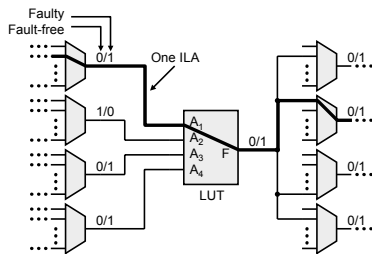
ILA Routing

- Fault-free logical path (FFLP) through mux
 - Path from input to output
 - Through turned-on PIPs
 - No PIP fault present
- Route output of mux_i
 - To FFLP path of mux_{i+1}
- Program bistable to be transparent (D flip-flop)
- Program LUT to propagate fault
 - $F=f(A1,A2,A3,A4)=f_{fault-free}$ if no fault present
 - $F=f_{fault-free}$ if at least one fault present



ILA with LUT Example

- $F=(A1'A2'A3'A4) '=0$



Outline

- FPGA Architecture
- Background
- Test Configuration Minimization
- **Conclusion**

Conclusion

- Configuration time dominates FPGA test time
- Minimum number of configurations
 - Limited by size of largest mux
 - One path through mux per configuration
 - 32:1 muxes in Spartan-3, Virtex-II
- Solution
 - Enable multiple paths through mux
 - 8 configurations
 - Detect all stuck-at 0/1, PIP stuck on/off