

Embedded Processor Based Fault Injection and SEU Emulation for FPGAs

Bradley Dutton, Mustafa Ali, John
Sunwoo, and Charles Stroud

BIAT Auburn
University
Built-In Self-Test



**ELECTRICAL
AND COMPUTER ENGINEERING**
AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Outline

□ Background

❖ What is Built-in Self-test for FPGAs?

- ✓ Example: BIST for Virtex-5 Configurable Logic Blocks

□ Motivation

❖ What is FPGA fault emulation?

❖ Why do we need it?

□ Two case studies of embedded processors used for fault injection

❖ Hard processor-based fault injection

- ✓ Atmel AT94K SoC

❖ Soft processor-based fault injection and Single-Event Upset (SEU) emulation

- ✓ Xilinx Virtex-4 and Virtex-5 FPGAs

□ Conclusions

BIST for FPGAs

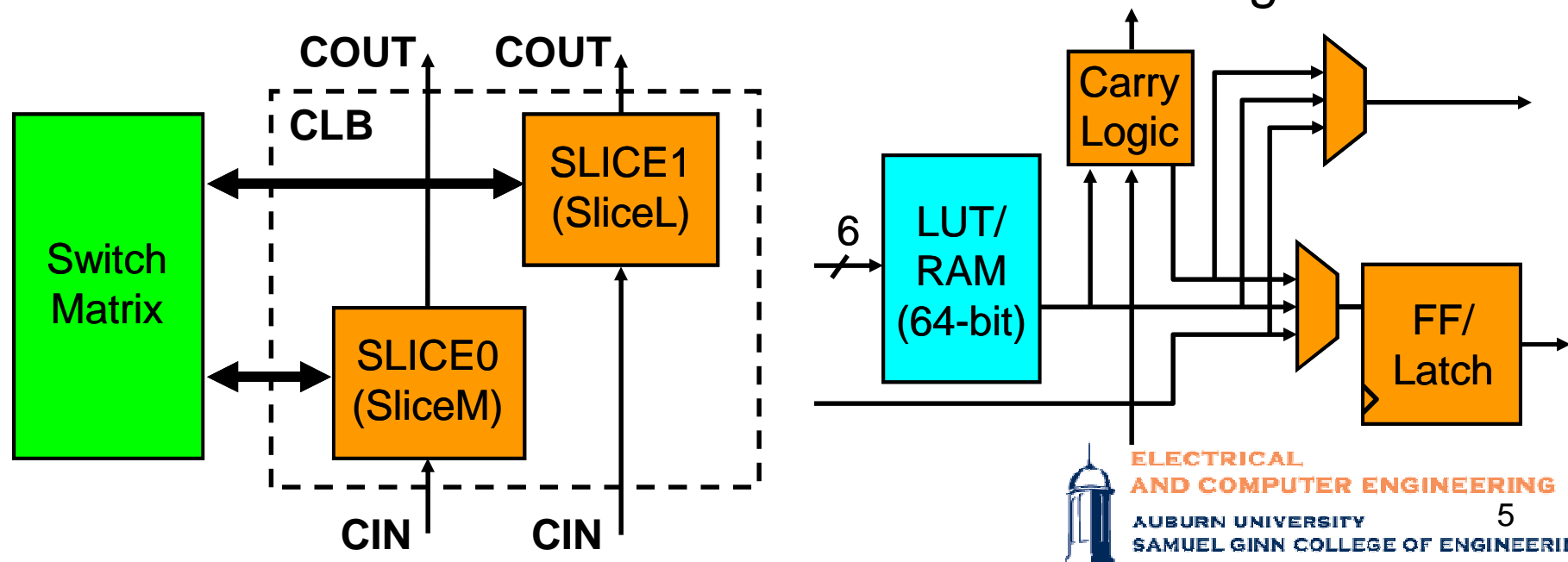
- **Basic idea:** reprogram FPGA to test itself
 - ❖ No area overhead or performance penalties
- Applicable to all levels of testing
 - ❖ Application independent testing
 - ✓ A generic test approach for a generic component
 - ❖ Good diagnostic resolution
- **Cost:**
 - ❖ Memory to store BIST configurations
 - ✓ **Goal:** minimize number and size of configurations
- Test time = download + execute + results
 - ❖ Dominated by download time
 - ✓ **Goal:** minimize downloads and/or download time
 - ❖ Results retrieval is second

AUBIST Approach

- Configure some logic resources to act as
 - ❖ Test Pattern Generators (TPGs)
 - ❖ Output Response Analyzers (ORAs)
- Configure other resources to be tested
 - ❖ Blocks Under Test (BUTs)
 - ❖ Wires Under Test (WUTs)
- For all configurations, maintain constant
 - ❖ placement of TPGs, ORAs, & BUTs
 - ❖ routing of TPG-to-BUT & BUT-to-ORA
 - ✓ minimizes download time via partial reconfiguration
- Automatic generation of BIST configurations
 - ❖ For any size device in FPGA family

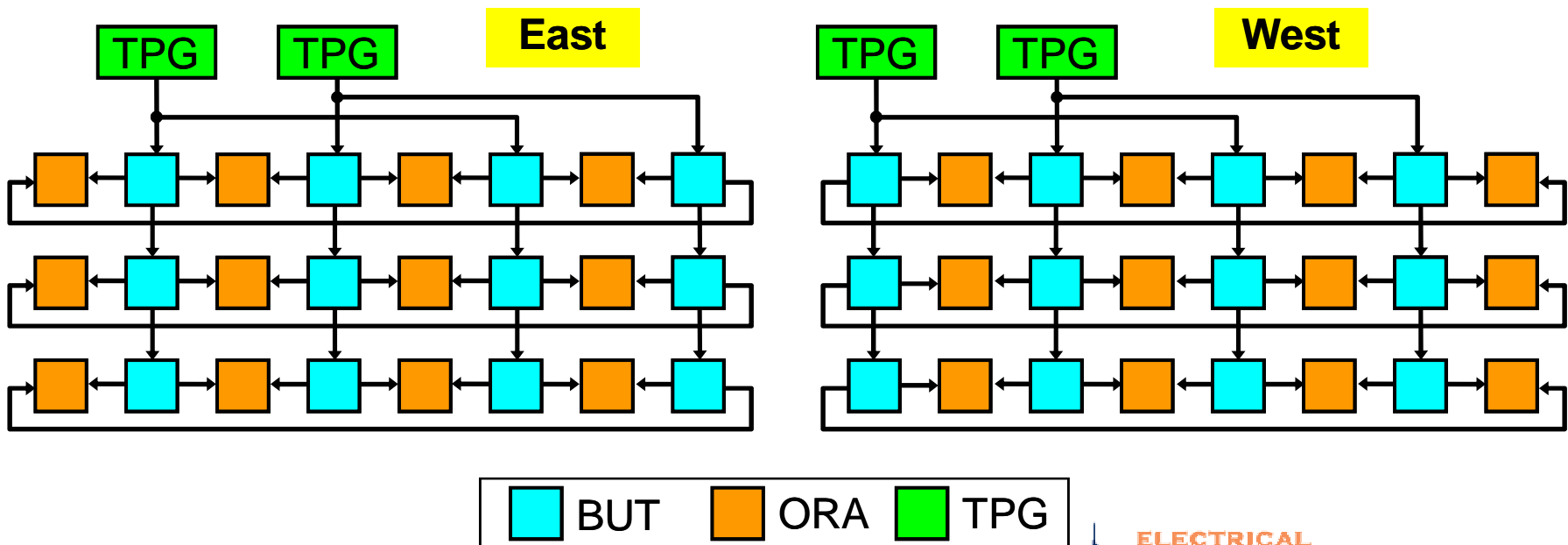
V-5 Configurable Logic Block (CLB) BIST

- CLB is most abundant logic resource
 - ❖ 25,920 CLBs in Largest Virtex-5
 - ✓ 207,360 FFs and 6-input LUTs
 - ❖ 652 configuration bits per CLB
 - ❖ Some CLBs include SliceMs (LUT RAMs)
 - ✓ SliceM can form small RAMs or Shift Register



SliceL BIST Architecture

- Two test sessions
 - ❖ East & West
 - ❖ Every CLB configured as both a Block Under Test (BUT) and Output Response Analyzer (ORA)
- Multiple test phases per test session
 - ❖ Test all modes of operation in CLB



03/17/2009

SSST

SliceL BIST Architecture

□ Test Pattern Generator

- ❖ 12 inputs to each basic logic element
- ❖ DSP configured as accumulator generates an exhaustive set of patterns
- ❖ Accumulate prime number 0xCA6691 [1]
 - ✓ Produces 2^{12} patterns in 2^{12} clock cycles with high number of transitions in most significant bits

□ Multiple TPGs connect to alternating columns of BUTs

- ❖ Eliminates fault-free TPG assumption

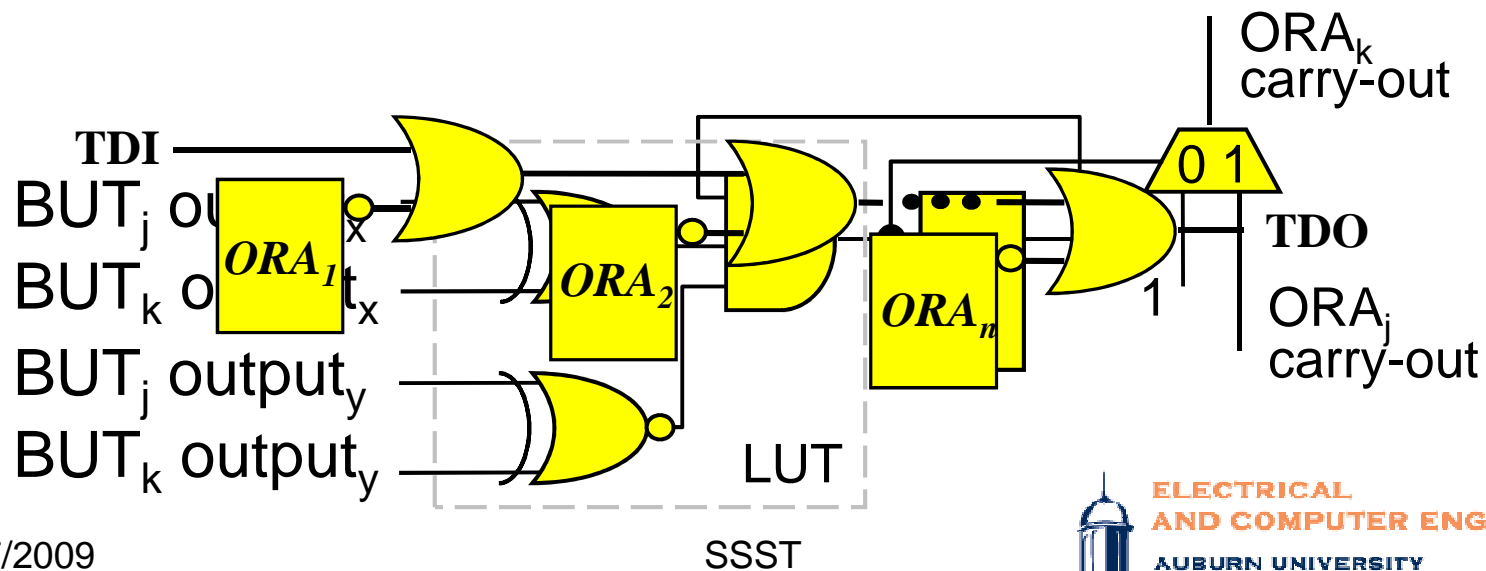
□ Comparison-based output response analysis

- ❖ Compare the outputs of two adjacent, identically configured CLBs
- ❖ Row based circular comparison

[1] S. Gupta, J. Rajski, and J. Tyszer, "Test pattern generation based on arithmetic operations," Proc. IEEE Int. Conf. on Computer-Aided Design, pp. 117-124, 1994.

Iterative-OR output response analyzer

- Each ORA compares two outputs of BUT
 - ❖ Initialized to logic 1
 - ❖ Any mismatch will latch a logic 0
- Results retrieved
 - ❖ Partial configuration memory read back
 - ✓ High diagnostic resolution when fault(s) detected
 - ❖ Via single-bit iterative-OR chain output



03/17/2009

SliceM BIST Architecture

□ Block RAM TPGs store RAM test vectors

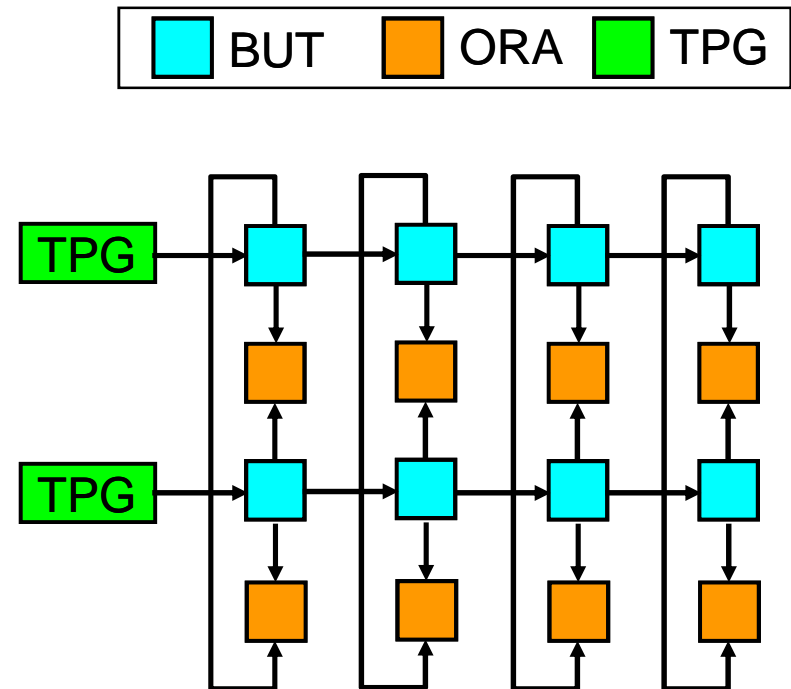
- ❖ March Y + Dual-port March [2]
- ❖ 2048 x 18-bit BRAM , $8N = 8 \cdot 256 = 2048$ vectors

□ Iterative-OR chain ORA

- ❖ Column based circular comparison

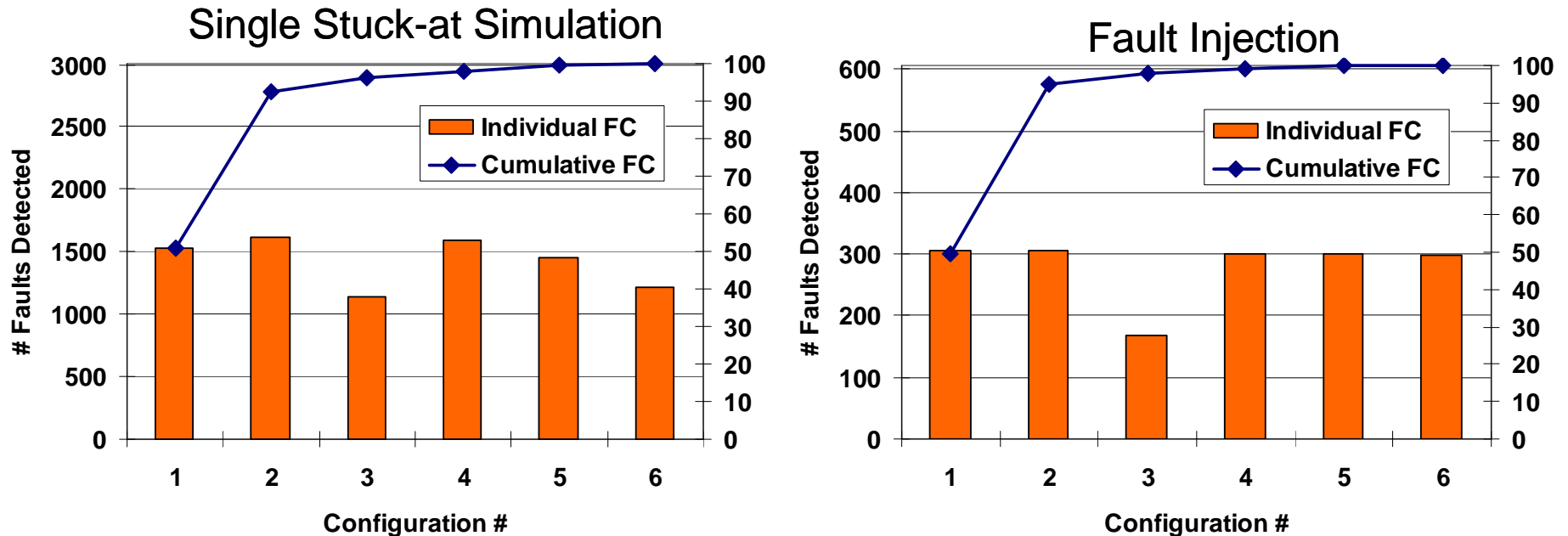
□ One multiple phase test session for all SliceMs

- ❖ Every CLB with a SLICEM has a SLICEL for ORA



[2] A. van de Goor, *Testing Semiconductor Memories: Theory and Practice*, John Wiley and Sons, 1991.

V-5 SliceL BIST Fault Coverage



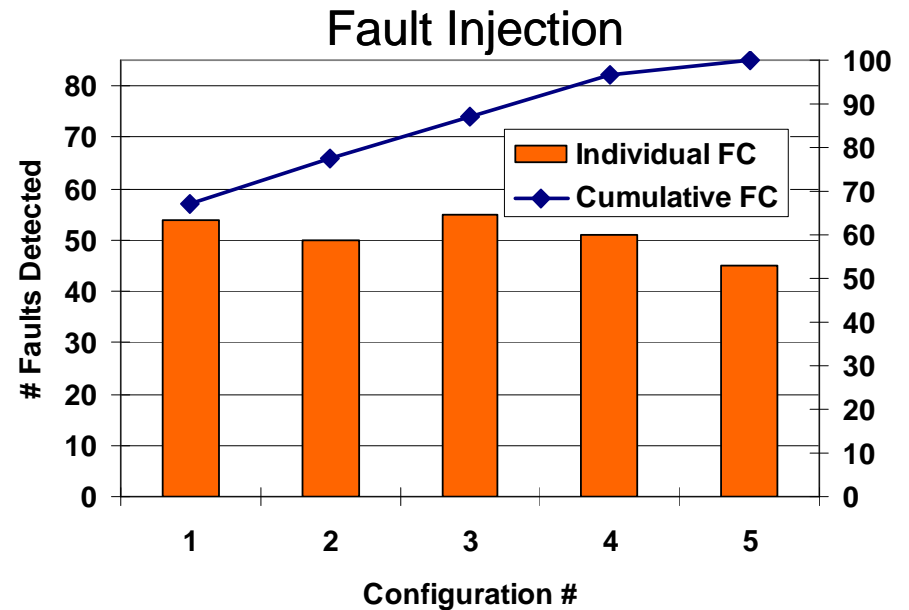
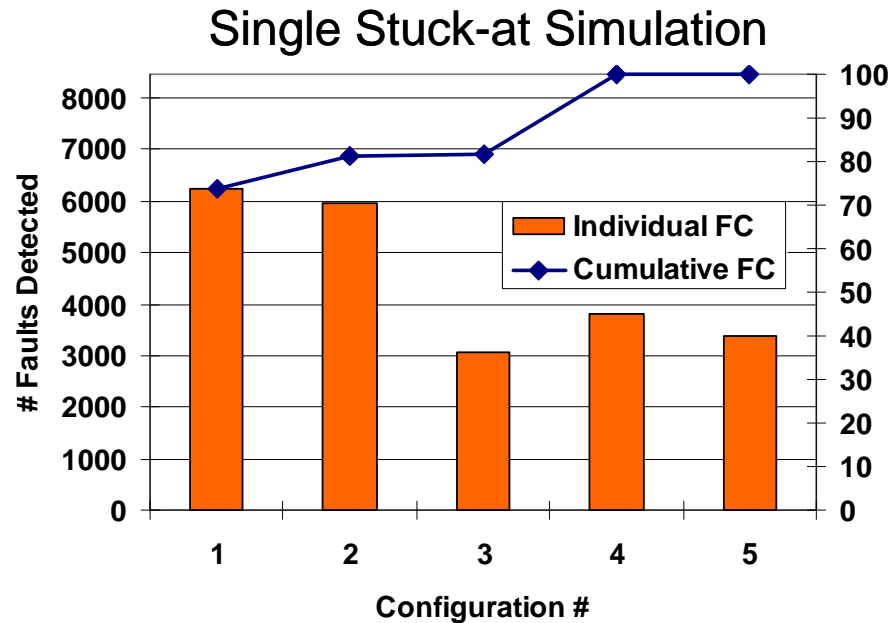
□ Gate-level model (AUSIM)

- ❖ 3008 gate-level collapsed stuck-at faults
- ❖ 100% cumulative coverage in 6 phases w/ DSP TPG

□ Configuration memory fault injection

- ❖ 614 configuration bit stuck-at faults
- ❖ 100% cumulative coverage

V-5 SliceM BIST Fault Coverage



□ Gate-level model (AUSIM)

- ❖ 8462 gate-level collapsed stuck-at faults
- ❖ 100% cumulative coverage in 5 phases w/ RAM tests

□ Configuration memory fault injection

- ❖ 85 configuration bit stuck-at faults
- ❖ 100% cumulative coverage

Motivation for Fault Injection

- Why do we need fault injection?
 - ❖ Actual faulty parts are difficult to obtain
 - ✓ Not very useful because faults are fixed
 - ❖ Fault injection allows emulation of any stuck-at fault in the configuration memory
 - ✓ Good indication of overall fault coverage
 - ❖ Does not permanently damage the device
 - ❖ No additional overhead when verifying BIST
- Limitation: low speed external configuration interface
 - ❖ Embedded core can manipulate configuration memory at higher speed

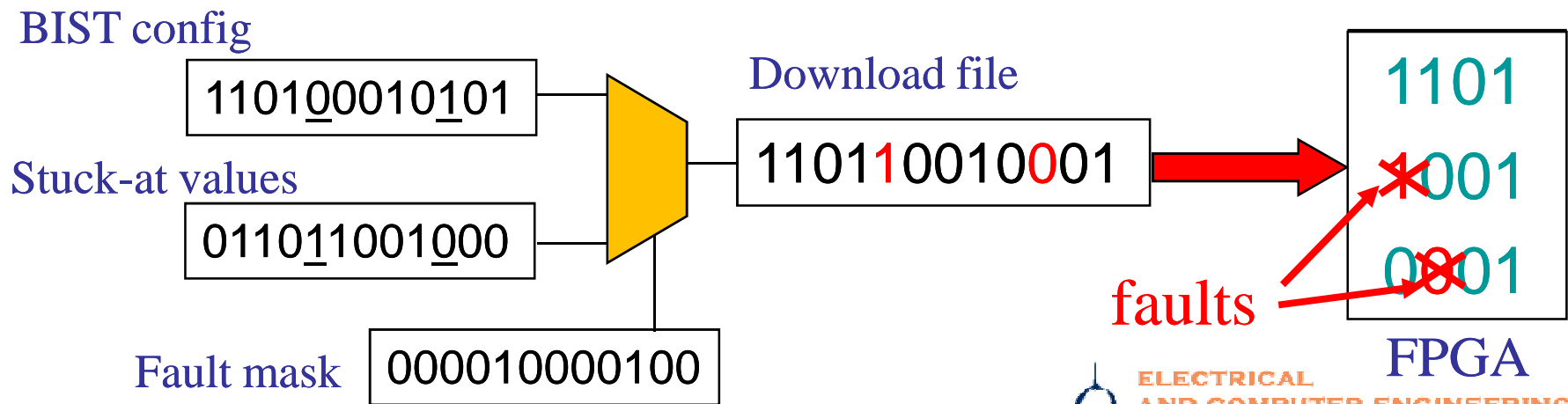
Physical Fault Injection

Physical fault insertion

- ❖ Etch package and damage device with laser

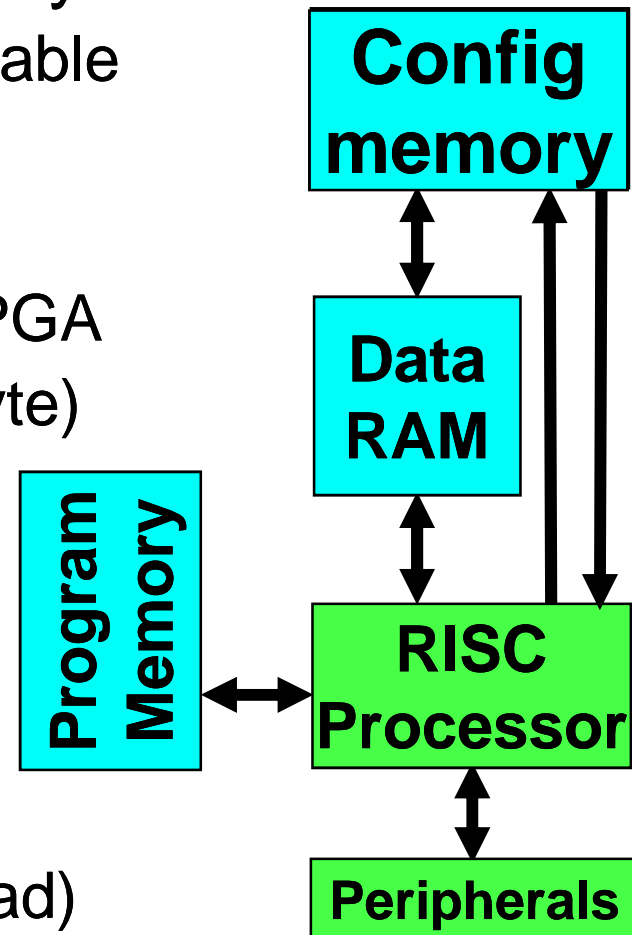
Fault Injection Emulation

- ❖ Modify configuration memory bits
- ❖ Fault Emulator can create single & multiple faults in:
 - ✓ PLBs: LUTs, flip-flops, etc.
 - ✓ Interconnect: PIPs stuck-on & stuck-off



Atmel AT94K FPSLIC Architecture

- ❑ Field Programmable Gate Array
 - ❖ up to 48x48 array of Programmable Logic Blocks (PLBs)
- ❑ RAM cores
 - ❖ 32x4 bit RAMs distributed in FPGA
 - ❖ Program memory (up to 32 Kbyte)
 - ✓ Single port to processor
 - ❖ Data RAM (up to 16 Kbyte)
 - ✓ Dual port to FPGA & processor
- ❑ 8-bit RISC processor core
 - ❖ Various peripherals
 - ❖ Processor can write (but not read) FPGA configuration memory



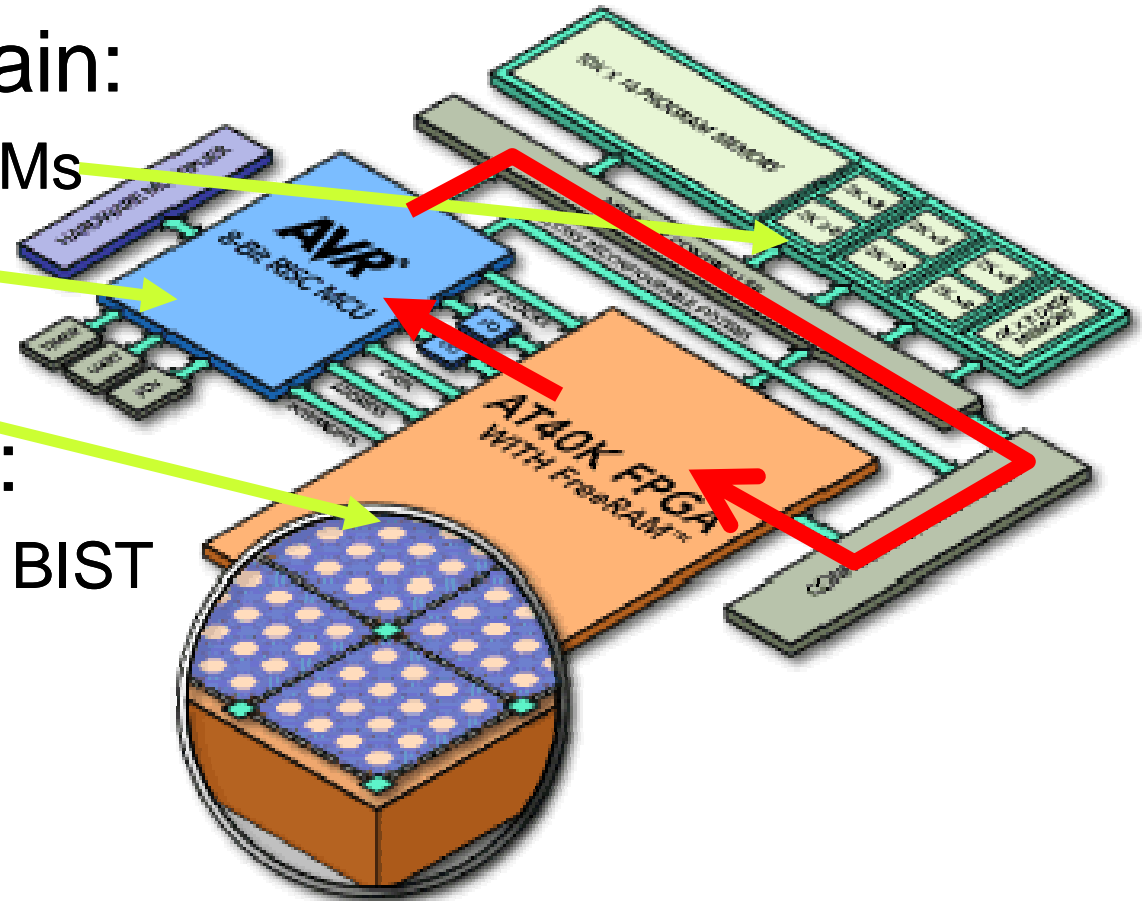
On-Chip BIST and Diagnosis

□ Atmel SoCs contain:

- ❖ Program & Data RAMs
- ❖ Processor core
- ❖ FPGA core

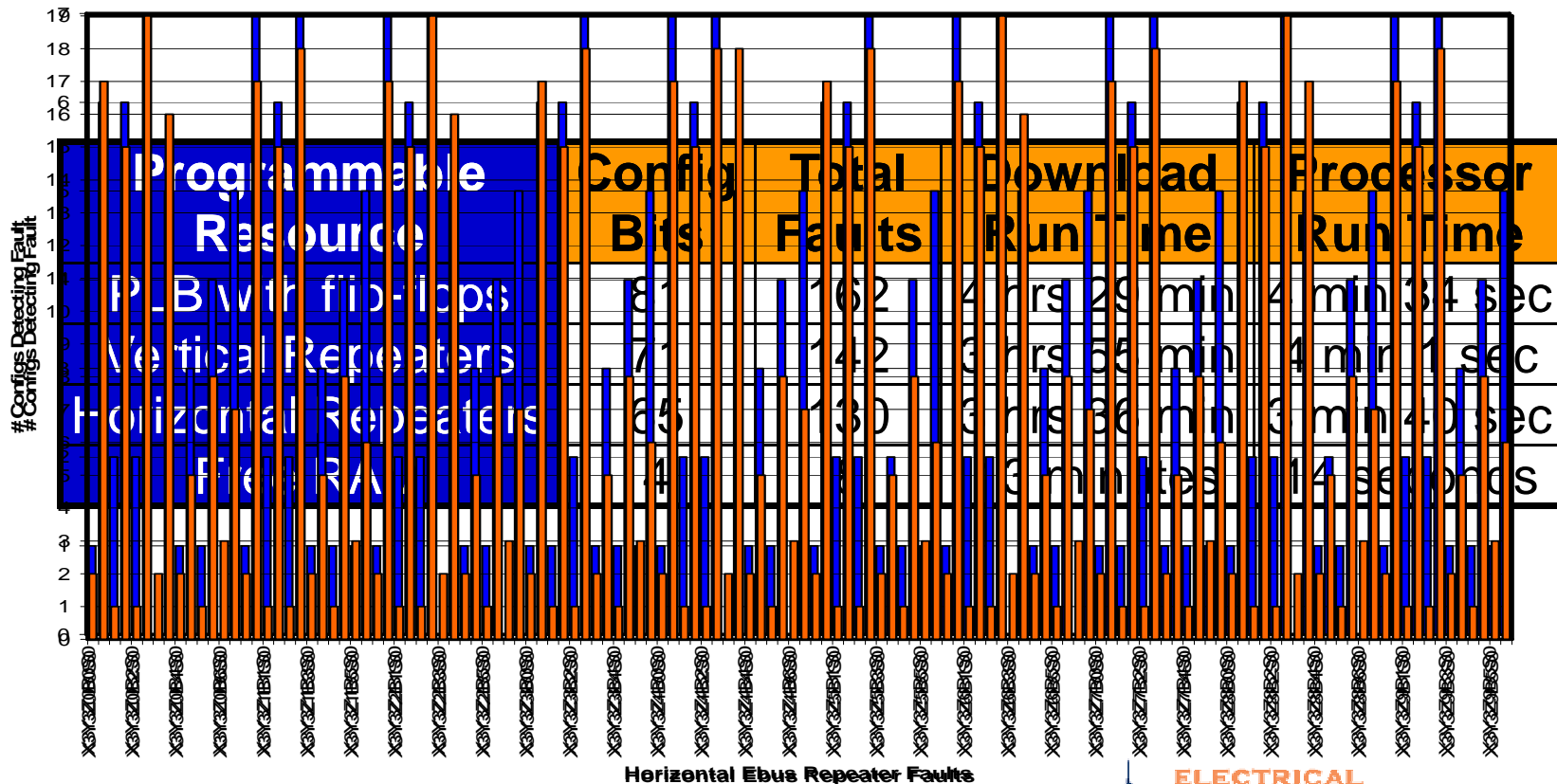
□ Use processor to:

- ❖ Configure FPGA for BIST
- ❖ Run BIST
- ❖ Get BIST results
- ❖ Perform diagnosis
 - ✓ Reduces test and diagnosis time by a factor of 36.9
 - ✓ Store only one BIST and diagnostic program on-chip



Hard Processor Fault Injection

- ❑ Fault injection emulation used for debugging, analysis & verification of BIST configurations
 - ❖ Embedded processor approach gives fast and thorough analysis

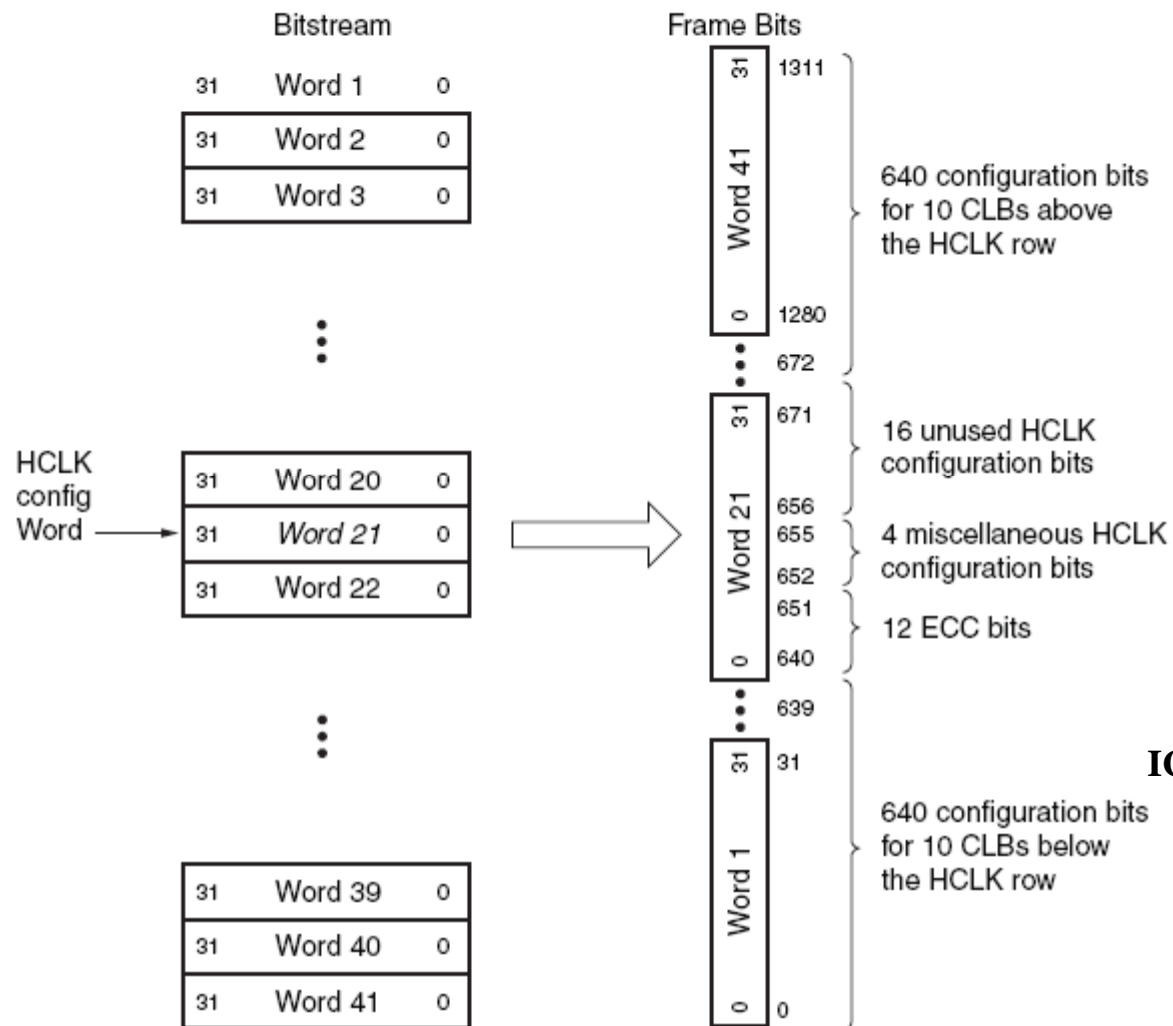


Horizontal Ebus Repeater Faults

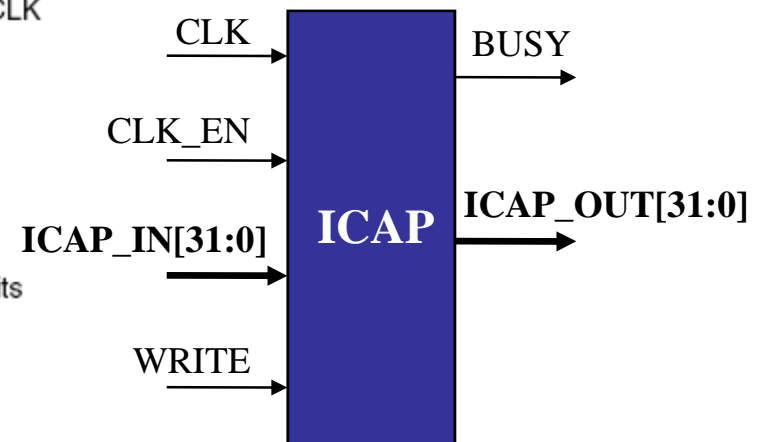
Soft Processor Fault Injection

- ❑ Not all Virtex-4 and Virtex-5 FPGAs include a dedicated “hard” processor
 - ❖ Do include an Internal Configuration Access Port (ICAP) in the FPGA fabric
- ❑ Soft-core processor is required
 - ❖ Modeled in VHDL
 - ❖ Synthesized to a device specific netlist, then incorporated with BIST circuitry
 - ✓ Separate, but operationally identical models for Virtex-4 and Virtex-5 due to architectural differences

Virtex-4 & Virtex-5 Config Memory



- ❑ 1 Frame = 41 32-bit words
- ❑ ICAP provides read/write access to config memory



UG191_c6_09_060407



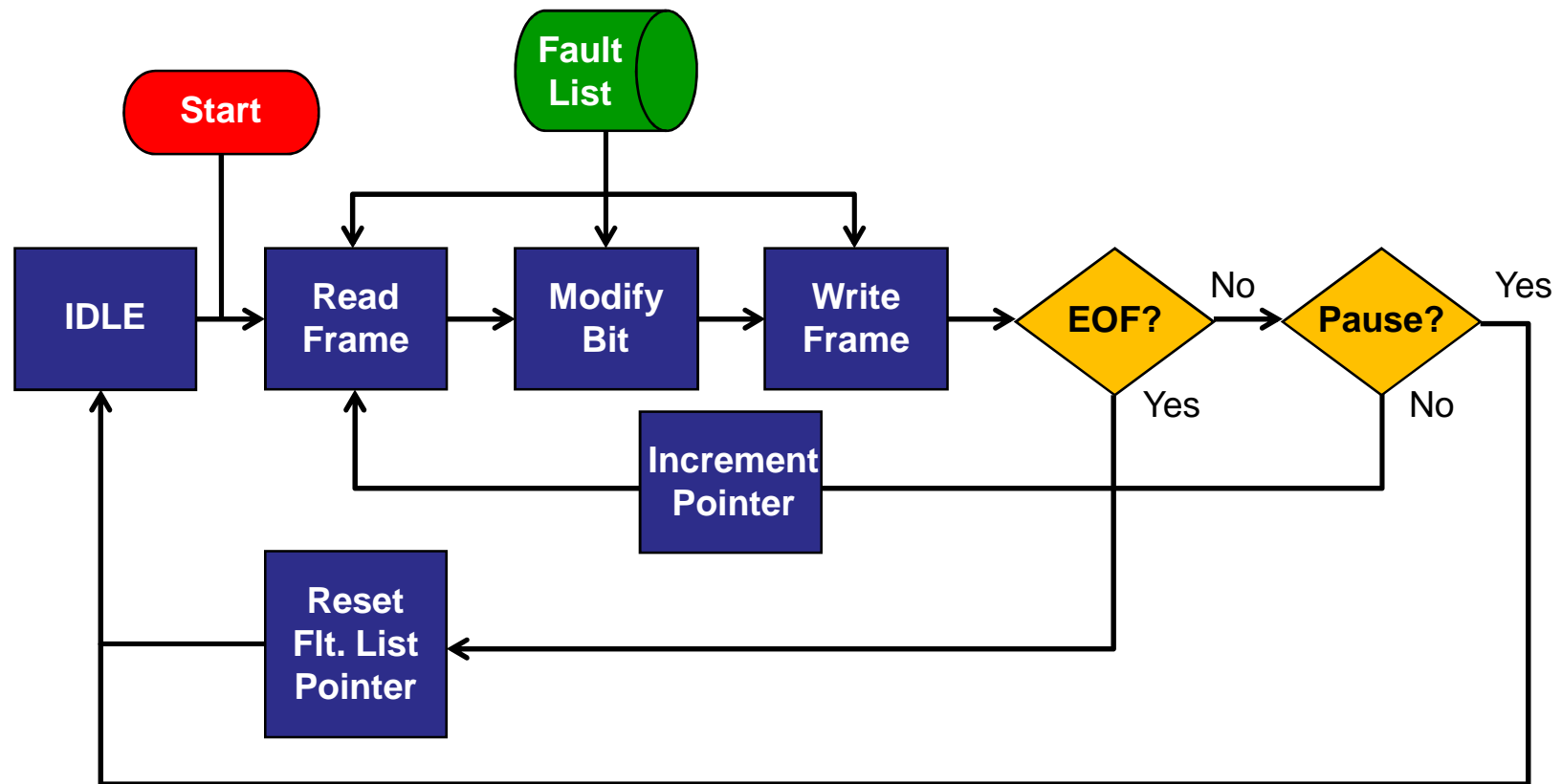
Embedded Fault Injection

- ❑ Fault list stored in block RAM (V-4 18 Kb, V-5 36 Kb)
 - ❖ Arranged in 36-bit words (V-4 = 512, V-5 1024)
 - ❖ Can be initialized during download or in-system via custom boundary scan user-defined register interface
 - ❖ Delimiters are used to control operation of processor
 - ✓ Pause delimiter: enables injection of multiple faults at once
 - ✓ End-of-file delimiter: enables any size fault list (up to maximum)
 - ❖ Fault code: specifies type of fault to inject

35:34	33:32	32:21	20:0
Delimiters	Fault Code	Bit Index	Frame Address

Parity[3:2]	Description	Parity[1:0]	Description
00	Continue to next fault	00	Stuck-at zero
01	Pause at fault	01	Stuck-at one
1X	End-of-file (EOF)	1X	Bit-flip (SEU)

Embedded Fault Injection Operation

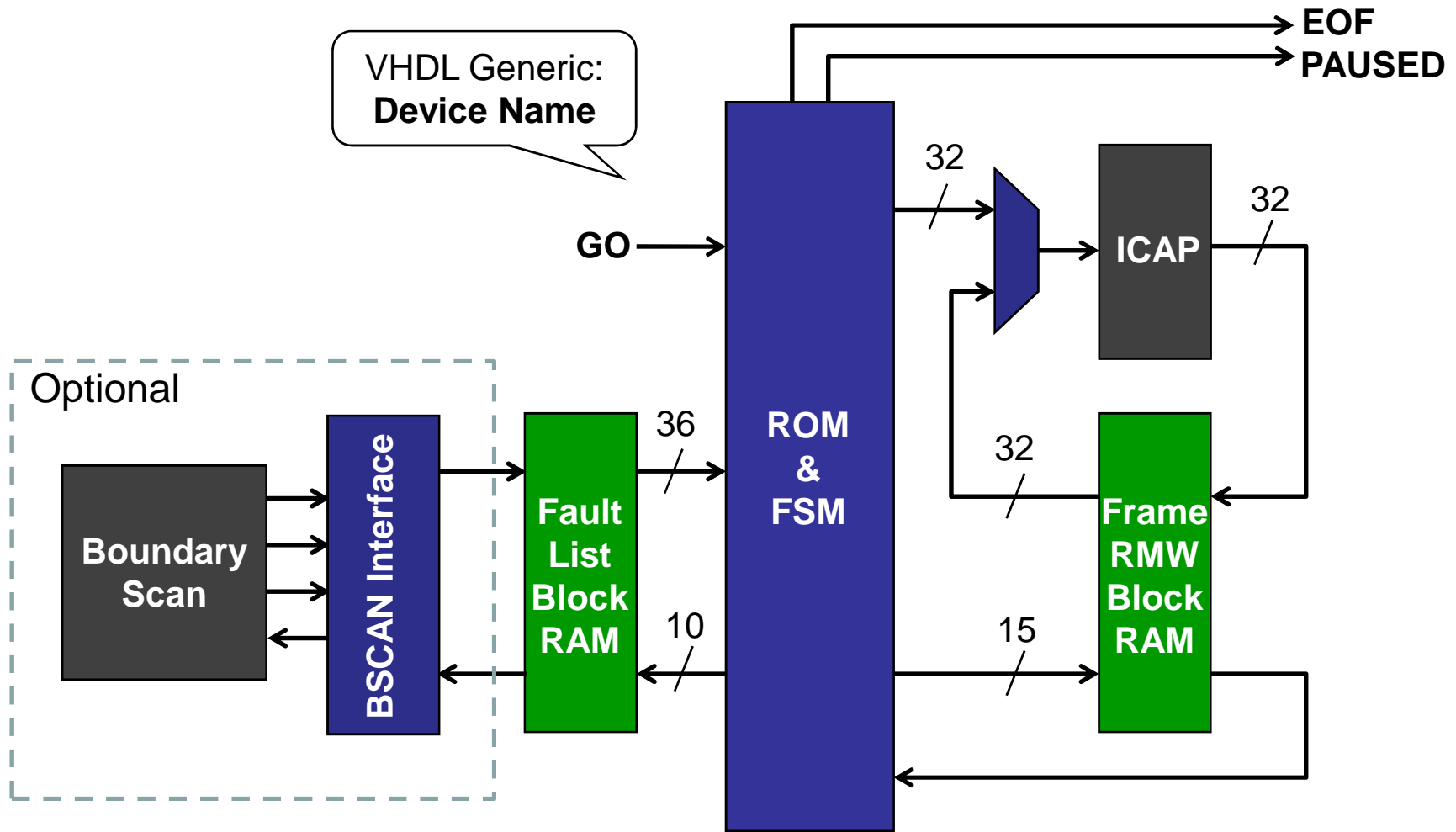


VHDL Component Declaration

```
component fltinject is
generic( DEVICE : string(1 to 6):="LX110T");
port( GO       : in std_logic;
      CLK      : in std_logic;
      EOF      : out std_logic;
      PAUSED   : out std_logic);
end component fltinject;
```

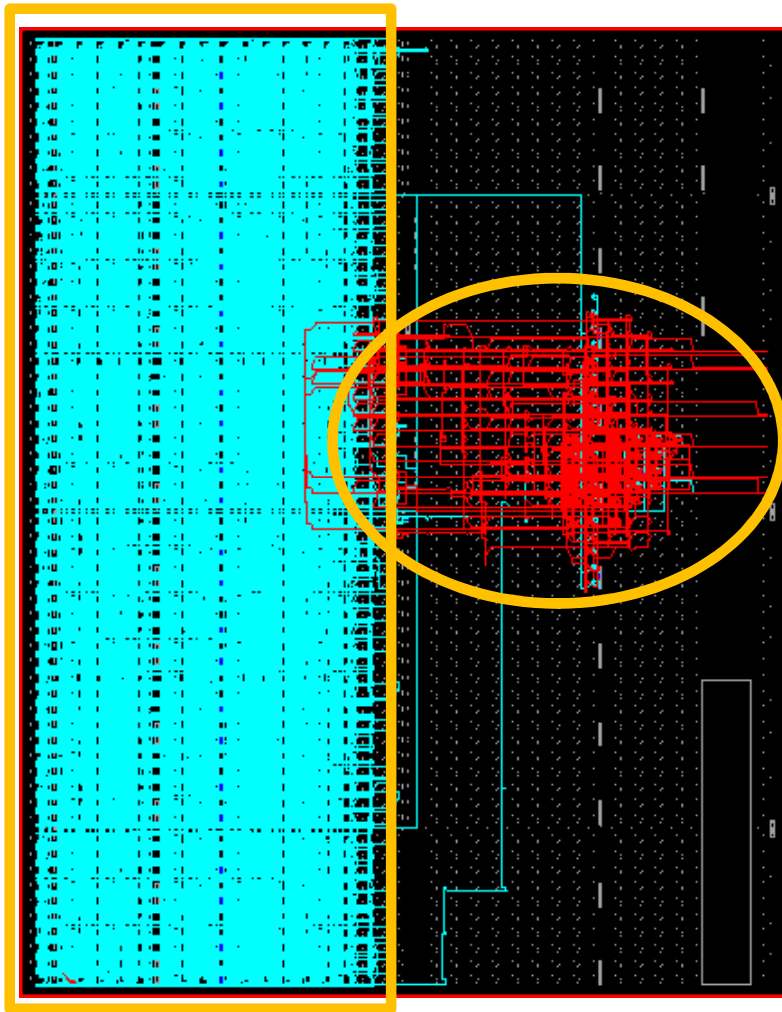
Name	Direction	Description
CLK	Input	Clock input to ICAP (up to 100 MHz)
GO	Input	Digital 1-shot input asserted to injection faults separated by "pause" delimiters
PAUSED	Output	Indicates injection of faults complete
EOF	Output	Asserted when end of list is reached

Fault Inject Core Block Diagram



- Configurable Logic
- Configurable RAMs
- Dedicated Logic

Implementation Results



Fault Injection Core

- Image: Embedded Fault Injection Core with BIST for Configurable Logic Blocks in Virtex-5 LX20T
- Minimum 64 times speed-up versus 50 MHz Boundary Scan
 - More realistic ~9100 times speed-up

	Virtex-4	Virtex-5
# lines of VHDL	~950	~950
# block RAMs	2	2
# Logic Slices	228	67

Conclusions

- ❑ Fault injection is a proven method for verifying fault coverage of BIST and SEU tolerance in FPGAs
 - ❖ Main limitation of previous approaches is download overhead
- ❑ Embedded fault & SEU injection approach
 - ❖ Moves complex circuitry on chip
 - ✓ Can be implemented in “hard” or “soft” processor
 - ✓ Applicable to any FPGA with internal access to the configuration memory
 - ❖ Can operate at system bus speeds
 - ✓ Minimum 64 times speed-up versus Boundary Scan in Virtex-4 and Virtex-5

Thank You

