FT-UNSHADES 2

A fault injection platform for early evaluation of SEE reliability of deep submicron and FPGA designs.

J.M. Mogollon, H. Guzmán-Miranda, J. Nápoles, J. Barrientos, M.A. Aguirre

Univ. of Sevilla-ESA-AICIA
Summary

- Principles of FTUNSHADES
- Wish List for FTU2
- FTU2 Approach
- FTU2-DB Hardware Design Finished
- Data transfer via PCI-Express
  - V5 IO High Speed Capabilities
- Advances on Virtex5
  - Readback/Reconfiguration
- User center at University of Sevilla
FT-UNSHADES

Fault Injection is a way, using an alternative technology, for an assessment of some potential vulnerabilities at early stages of the design process.

- Let’s think on DIGITAL circuits
- Let’s think on a DYNAMIC test. The starting point is:
  - NETLIST described in VHDL, VERILOG, EDIF, SCHEMATIC
  - STIMULI SET moving the circuit sufficiently
- Initial objectives:
  - Detect reliability of the design against SEUs.
  - Detect SEU vulnerable points (Collapsed triplets, protections,…)
  - Detect how faults are propagated through the netlist.
  - Study hardening by functional self repairing structures.
- Other outcomes:
  - Check the correct initialization strategy (RESET)
  - Check protections by hierarchical module
  - Check Multi-SEU (MBU, …)
Exploiting Run-Time reconfiguration for FAULT INJECTION

- The execution is made using an SRAM-FPGA from Xilinx
- The injection is performed using the configuration circuit
- The injection is made modifying either the REGISTER CONTENTS or THE CONFIGURATION CIRCUIT
Quick Overview of FTUNSHADES

- Development of 2004
- A Fault Injection system based on concepts related to hardware debugging: **observability** and **controlability**.*
- The method is **non intrusive**. The design is analyzed with few modifications.
- The design is analyzed using a stimuli set or **application (workload)**
- **Analysis** of a design reliability by attacking user registers, memory elements or configuration bits. The results are analyzed from the design behavior point of view.
- A hardware accelerator allows to speed up the analysis. (This is the meaning of **emulation**, instead of simulation)
- **Massive injection campaign** and **detailed analysis** of the design are performed in the same platform.
- Figure: 100 faults/second

* in this context, **observability** means accessing to the whole registers of the design at every clock cycle. **Controlability** is the capability to reproduce an event in the circuit at any time of the workload.
Quick View of FTUNSHADES

WORKLOAD

GOLDEN INSTANCE

SEU INSTANCE

FPGA

Comparator
System Clock
Workload inputs
Quick View of FTUNSHADES
This model is similar to a system in an accelerator.

Two identical instances of the design are implemented in one FPGA.

The inputs are stored in external memories and the outputs are compared between both instances.

The system clock is common to each instance. Both instances work in parallel.

The injection is always to the faulty instance. The gold one works for comparison.

The register selection is made using a demapping information provided by Xilinx tools.

Time variable is also controlled.

It is a deterministic attack using the design hierarchical organization.

- This model is similar to a system in an accelerator.
- Two identical instances of the design are implemented in one FPGA.
- The inputs are stored in external memories and the outputs are compared between both instances.
- The system clock is common to each instance. Both instances work in parallel.
- The injection is always to the faulty instance. The gold one works for comparison.
- The register selection is made using a demapping information provided by Xilinx tools.
- Time variable is also controlled.
- It is a deterministic attack using the design hierarchical organization.
Wish List for FTU2

- Bigger and more complex designs
- Faster Campaigns
- Even less intrusive
- Easy to use, avoiding complex recipes to prepare campaigns
- Open to other possibilities:
  - Testing any FPGA family
  - Comparison FPG/ASIC
  - In beam testing
  - Diagnostic of failures in systems

22/08/2011 ReSPACE/MAPLD 2011
This model is similar to a system in an accelerator.

Two identical instances of the design are implemented in different FPGAs.

The inputs are stored in external DDR external memory and the outputs are compared in the control FPGAs.

The injection is always to the faulty FPGA. The gold one works for comparison.

The register selection is made using a demapping information provided by Xilinx tools in the host PC.

Preserves the deterministic injection using the design hierarchical organization.

---

**Standard model**

- This model is similar to a system in an accelerator.
- Two identical instances of the design are implemented in different FPGAs.
- The inputs are stored in and external DDR external memory and the outputs are compared in the control FPGAs.
- The injection is always to the faulty FPGA. The gold one works for comparison.
- The register selection is made using a demapping information provided by Xilinx tools in the host PC.
- Preserves the deterministic injection using the design hierarchical organization.
FTU2-DB Approach
FTU2-DB Hardware Design
Finished
FTU2 in Numbers

- **Maximum number of device under test (DUT) I/O signals**

<table>
<thead>
<tr>
<th>Max Signals</th>
<th>Configuration Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>581</td>
<td>SelectMap x8</td>
</tr>
<tr>
<td>573</td>
<td>SelectMap x16</td>
</tr>
<tr>
<td>557</td>
<td>SelectMap x32</td>
</tr>
</tbody>
</table>

- **Maximum DUT design size**

<table>
<thead>
<tr>
<th></th>
<th>V5FX70T</th>
<th>V5SX95T</th>
<th>V5VLX155T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slices</td>
<td>11200</td>
<td>14720</td>
<td>24320</td>
</tr>
<tr>
<td>Flip-Flops</td>
<td>44800</td>
<td>58880</td>
<td>97280</td>
</tr>
<tr>
<td>Block Rams</td>
<td>5.20Mb</td>
<td>8.58Mb</td>
<td>7.45Mb</td>
</tr>
<tr>
<td>Bitstream (bits)</td>
<td>27025408</td>
<td>35716096</td>
<td>43042304</td>
</tr>
</tbody>
</table>
FTU2 in Numbers

- Platform Memory Size to store Test Vectors
  - 2GB of DDR2 memory in two modules

  Maximum workload:

  \[ \text{Nº Inputs} \times \text{CLK cycles} = 2 \times 8 \times 2^{30} \]

  Example: 200 inputs => more than 85 millions of clk cycles

- Memory can be easily expanded replacing DIMM Modules
FTU2 in Numbers

- Maximum DUT clock frequency\(^1\)

<table>
<thead>
<tr>
<th>X = Max. Number of Inputs or Outputs</th>
<th>Max. Frequency (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X&lt;128</td>
<td>250</td>
</tr>
<tr>
<td>128&lt;X&lt;256</td>
<td>125</td>
</tr>
<tr>
<td>256&lt;X&lt;384</td>
<td>83.3</td>
</tr>
<tr>
<td>384&lt;X&lt;512</td>
<td>62.5</td>
</tr>
<tr>
<td>512&lt;X&lt;581</td>
<td>50</td>
</tr>
</tbody>
</table>

\(^1\)Considering the limit due to the serialization/deserialization issues (GTX transceivers at maximum speed of 4Gb/s and x8 lanes).
FTU2 in Numbers

- **Run time**
  \[
  T_{\text{RUN}} \leq T_{\text{bit\_flip}} + T_{\text{Workload}}
  \]

  - Time necessary to inject a fault.
    - By hardware => 2000 CLK cycles.
    - \( F=100\text{MHz} \rightarrow T_{\text{bit\_flip}}=20\ \mu\text{s} \)

<table>
<thead>
<tr>
<th>Workload</th>
<th>CLK (MHz)</th>
<th>( T_{\text{RUN}} )</th>
<th>Nº RUNS/s</th>
<th>( T_{\text{bit_flip}}) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>500000</td>
<td>125</td>
<td>4020( \mu\text{s} )</td>
<td>248</td>
<td>0.5</td>
</tr>
<tr>
<td>250000</td>
<td>125</td>
<td>2020( \mu\text{s} )</td>
<td>495</td>
<td>1</td>
</tr>
<tr>
<td>125000</td>
<td>125</td>
<td>1020( \mu\text{s} )</td>
<td>980</td>
<td>2</td>
</tr>
<tr>
<td>50000</td>
<td>125</td>
<td>420( \mu\text{s} )</td>
<td>2380</td>
<td>4.7</td>
</tr>
<tr>
<td>125000</td>
<td>250</td>
<td>520( \mu\text{s} )</td>
<td>1923</td>
<td>3.84</td>
</tr>
</tbody>
</table>
V5 IO High Speed Capabilities

RocketIO GTP Transceivers (LXT/SXT only)
- Full-duplex serial transceiver capable of 100 Mb/s to 3.75 Gb/s baud rates
- 8B/10B, user-defined FPGA logic, or no encoding options
- Channel bonding support
- CRC generation and checking
- Programmable pre-emphasis or pre-equalization for the transmitter
- Programmable termination and voltage swing
- Programmable equalization for the receiver
- Receiver signal detect and loss of signal indicator
- User dynamic reconfiguration using secondary configuration bus
- Out of Band (OOB) support for Serial ATA (SATA)
- Electrical idle, beaconing, receiver detection, and PCI Express and SATA spread-spectrum clocking support
- Less than 100 mW typical power consumption
- Built-in PRBS Generators and Checkers

RocketIO GTX Transceivers (TXT/FXT only)
- Full-duplex serial transceiver capable of 150 Mb/s to 6.5 Gb/s baud rates
- 8B/10B encoding and programmable gearbox to support 64B/66B and 64B/67B encoding, user-defined FPGA logic, or no encoding options
- Channel bonding support
- CRC generation and checking
- Programmable pre-emphasis or pre-equalization for the transmitter
- Programmable termination and voltage swing
- Programmable continuous time equalization for the receiver
- Programmable decision feedback equalization for the receiver
- Receiver signal detect and loss of signal indicator
- User dynamic reconfiguration using secondary configuration bus
- OOB support (SATA)
- Electrical idle, beaconing, receiver detection, and PCI Express spread-spectrum clocking support
- Low-power operation at all line rates

3.75Gb/s  6.5Gb/s

22/08/2011  ReSPACE/MAPLD 2011
Data transfer via PCI-Express

- **Experimental Results**

<table>
<thead>
<tr>
<th>PCIe v2.x</th>
<th>x1 (Real)</th>
<th>x1 (Effective)</th>
<th>CLK (Mhz)</th>
<th>Bus Width (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total BW</td>
<td>5Gbs/s</td>
<td>4Gbs/s</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BW per direction</td>
<td>2.5Gbs/s</td>
<td>2Gbs/s</td>
<td>62.5/125</td>
<td>32/16</td>
</tr>
<tr>
<td>Total BW</td>
<td>10Gbs/s</td>
<td>4Gbs/s</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BW per direction</td>
<td>5Gbs/s</td>
<td>4Gbs/s</td>
<td>125/250</td>
<td>32/16</td>
</tr>
</tbody>
</table>

The total effective BW does not have any overhead (only send the comma character periodically as desired by the user to keep the alignment in the user parallel interface).
Structure of the system

- FT-UNSHADES
- User Commands (Scripts)
  - VCD format
  - Session.log
  - Excel Sheet
  - Console
- WORKLOAD
- BITSTREAM
- BIT ALLOCATION

FT-UNSHADES Analysis Example

CLOCK: 133937
REGISTER: leon0_mcore0_proc0_cx.c0_icache0_r.waddress_16
DAMAGE DETECTED: YES
LATENCY: 1 CLK
PORT: address

RUN 1
Selected clk cycle for SEU insertion: 133937
Selected reg for SEU insertion: SEU_MUT/leon0_mcore0_proc0_cx.c0_icache0_r.waddress_16
OK
Output error detected in port: address
Damage detected 1 clk cycle after SEU insertion
Total elapsed time: 0.062501
Target size: 1, registers
Total FPGA Cycles: 0x0,00020B32 (133938)
FT-UNSHADES user center at University of Sevilla
User friendly interface
Conclusions

- FT-UNSHADES2 is the evolution of the FT-UNSHADES system presented in MAPLD in 2005.
- The original concept has been demonstrated to be powerful and versatile.
- FT-UNSHADES2 is the new generation of Fault Injectors for SEU and MBU protections.
- It has been conceived as a simple to use and flexible system for complex netlist evaluation.
- The model based on MB and DB is flexible enough to migrate to new FPGAs, ASICs, and Systems.
- The remote access service allows the use of FTUNSHADES through web.
New FTU2 current uses

- In FTU2, the Module Under Test is implemented alone in a dedicated FPGA → we are evaluating the use of the platform an in beam radiation experiment
- New possibilities of using HASH CODES for fault injection in Systems. This idea will open a window to diagnostic of multiple chip based systems.
- Fault Injection based on run-time reconfiguration offers possibilities of SEU and MBU testing of:
  - ASIC netlists
  - SRAM-FPGAs (even non-Xilinx)
  - ASIC in beam (National Accelerators Centre in SPAIN)
  - SYSTEMs
- We are also evaluating the use of FTU2 as a platform to emulate stuck-at bits
Thank you for your attention

Q&A

aguirre@gte.esi.us.es

Hope to SEE you in RADECS 2011