FT-Unshades maintenance

TEC-ED & TEC-SW

Final presentation days, 9th May 2017

Contents

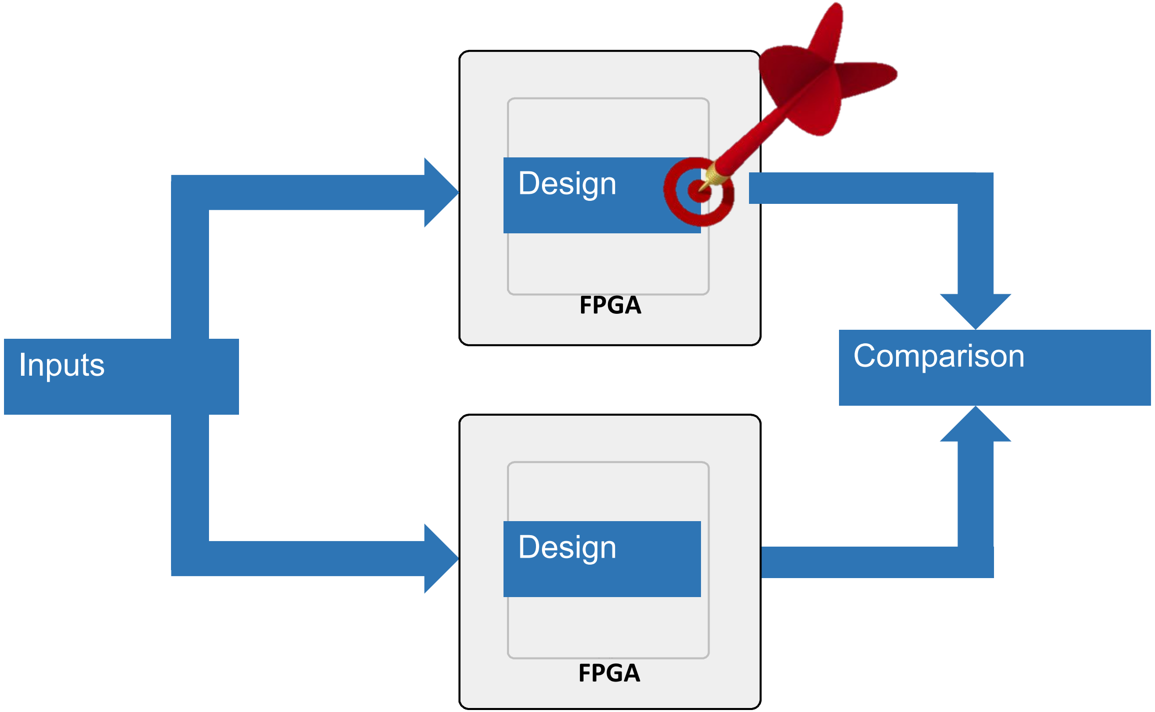
- Introduction
- Usability improvements
- Bug fixes
- Extension of injection coverage
- Analog FTU integration
- Beam experience
- Conclusions and future work

Contents

- Introduction
- Usability improvements
- Bug fixes
- Extension of injection coverage
- Analog FTU integration
- Beam experience
- Conclusions and future work

Introduction

FT-Unshades2: FPGA-based fault injection emulator (SEUs).



Also an analog utility, **AFTU** (SETs).

Introduction

FT-Unshades2: FPGA-based fault injection emulator. Also an analog utility (**AFTU**).

Objectives of the maintenance contract:

- Improve usability & documentation
- Bug fixes
- Continue development & improvements

Contents

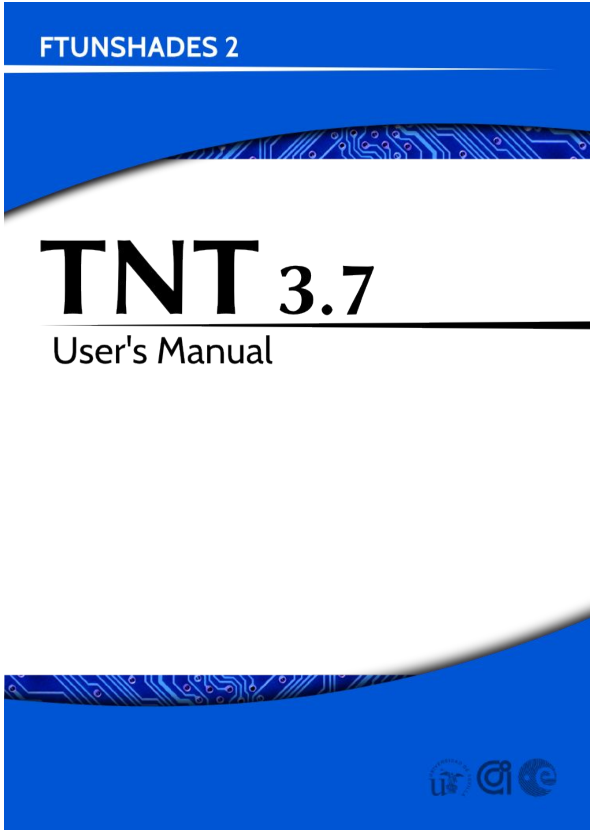
- Introduction
- **Usability improvements**
- Bug fixes
- Extension of injection coverage
- Analog FTU integration
- Beam experience
- Conclusions and future work

Usability improvements

- Improvements on the User Friendly Interface (UFF)
  - Ease of use, waveform presentation
- A report on the usability of the Analog FTU tool has been written and delivered
- TNT user manual and scripting guide
- UFF user manual

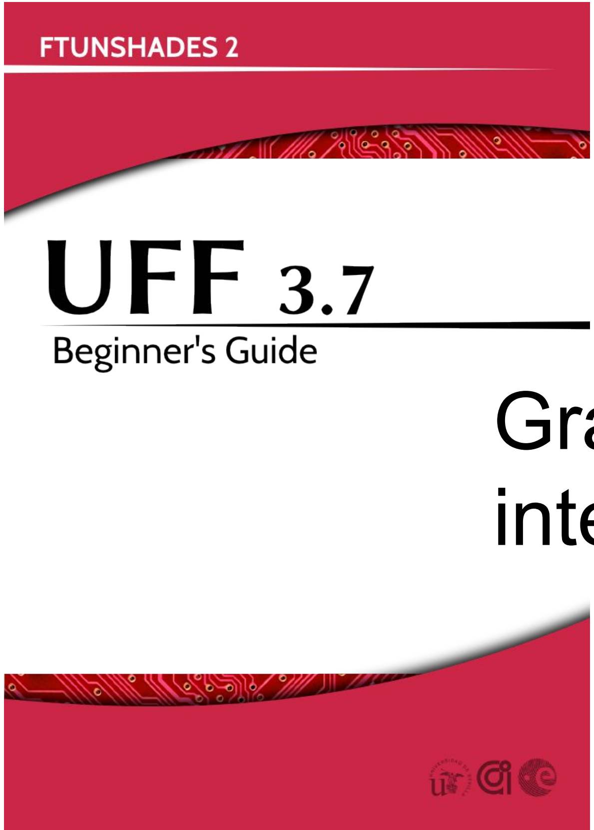
New user guides

Shell


FTUNSHADES 2

TNT 3.7

User's Manual


FTUNSHADES 2

UFF 3.7

Beginner's Guide

Graphical interface

Contents

- Introduction
- Usability improvements
- **Bug fixes**
- Extension of injection coverage
- Analog FTU integration
- Beam experience
- Conclusions and future work

Issues solved

More than 100 issues solved:

- HW / SW stability
- Bug fixes
- New features

Contents

- Introduction
- Usability improvements
- Bug fixes
- **Extension of injection coverage**
- Analog FTU integration
- Beam experience
- Conclusions and future work

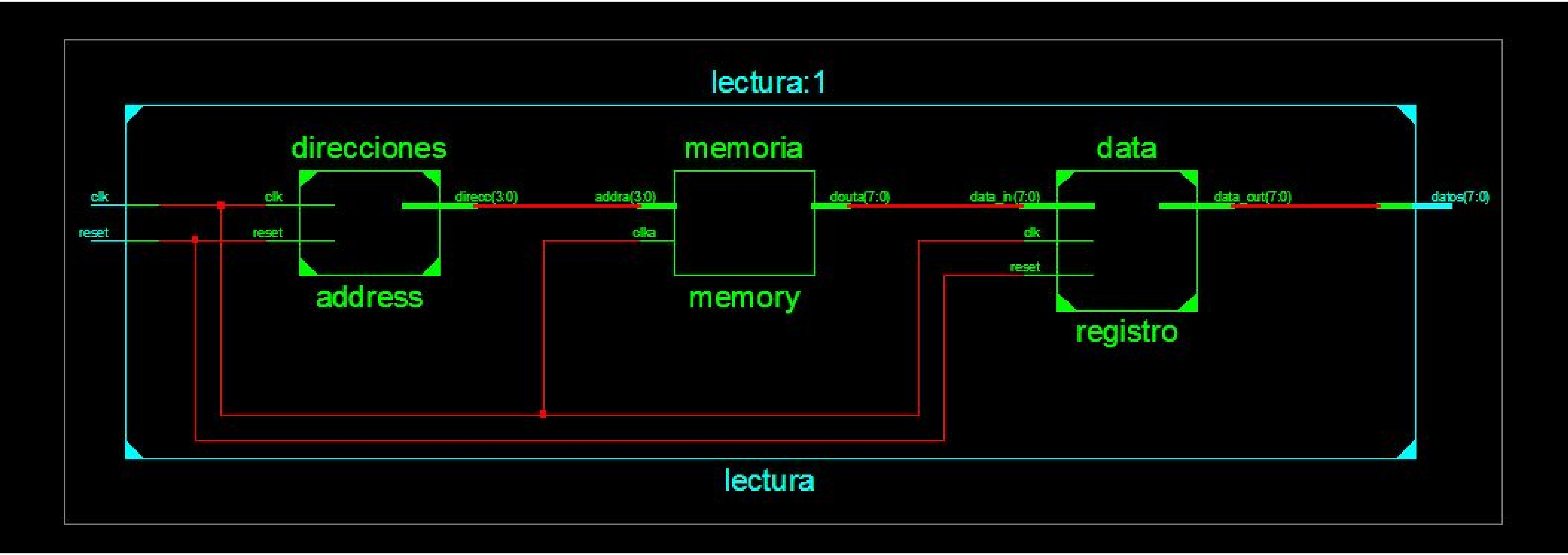
Extension of Injection Coverage

Previously, only injection on Flip-flops was supported:

- High demand for injection in Block RAMs
- -> Study SEUs in Microprocessor memories

Injection in Block RAMs **now supported**

Extension of Injection Coverage



Design with embedded BRAM

Extension of Injection Coverage

Read from BRAM (TNT)

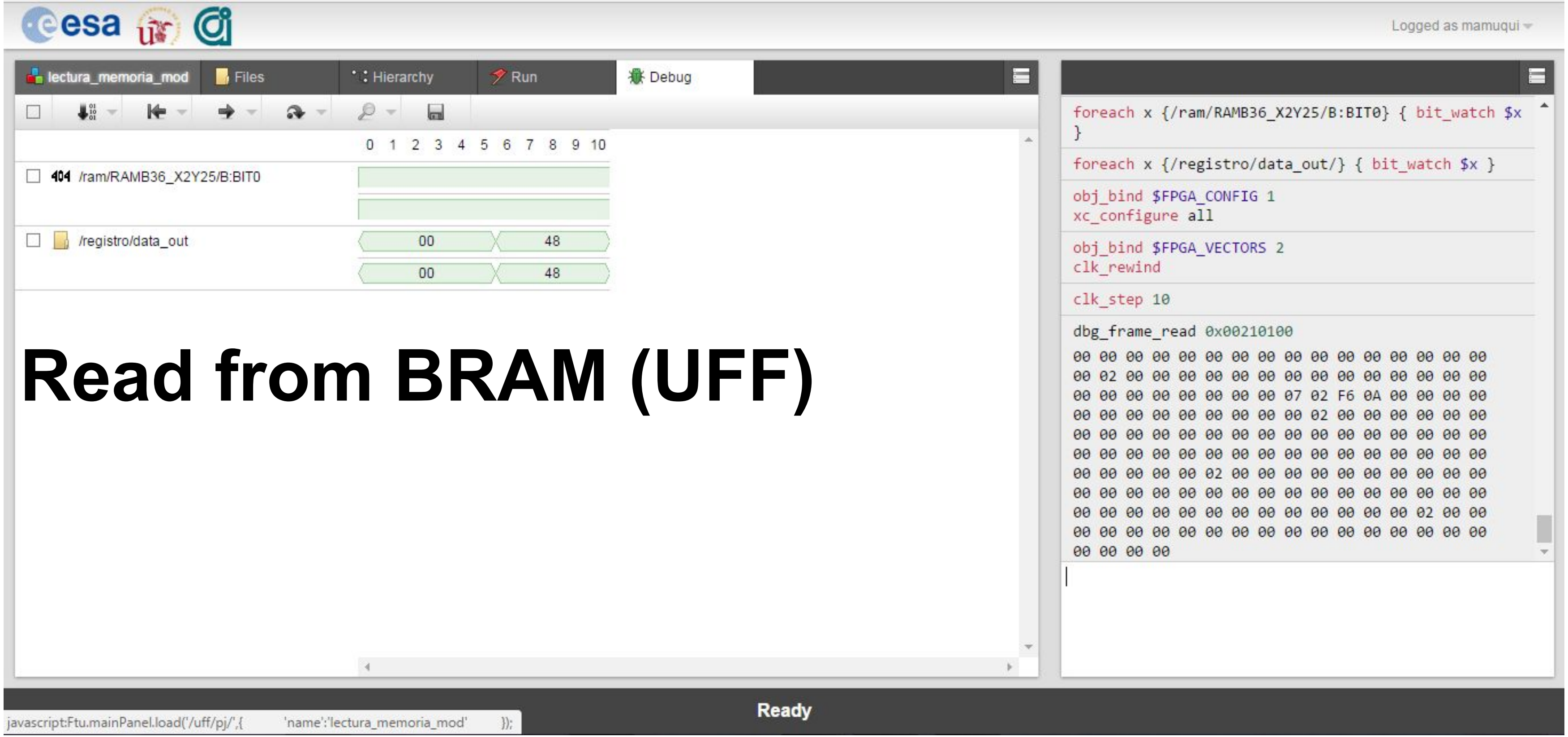
```
% load_vectors lectura.dat
6
% xc_configure all
% clk_rewind
% clk_step 10
% dump_logs 0 10
/ram/RAMB36_X2Y25/B:BIT0.g: 1 1 1 1 1 1 1 1 1 1 1
/ram/RAMB36_X2Y25/B:BIT0.f: 1 1 1 1 1 1 1 1 1 1 1
/registro/data_out.g: 00 00 00 00 00 00 48 48 48 48 48
/registro/data_out.f: 00 00 00 00 00 00 48 48 48 48 48
% dbg_frame_read 0x00210100
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 07 02 F6 0A 00 00 00 00
00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
```

Extension of Injection Coverage

Write to BRAM (TNT)

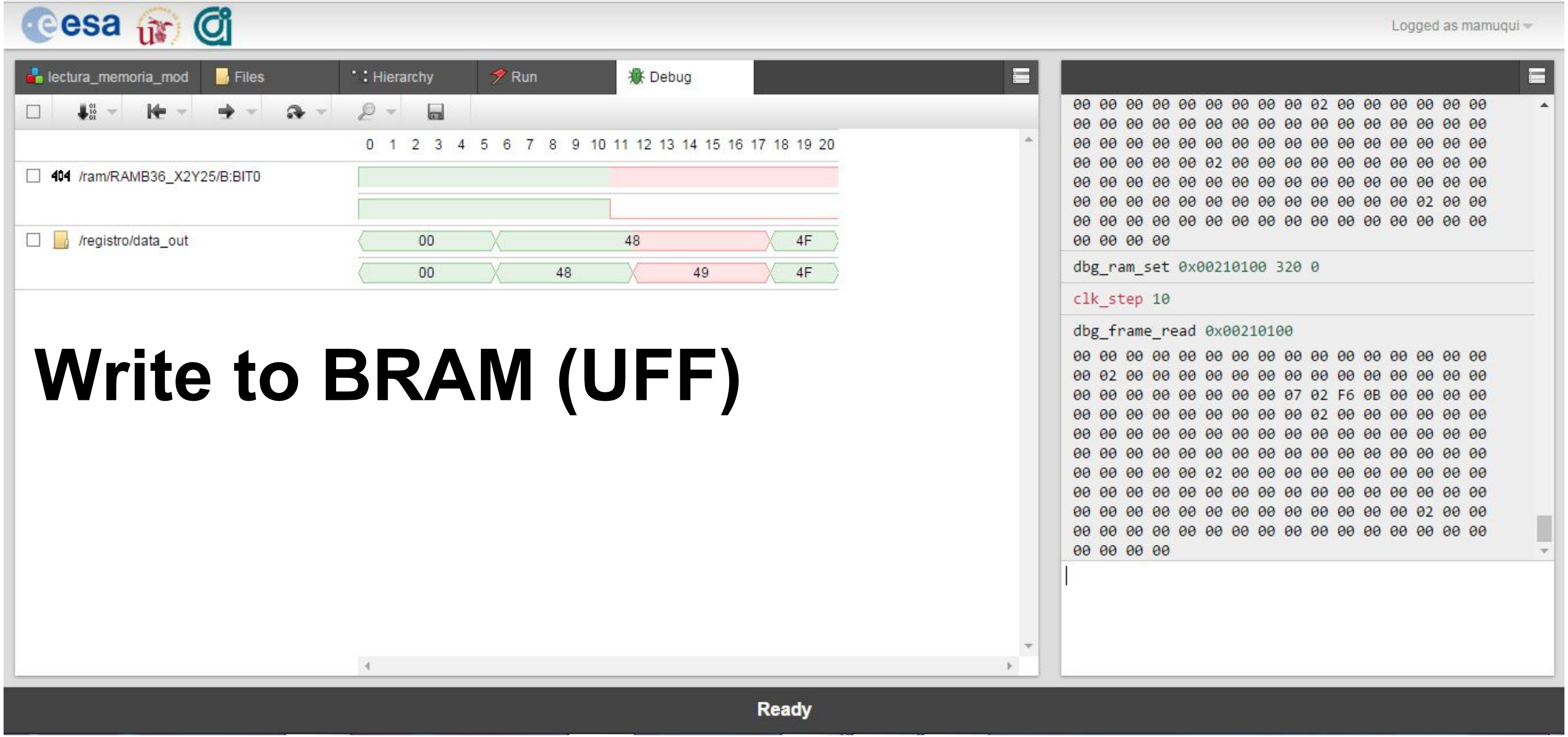
```
% dbg_ram_set 0x00210100 320 0
% clk_step 10
% dump_logs 0 20
/ram/RAMB36_X2Y25/B:BIT0.g: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
/ram/RAMB36_X2Y25/B:BIT0.f: 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
/registro/data_out.g: 00 00 00 00 00 00 48 48 48 48 48 48 48 48 48 48 48 48 4F 4
F 4F
/registro/data_out.f: 00 00 00 00 00 00 48 48 48 48 48 48 49 49 49 49 49 49 4F 4
F 4F
% dbg_frame_read 0x00210100
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 07 02 F6 0B 00 00 00 00
00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
```

Extension of Injection Coverage



Read from BRAM (UFF)

Extension of Injection Coverage



Write to BRAM (UFF)

Contents

- Introduction
- Usability improvements
- Bug fixes
- Extension of injection coverage
- **Analog FTU integration**
- Beam experience
- Conclusions and future work

Analog FT-Unshades

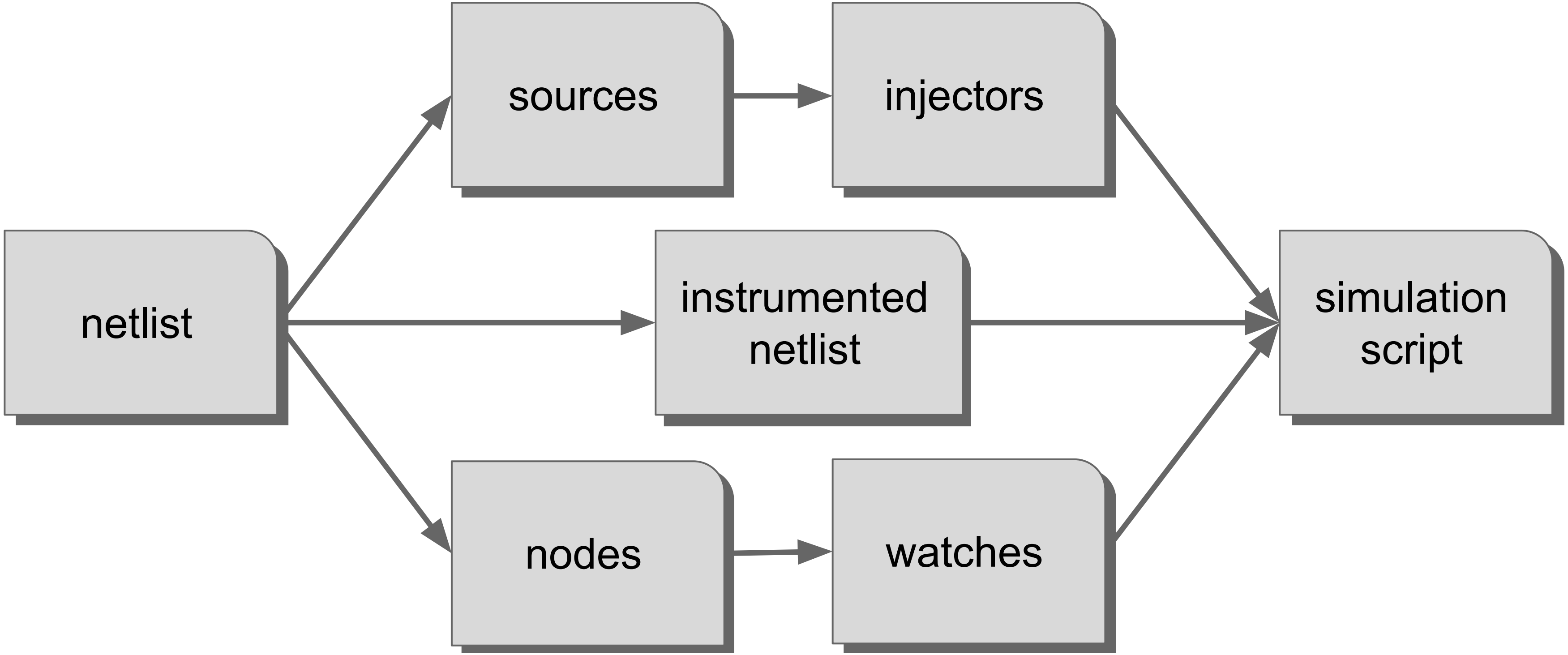
What does it do?

- Analyzes the effects of radiation on analog circuits

How does it do it?

- Instrument a circuit
- Create injectors
- Create watches
- Build the actual simulation

Workflow



sources
injectors
netlist
instrumented netlist
simulation script
nodes
watches

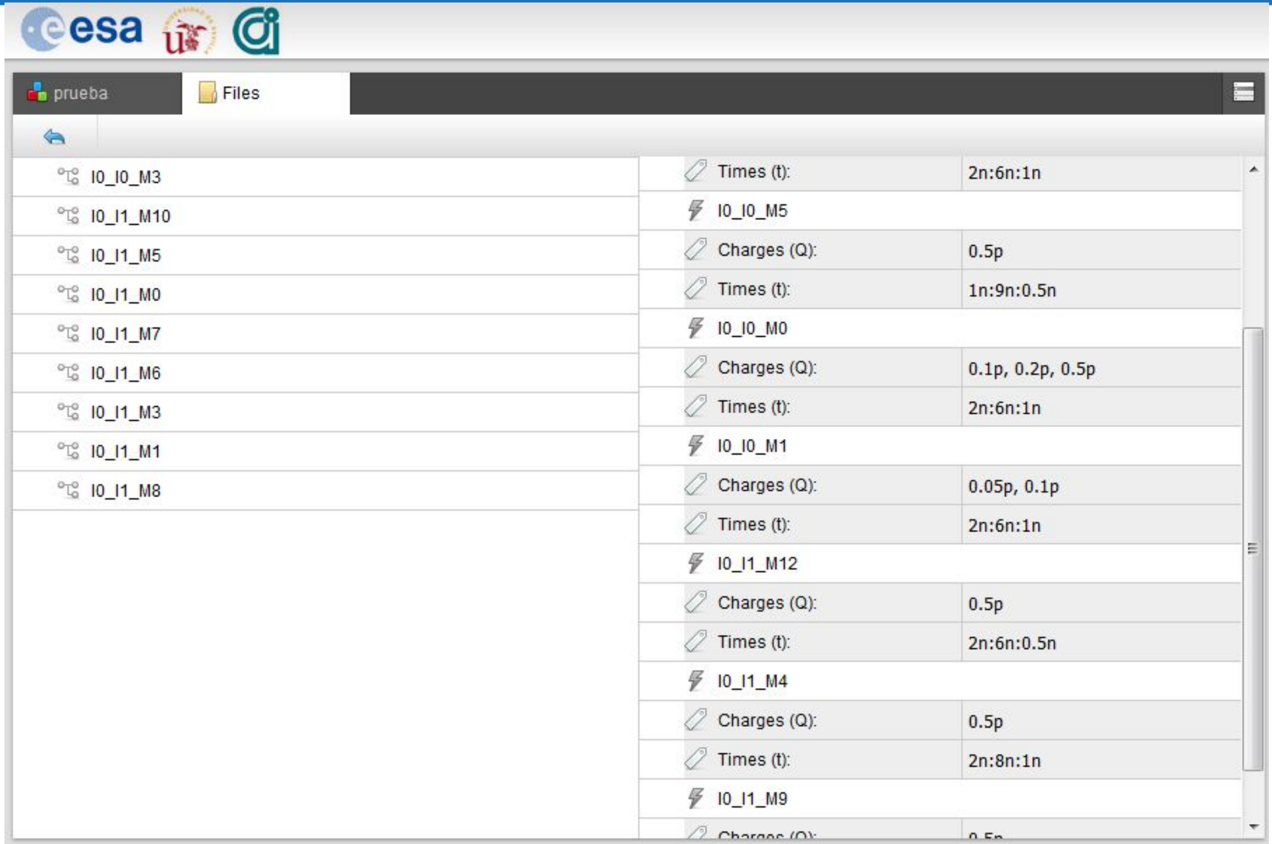
Instrumentation

- Take the design of a circuit and add instruments to alter its behavior.
- For every technology, a small database of instrumentable elements is used.
- It generates three files:
  - an instrumented version of the original netlist;
  - a list of sources where faults may be injected;
  - a list of nodes that can be watched during the simulation.

Injectors

- Determine where and when faults are injected during the simulation.
- Candidates are in the source list generated during the instrumentation step.
- Each element is injected at impacts times and with charges provided by the user.

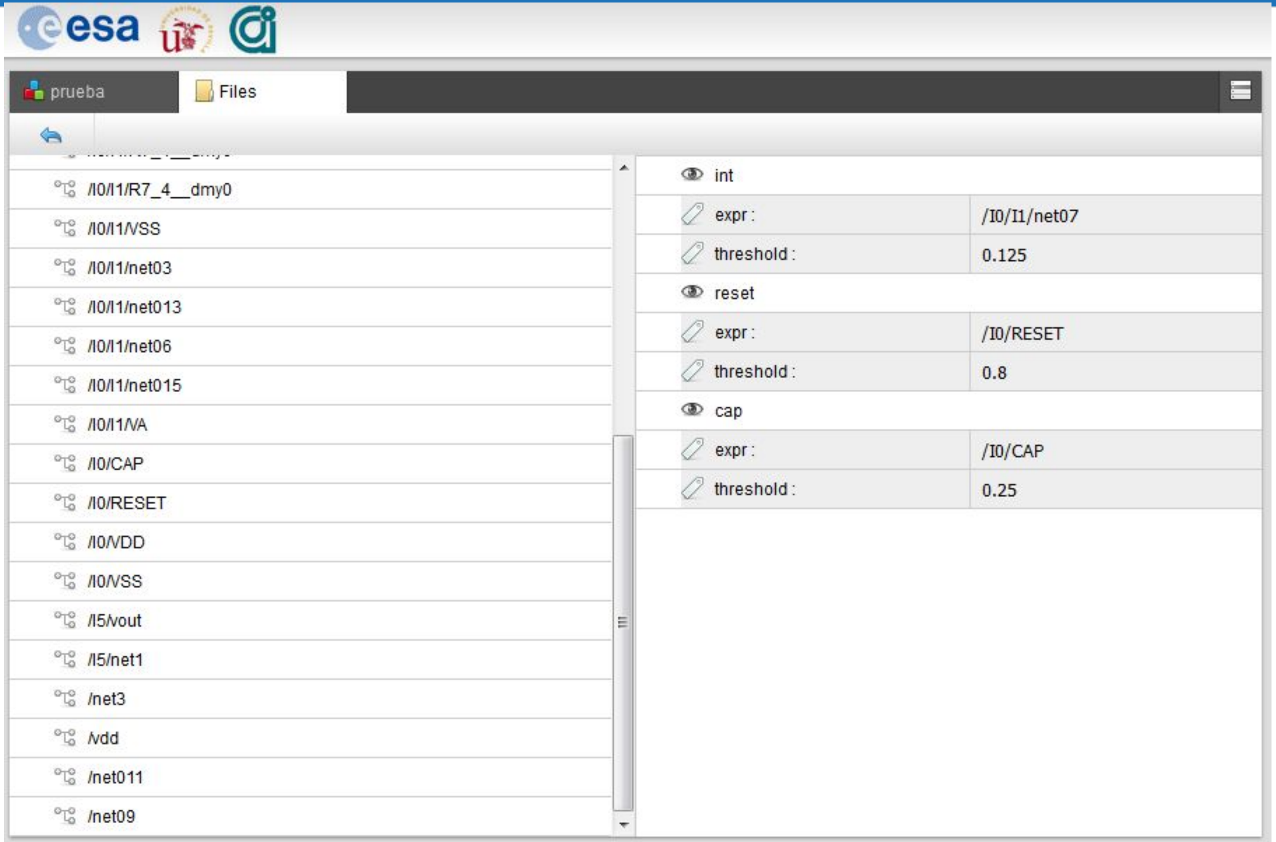
Injectors


esa
prueba
Files
I0_I0_M3
I0_I1_M10
I0_I1_M5
I0_I1_M0
I0_I1_M7
I0_I1_M6
I0_I1_M3
I0_I1_M1
I0_I1_M8
Times (t): 2n:6n:1n
I0_I0_M5
Charges (Q): 0.5p
Times (t): 1n:9n:0.5n
I0_I0_M0
Charges (Q): 0.1p, 0.2p, 0.5p
Times (t): 2n:6n:1n
I0_I0_M1
Charges (Q): 0.05p, 0.1p
Times (t): 2n:6n:1n
I0_I1_M12
Charges (Q): 0.5p
Times (t): 2n:6n:0.5n
I0_I1_M4
Charges (Q): 0.5p
Times (t): 2n:8n:1n
I0_I1_M9

Watches

- Determine which elements are analyzed during the simulation.
  - Node signals or composed expressions
- Candidates are in the node list generated during the instrumentation step.
- A heuristic is used, simulation-wide, to determine how the circuit differs from its correct behaviour, defining the test campaign.

Watches


esa
prueba
Files
/I0/I1/R7_4__dmy0
/I0/I1/VSS
/I0/I1/net03
/I0/I1/net013
/I0/I1/net06
/I0/I1/net015
/I0/I1/VA
/I0/CAP
/I0/RESET
/I0/VDD
/I0/VSS
/I5/vout
/I5/net1
/net3
/vdd
/net011
/net09
int
expr : /I0/I1/net07
threshold : 0.125
reset
expr : /I0/RESET
threshold : 0.8
cap
expr : /I0/CAP
threshold : 0.25

The Simulation

The final output is:

- A modified netlist on which the simulation runs;
- An Ocean-based script that evaluates the circuit's radiation performance.

With both, an SET sensitivity analysis can be performed in the Spectre-based simulator.

- .csv output file

Analog FTU maintenance

Work from previous version of AFTU:

- Implementation of a GUI in UFF 3.7, integrated with digital fault injection tool.
- Adaptation to new technologies (IHP 130 nm, TSMC 65nm, UMC 65nm)
- Tool development and usability tests with analysis of real circuits (ours and 3rd-party).

Analog FTU maintenance

Potential new features related to:

- Test campaign and fault injection upgrades
  - Increase user-friendliness (file formats, GUI options)
  - Allow hierarchical & random injections
- Models and heuristics improvement
  - Computer efficiency, additional heuristics
  - Additional total dose studies (TID + SET)
- Study of interoperability with FTU (digital)
- Multiple impacts emulation

Contents

- Introduction
- Usability improvements
- Bug fixes
- Extension of injection coverage
- Analog FTU integration
- **Beam experience**
- Conclusions and future work

Beam experience

F.I. and radiation tests tipically differ because:

- Different FPGA device
- Different implementation pin constraints
- Different P&R solutions
- Different time windows
- [...]

What if we use **exactly the same hardware** with **exactly the same bitstream**?

Beam experience

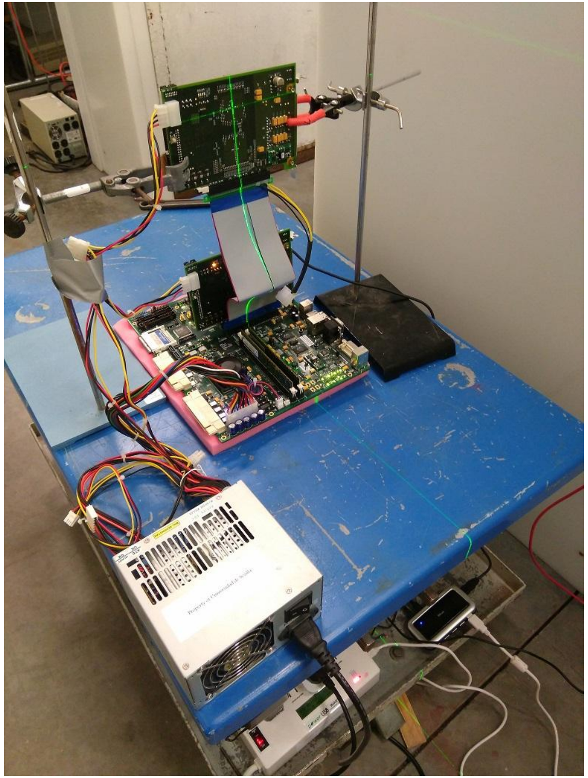
- Bridging the gap between F.I. & radiation
  - F.I. does not pretend to substitute radiation testing
- It is important for us to characterize FTU2
  - Determine which conditions are needed to get the maximum accuracy with respect to radiation testing, and which accuracy is that.
  - Implemented full FPGA **readback** capability -> get number of upsets in FPGA
- First time to test FTU2 with actual radiation.

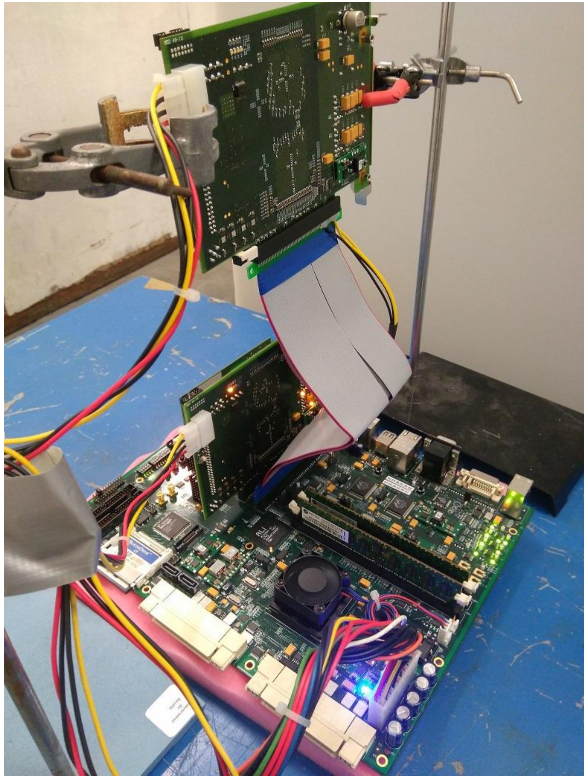
Beam experience

Los Alamos Neutron Science CEnter (**LANSCE**)

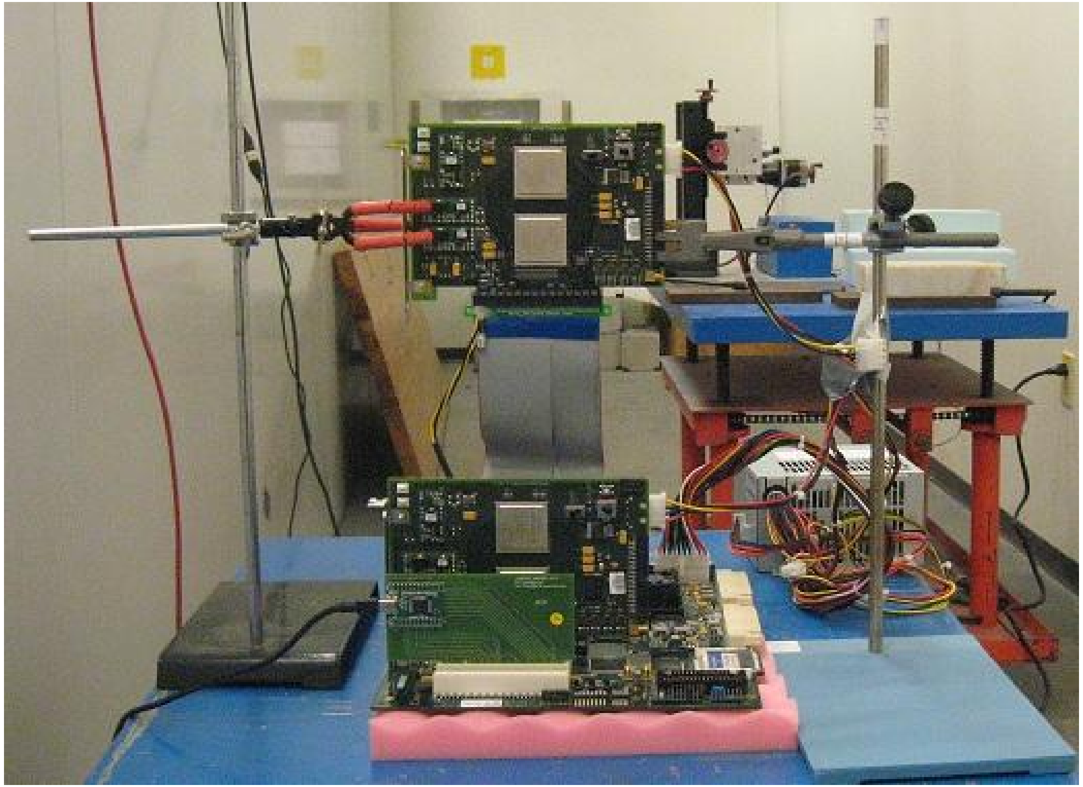
- New Mexico, November 2016
- Weapons Neutron Research facility (WNR)
    - Neutron beam: Tungsten spallation source
    - The shape of the neutron spectrum is very similar to the produced in the atmosphere by cosmic rays (scaled)
    - The DUT was placed at 25.8 m from the neutron source
    - We obtained a spot with 40 mm of diameter
    - A neutron flux of $2.0 \cdot 10^5$ n/(s·cm^2), above 10 MeV, has been obtained
    - The total fluence per experiment was $6.9 \cdot 10^8$ n/cm^2

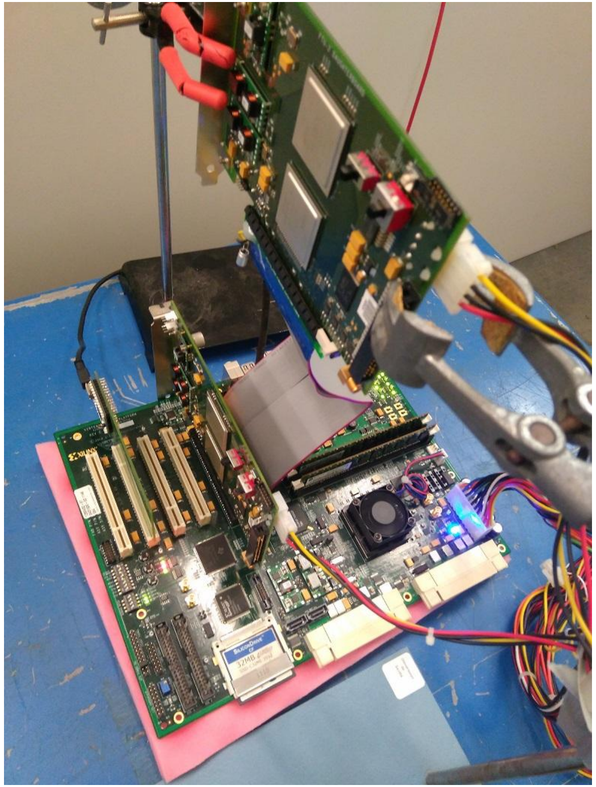
Beam experience





Beam experience





Beam experience

We developed a running script for the experiment.

- FTU2 allows scripting
- Disable injecting of faults (SEU will occur naturally)
- Control the platform remotely with a portable computer
- Control the power source with a USB controlled power strip connected to the computer

Beam experience

The Test Loop

- Shut the platform down and then power it on again every hour
    - Assure no uncorrectable SEU in the FPGA can produce a long-lasting Single Event Functional Interrupts (SEFI)
- Each hour, it has executed 750 ‘runs’
    - It made a readback after every ten runs
    - After each readback, the target FPGA was reconfigured and the design was reset

Beam experience

Results: LANSCE

- Discrepancies between each readback and the golden readback
  - With output errors
    - Generated an average of 7 output errors
  - Without output errors
    - Calculated SEU/s rate
- Selected lots (1 lot = 10 runs) without output errors
  - 13 SEU/lot or 1.3 SEU/run
  - Duration of 750 runs = 3471 seconds
  - 46.28 s/lot or **0.28 SEU/s**

Beam experience

FTU2

- Objective:
  - Reproduce the radiation experiment using FTU2 (same hardware, change beam for fault injection)
- Issue:
  - Not all runs under the beam involved an actual SEU on the essential bits of the target FPGA
  - FTU2 can only inject safely on essential bits. Addressable configuration locations not marked as essential bits are not guaranteed to physically exist on the device

Beam experience

FTU2

- Action: We had to scale the SEU/run rate that we were going to inject into the essential bits of our design
  - FX70T FPGA model - ebd (essential bit Data) file: 18936096 essential bits
  - Our design consists of 2715926 essential bits
    - 3% belong to user bits
    - 97% belong to configuration bits
  - There is only a 14% probability that an SEU will occur over the essential bits of our design

Beam experience

Results: FTU2

We obtained an average of 6 output errors

We obtained a lower SEU/lot rate than in the radiation test

These results can be improved in two ways:

1. We could modify our platform to inject randomly over all the configuration bits of the FPGA
2. We could mask the essential bits of the radiation test readbacks and then only select those discrepancies that affect essential bits of the design

Conclusions (Beam experience)

- A new experience for a beam-able fault injection platform. It seeks to analyze the similarity of the results of the injection of faults with reality
- We need to perform more experiments to apply statistical data and increase confidence in the results
- We have discovered new ideas on how to improve our fault injection platform to recreate the radiation test
- The experience makes us reinforce the idea of being able to use FTU2 as an analysis tool prior to performing a radiation test

Contents

- Introduction
- Usability improvements
- Bug fixes
- Extension of injection coverage
- Analog FTU integration
- Beam experience
- **Conclusions and future work**

Conclusions

- Improvements and bug fixes
  - Keep the tools updated, continue improving & adding functionality, support users
- Extension of injection coverage and scrubbing emulation
  - Allows F.I. testing of new circuits + schemas (microprocessors, partial reconfiguration)
- AFTU easier to use & integrated in the same graphical interface
  - Also supports more technologies
- Beam-able F.I. platform: ease reproducibility of radiation tests
  - Towards characterization of how similar F.I. is to radiation and what conditions are needed

Future work

- Speed up injection in Block RAMs
- Extend injection to distributed memories
- Full PCI express protocol for PC interoperability
- FTU-BRAVE (w/ NanoXplore NG-Medium)
- Daughter boards with external memory
- Improve statistics on fault injection vs radiation comparison (need more radiation data)

Q & A

Enquiries about usage / collaborations / research periods @unisevilla:

Hipólito Guzmán Miranda: **hguzman@us.es**

Extra slides

Scrubbing emulation

FTU2 allows working with partial bitstreams.

- Partial reconfiguration is done preserving user flip-flop contents

Can be used to:

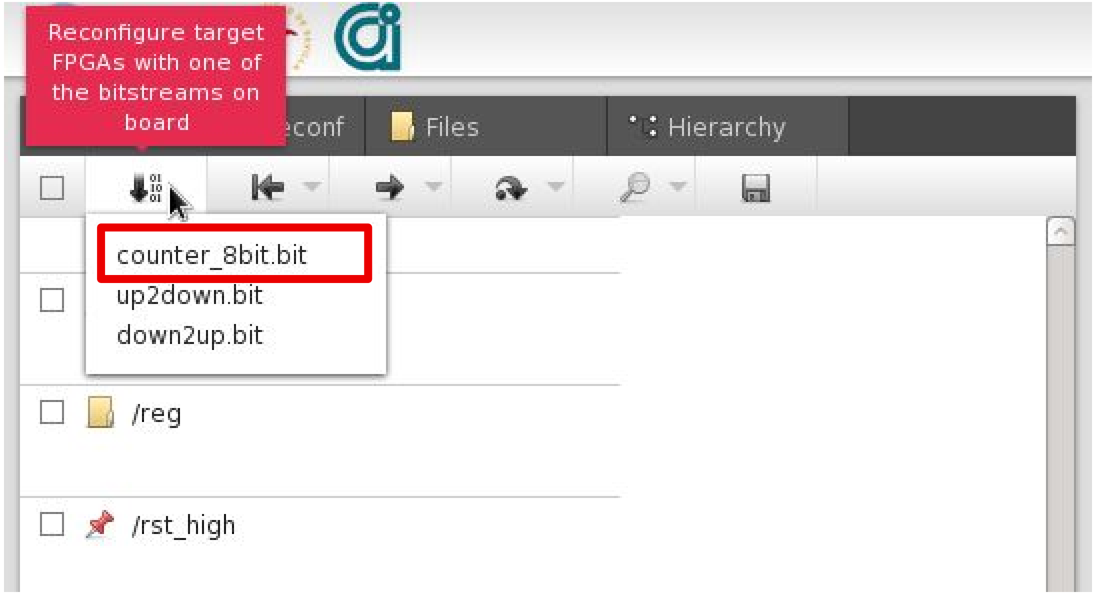
- Measure effectiveness of adaptive reconfiguration strategies
- Test scrubbing schemas

Working with partial bitstreams

Full bitstream

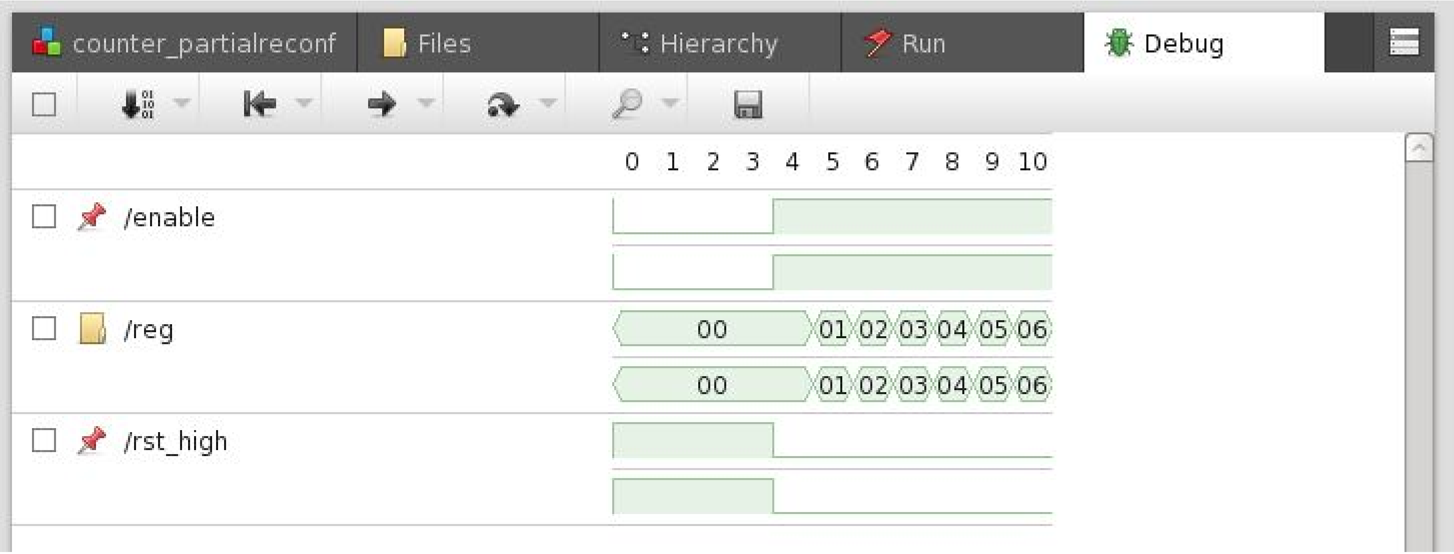
Partial bitstreams

Working with partial bitstreams



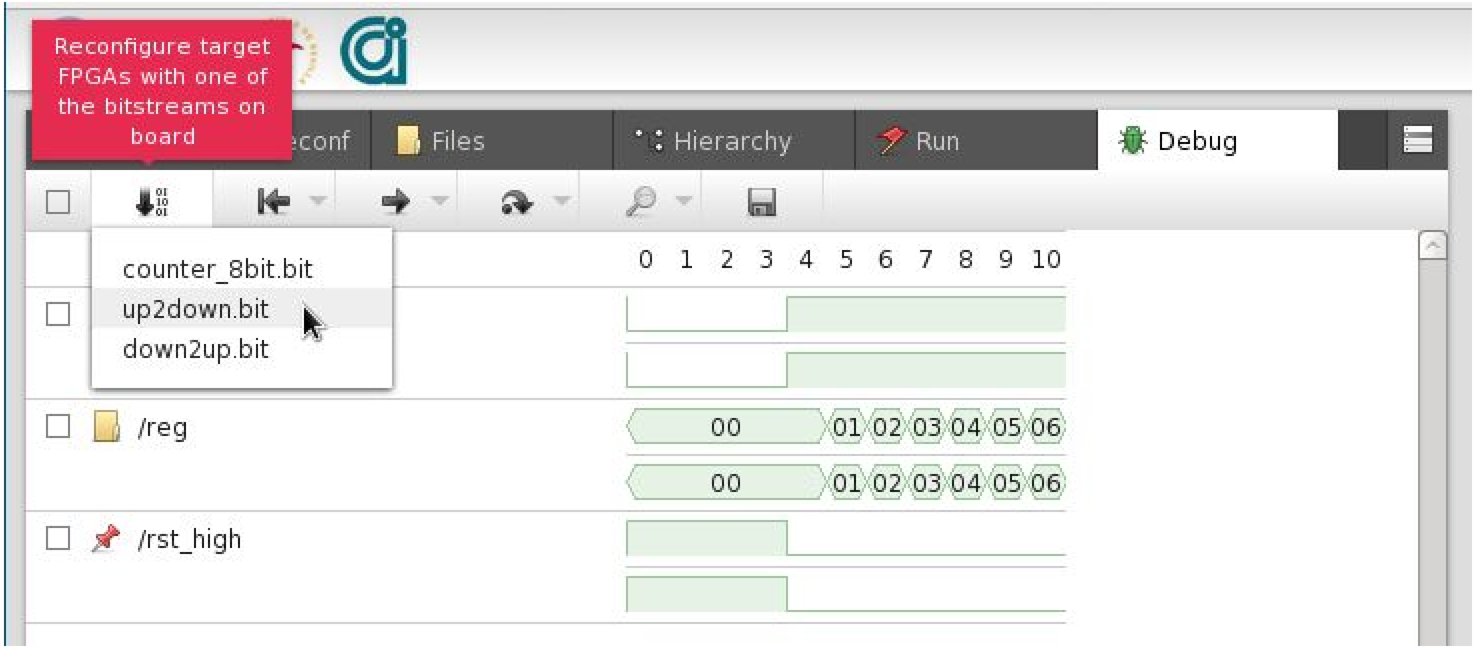
Configure with full bitstream

Working with partial bitstreams



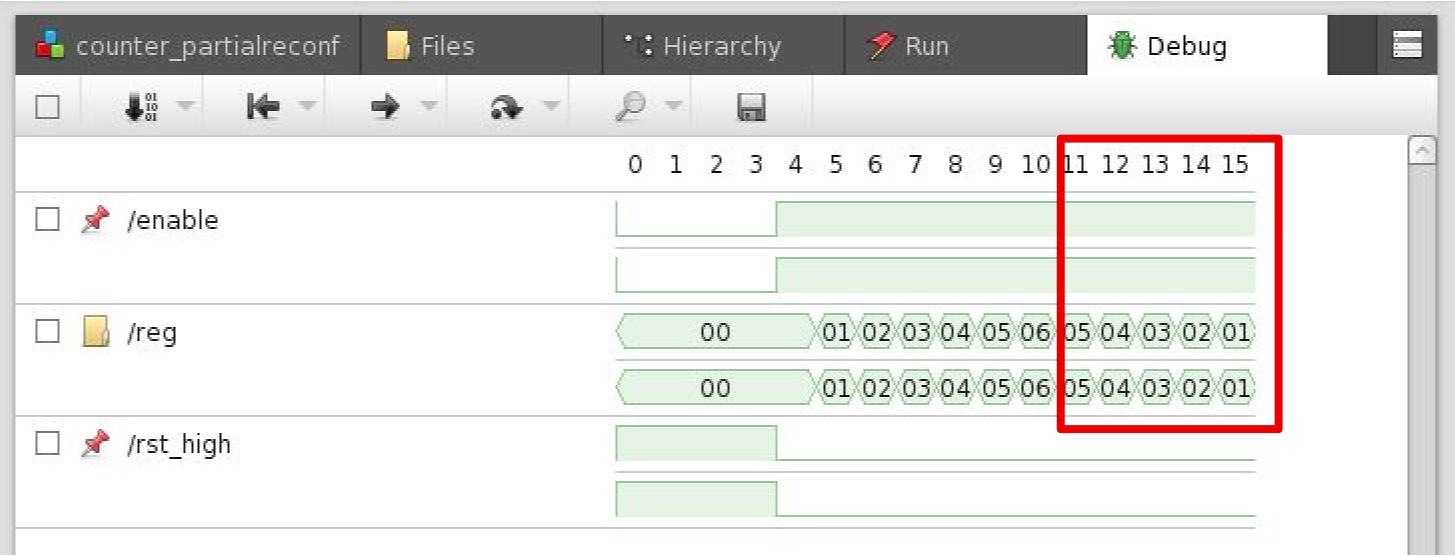
Design works as expected
(counter incrementing)

Working with partial bitstreams



Partial reconfiguration

Working with partial bitstreams



Counter is now decrementing instead

Design state is kept between reconfigurations