FT-UNSHADES2. Adaption of the platform for SRAM-FPGA reliability assessment

J.M. Mogollón, H.G. Miranda, M.A. Aguirre, J. Nápoles, J. Barrientos, L. Sanz

Electronic Engineering Dpt. School of Engineering.

University of Sevilla (SPAIN)

ESA/ESTEC. Workshop SEFUW. Nov 2012

Summary

1. FT-UNSHADES2 in a nutshell
2. The FTU2 design preparation flow
3. Connecting with STAR (Politecnico di Torino)
4. A new vision of the SRAM-FPGA fault injection:
   - Deterministic injection
   - Analysis in detail
   - Scrubbing policy analysis
5. FTU2 in numbers
6. Final remarks

FT-UNSHADES2 in a nutshell

- Fault injection is a method of assessing the effects of particles hitting over registers of a digital design.
- Radiation effects on configuration bits of FPGAs cause permanent faults. Injecting faults means the induction of changes in the configuration. The goal is to perform the injections in a deterministic manner.
- FT-UNSHADES2 assess the vulnerability of a design in an FPGA to SEE.
- It is not constrained to any FPGA
- Analyzes in detail the propagation of a fault in the configuration, and the possible corruption of the user logic.
- The study of techniques for scrubbing policy

Other features:

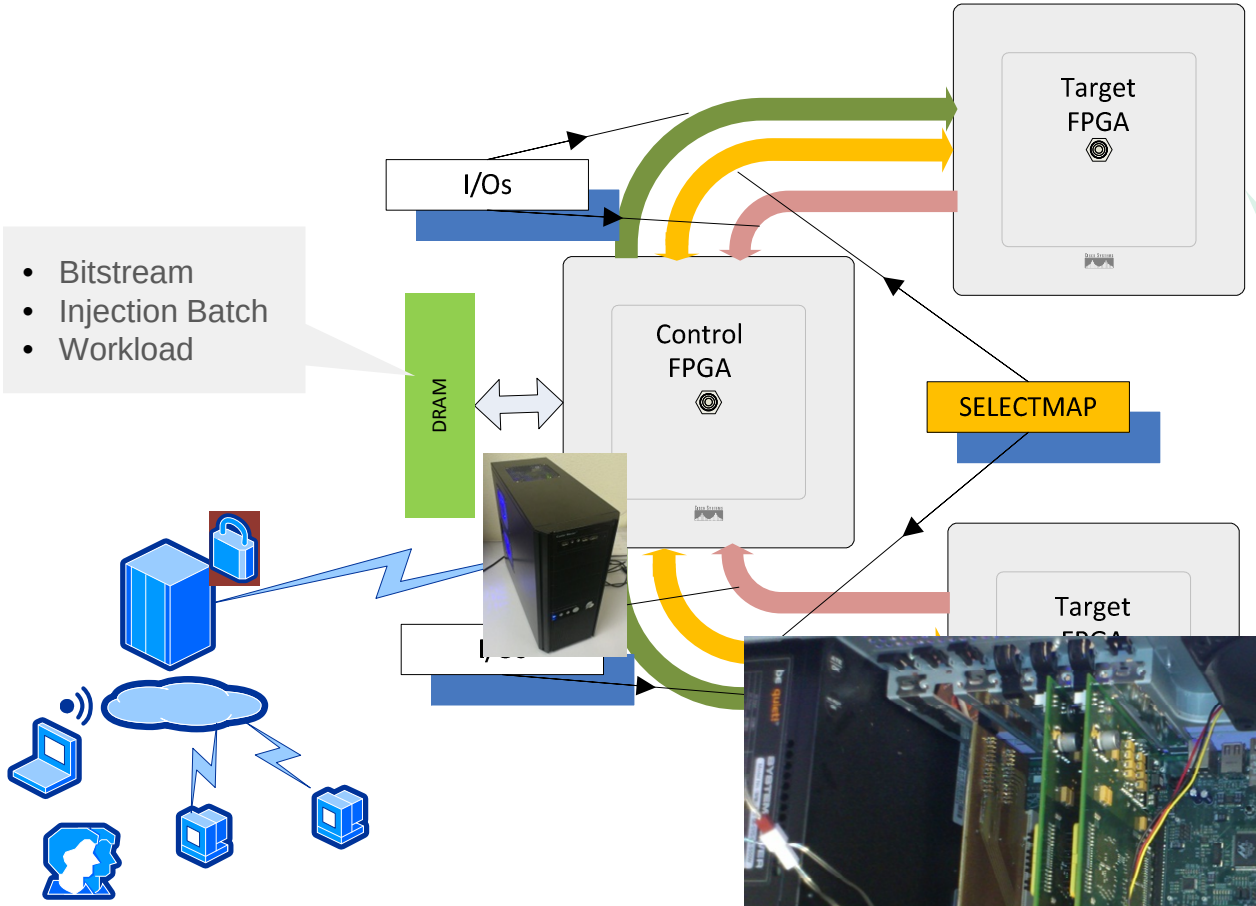
- Remote access
- Detailed analysis
- Test in the beam

FT-UNSHADES2 in a nutshell

- FTU2 is a contract with ESA: "High Capacity, High Speed IC Test System for Automatic Fault Injection and Analysis (FT-UNSHADES 2)"
- FTU2 is a platform designed to assess the reliability of a netlist at design level.
- The PARTIAL RECONFIGURATION feature of Xilinx FPGAs is exploited to generate bit-flips into the USER REGISTERS (…not only)
- The process is designed knowing a priori where, when and how to inject the faults.
- In FTU2 both sights are implemented: fault campaigns and single fault detailed analysis
- Several SEEs are implemented: SEU and MBU, and under certain conditions, SETs.
- The design flow has been is reduced to a FPGA standard implementation and a simulation.

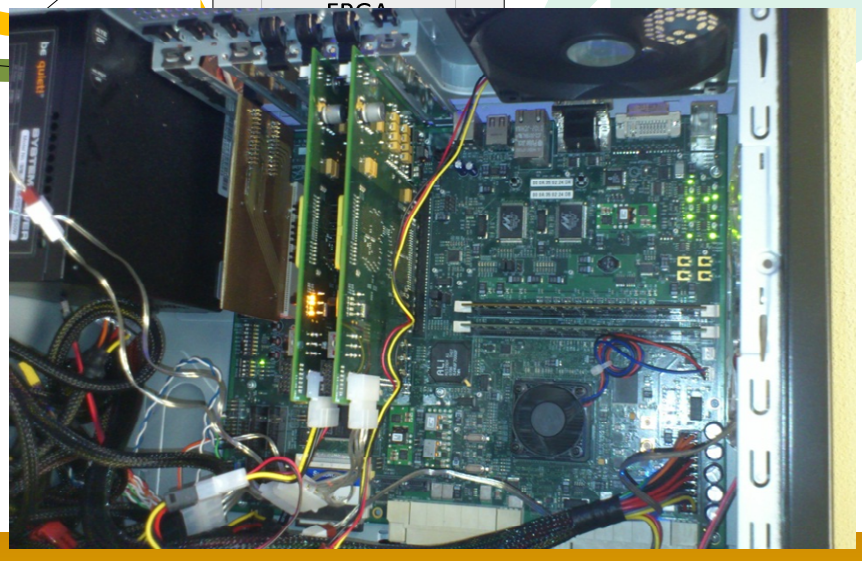
Originally FTU was thought trough the assessment of ASIC netlist. FTU2 has been extended to inject on SRAM-FPGAs configuration.

FT-UNSHADES2 in a nutshell: the hardware structure

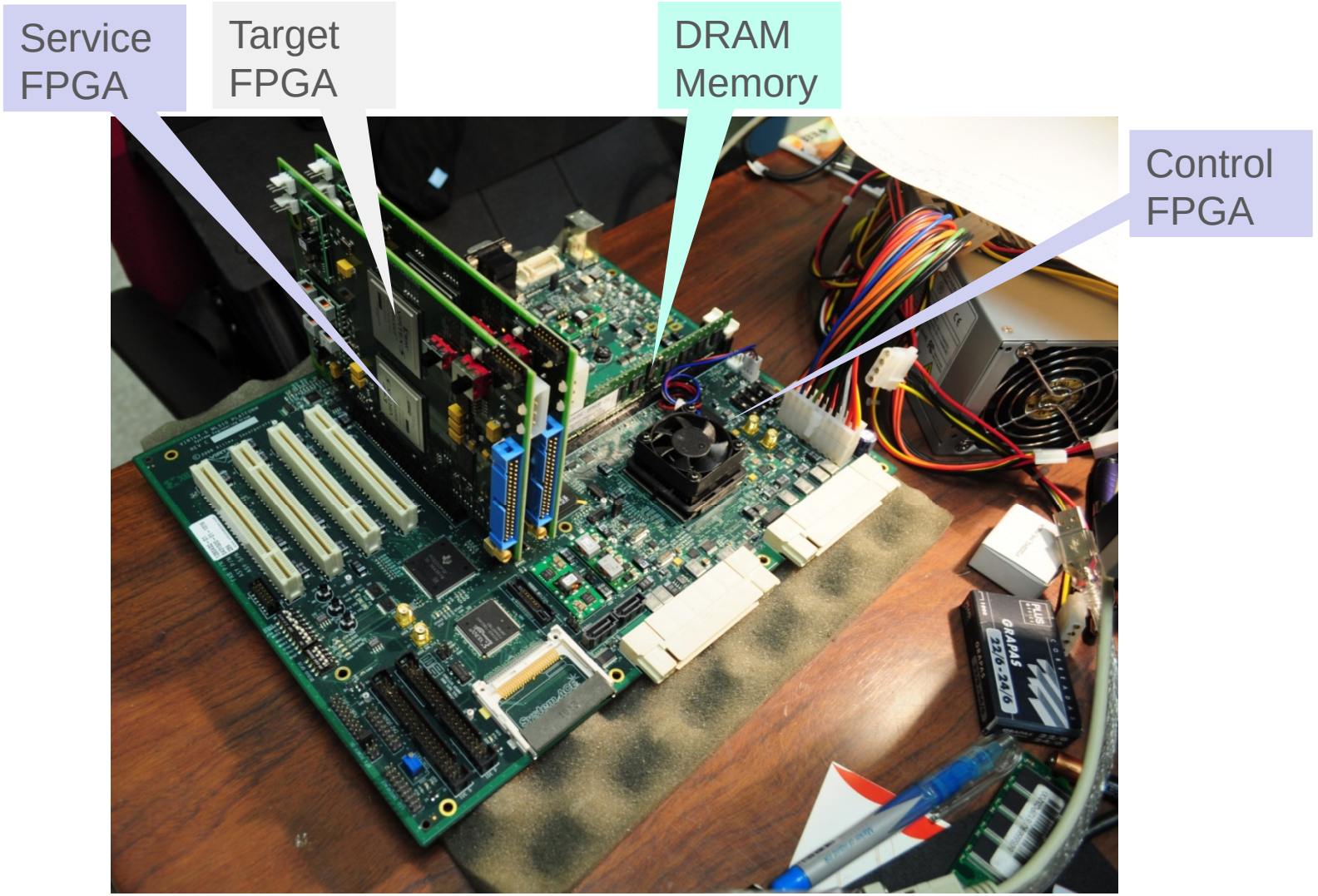


- Bitstream
- Injection Batch
- Workload

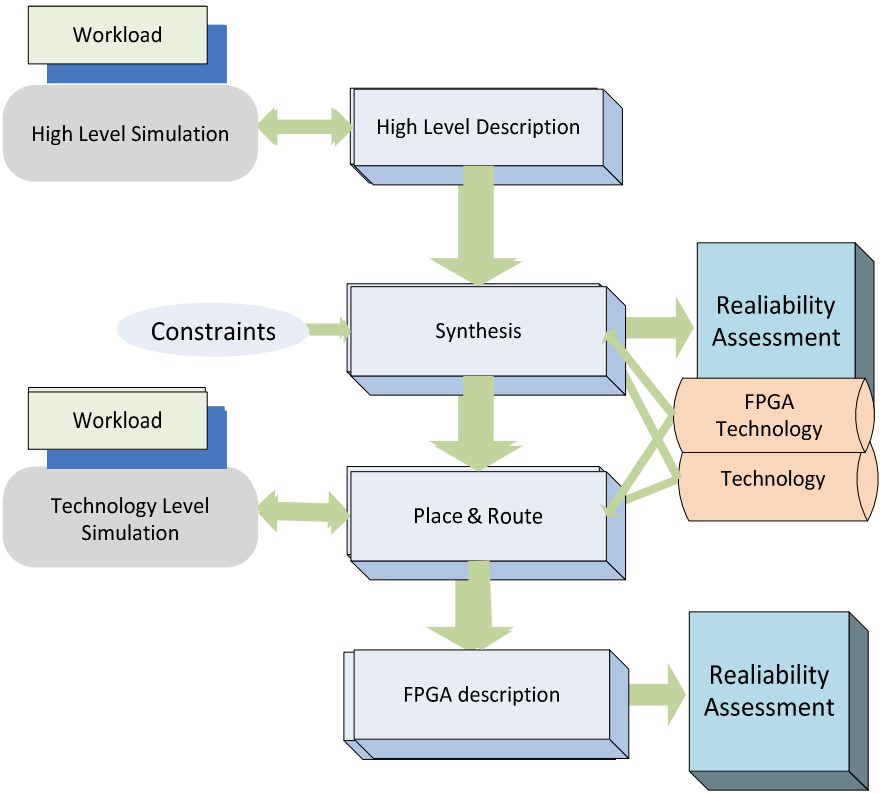
Two twin FPGAs hosts two copies of the netlist. The design runs along in parallel



FT-UNSHADES2 in a nutshell: the hardware structure



FT-UNSHADES2 in a nutshell: the design preparation flow



- Bitstream (.bit)
- Bit allocation file (.ll)
- Port location (.pin)
- VCD stimuli (*.vcd)

For FPGA flow

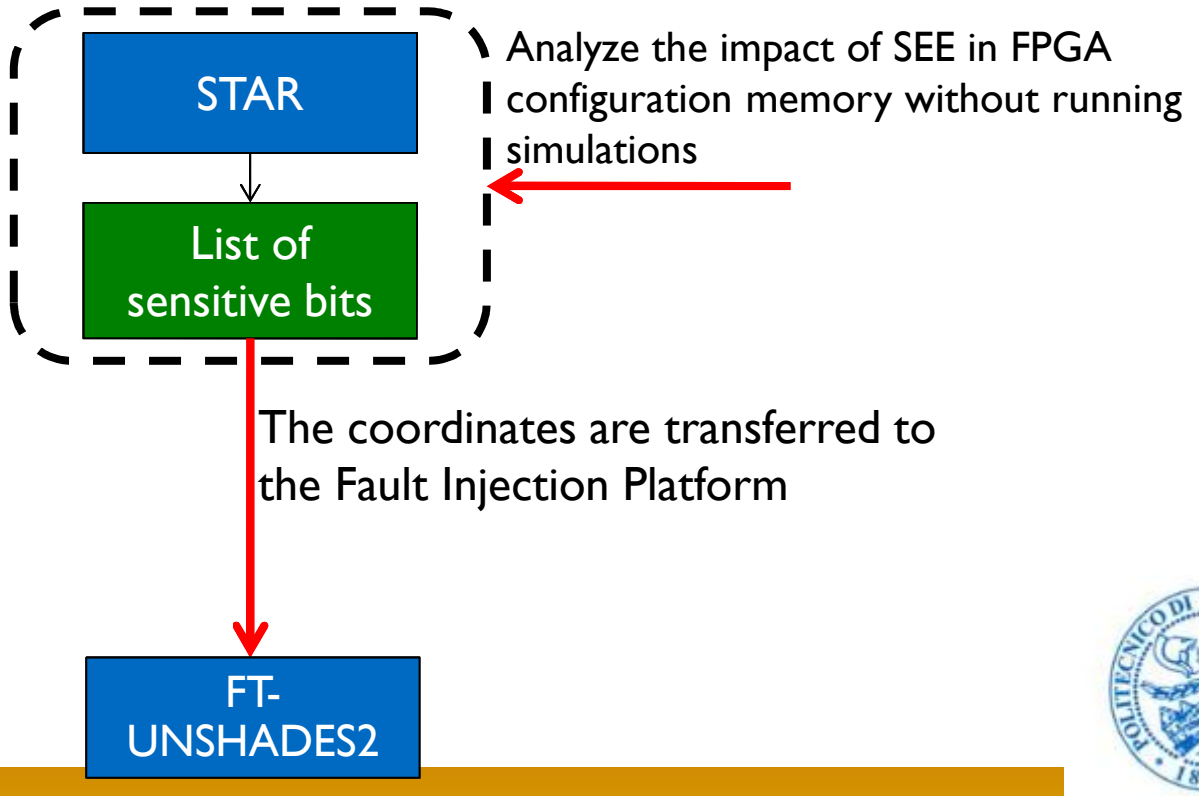
- **Configuration allocation (*.cl)**

1. Fit your design in the target FPGA
2. Allocate I/Os
3. Simulate and extract inputs
4. Finish the standard flow

FT-UNSHADES2 and STAR

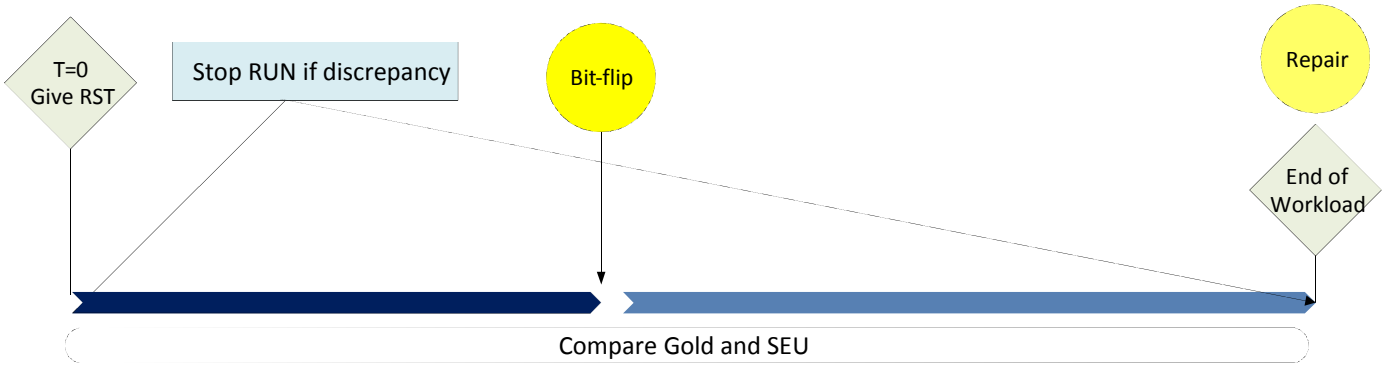
1. STAR analyzes the allocation of the design in the FPGA.
2. Determines the most critical configuration bits.
3. Generates the coordinates of those critical bits.

```
Bit  96615 0x000c0400  199 Block=PIP_Conf  Net=/.config/Frame199/ISPFIFO/mem3(3)
```

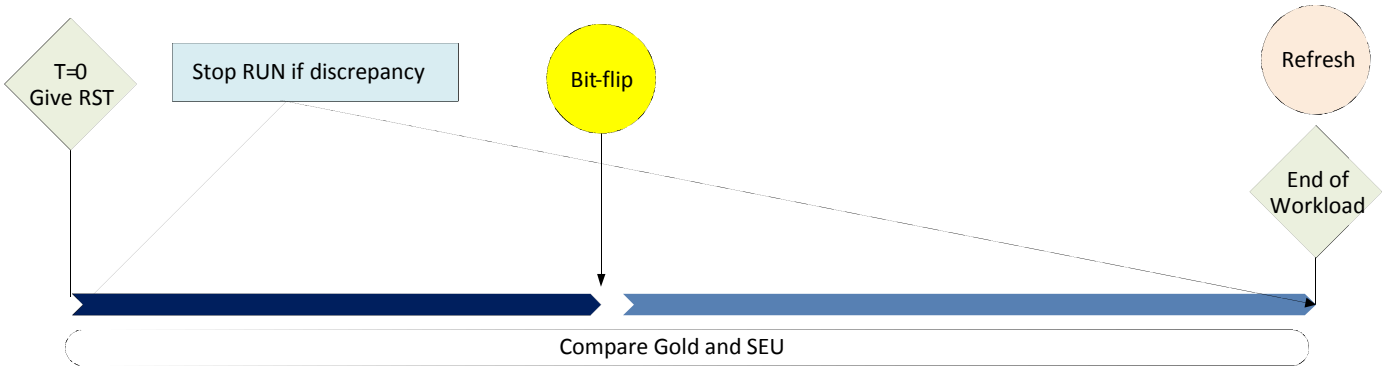


A new vision of the SRAM-FPGA fault injection

- Inject and repair at end

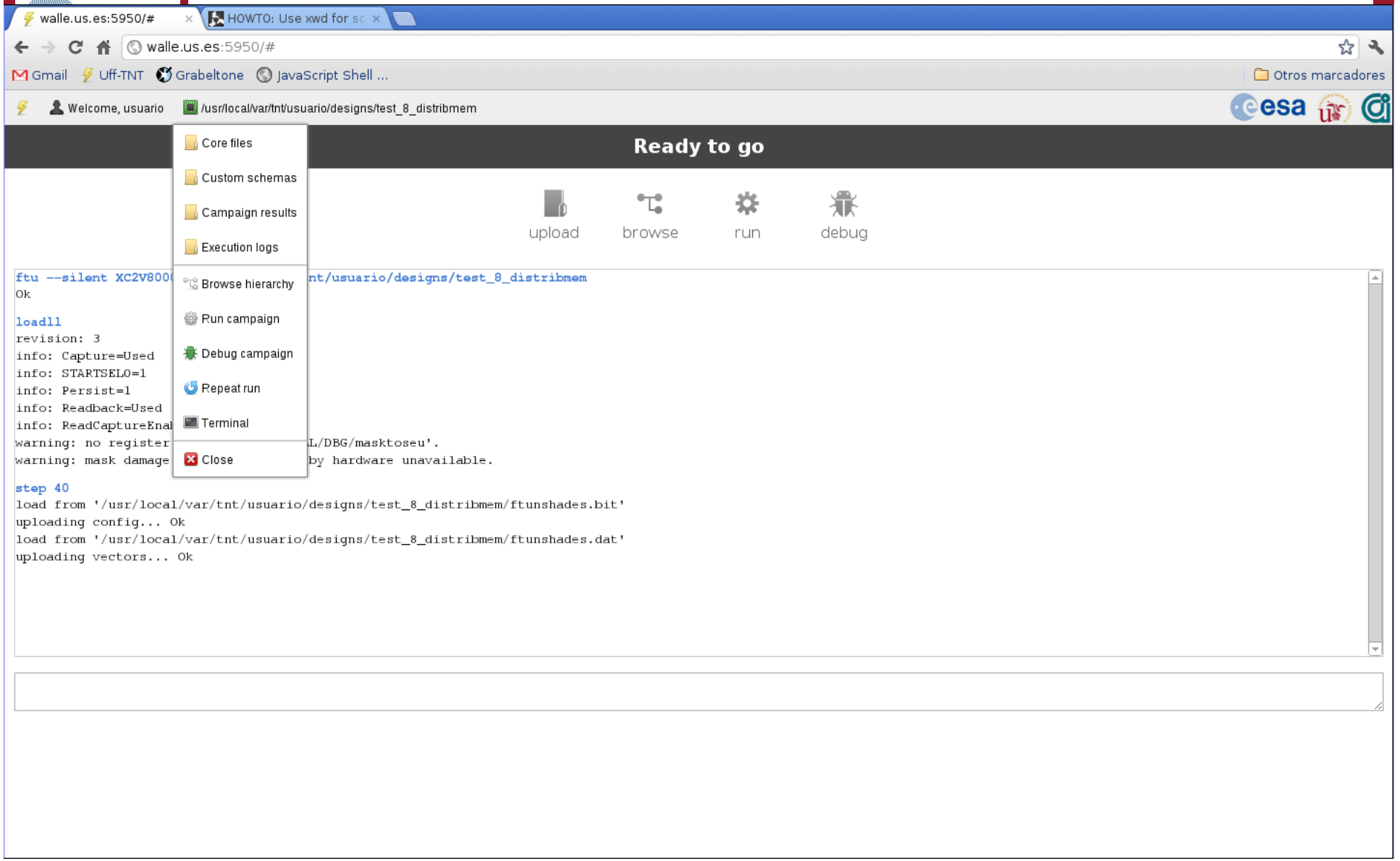


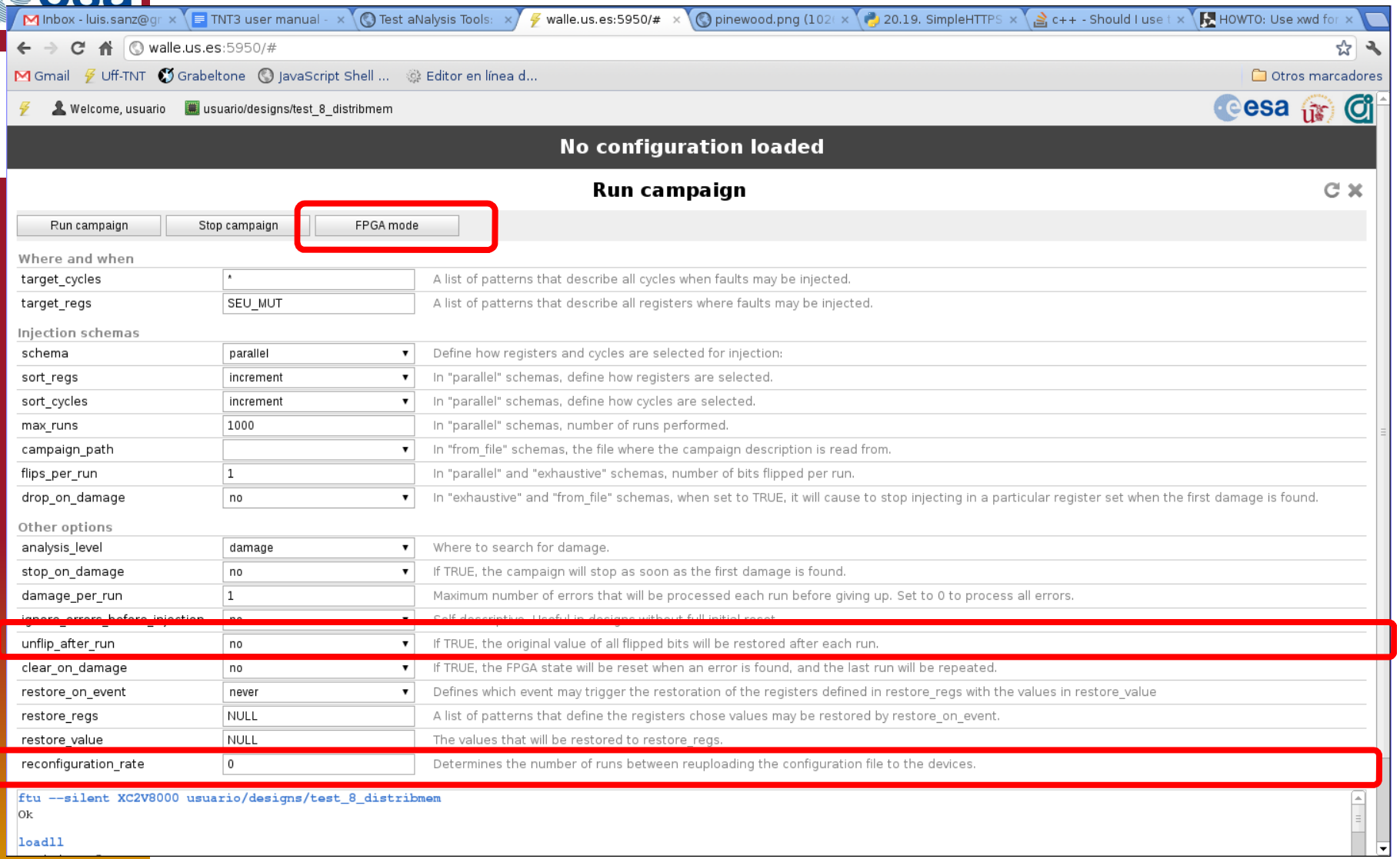
- Full configuration refresh

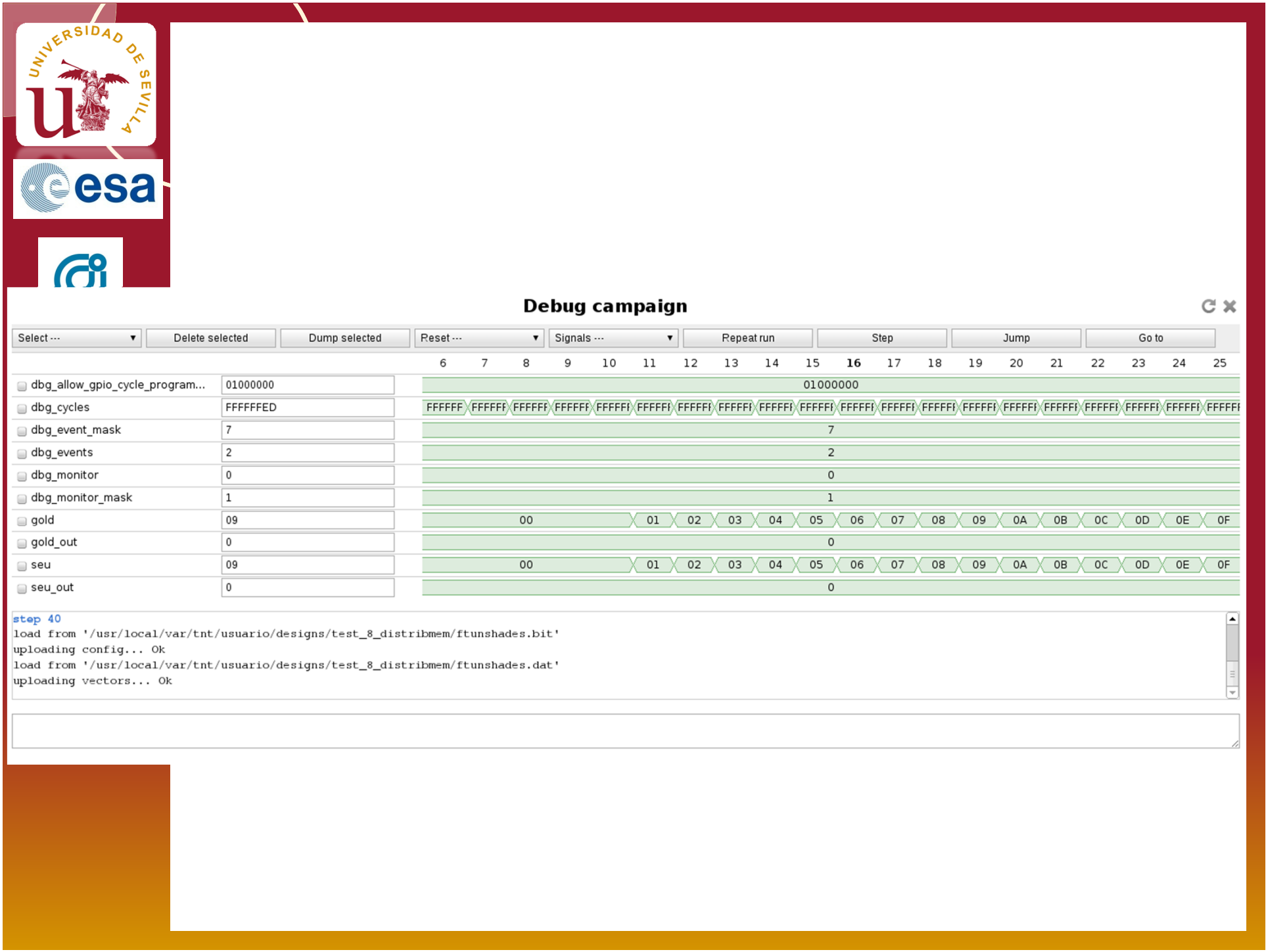


- Define a technique with both approaches

FTU2: User interface




walle.us.es:5950/#
Gmail
Uff-TNT
Grabeltone
JavaScript Shell ...
Editor en línea d...
Otros marcadores
Welcome, usuario
usuario/designs/test_8_distribmem
No configuration loaded
Run campaign
Run campaign
Stop campaign
FPGA mode
Where and when
target_cycles
*
A list of patterns that describe all cycles when faults may be injected.
target_regs
SEU_MUT
A list of patterns that describe all registers where faults may be injected.
Injection schemas
schema
parallel
Define how registers and cycles are selected for injection:
sort_regs
increment
In "parallel" schemas, define how registers are selected.
sort_cycles
increment
In "parallel" schemas, define how cycles are selected.
max_runs
1000
In "parallel" schemas, number of runs performed.
campaign_path
In "from_file" schemas, the file where the campaign description is read from.
flips_per_run
1
In "parallel" and "exhaustive" schemas, number of bits flipped per run.
drop_on_damage
no
In "exhaustive" and "from_file" schemas, when set to TRUE, it will cause to stop injecting in a particular register set when the first damage is found.
Other options
analysis_level
damage
Where to search for damage.
stop_on_damage
no
If TRUE, the campaign will stop as soon as the first damage is found.
damage_per_run
1
Maximum number of errors that will be processed each run before giving up. Set to 0 to process all errors.
unflip_after_run
no
If TRUE, the original value of all flipped bits will be restored after each run.
clear_on_damage
no
If TRUE, the FPGA state will be reset when an error is found, and the last run will be repeated.
restore_on_event
never
Defines which event may trigger the restoration of the registers defined in restore_regs with the values in restore_value
restore_regs
NULL
A list of patterns that define the registers chose values may be restored by restore_on_event.
restore_value
NULL
The values that will be restored to restore_regs.
reconfiguration_rate
0
Determines the number of runs between reuploading the configuration file to the devices.
ftu --silent XC2V8000 usuario/designs/test_8_distribmem
Ok
loadll


Debug campaign
Select ---
Delete selected
Dump selected
Reset ---
Signals ---
Repeat run
Step
Jump
Go to
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
dbg_allow_gpio_cycle_program... 01000000
dbg_cycles FFFFFFED
dbg_event_mask 7
dbg_events 2
dbg_monitor 0
dbg_monitor_mask 1
gold 09
gold_out 0
seu 09
seu_out 0
step 40
load from '/usr/local/var/tnt/usuario/designs/test_8_distribmem/ftunshades.bit'
uploading config... Ok
load from '/usr/local/var/tnt/usuario/designs/test_8_distribmem/ftunshades.dat'
uploading vectors... Ok

FTU2 in numbers

1. Fault rate achieved: 10.000 faults/sec (in a very easy example, ASIC mode)
2. 512 I/Os, the capacity depends on the model of the target device.
3. Current configurations Virtex 5, LX50T and FX70T (FF1136 package)
4. Latent analysis and selective *diff* options are available

Extensions of FTU2 FPGA flow

1. Changing the device/family is a matter of assembling new customized boards
2. Any scrubbing technique can be emulated in the platform with low effort
3. The platform is suitable to be used as test fixture in a beam test. The comparison is made with a shielded unit.
4. Moving to a non-Xilinx FPGA is possible if partial reconfiguration port is available and a non-intrusive READBACK mechanism exists.

Conclusions & Open questions

- FTU2 extends its capabilities to the study and assessment of designs in SRAM-FPGA
- Provides exhaustive information of campaigns (gross numbers) and analysis of single faults (detailed propagation)
- Hierarchical information of the faults and criticality is also provided.

Industry would tell us their needs. The complete **know-how** is in the **UoS**, and could be adapted.

Access to FTU2

For Academia:

- Access is available to academia for research purposes via internet.
- User/Password is given
- Beta users coordinated by ESA

For industrial customers:

- An agreement with UoS (and Polito)
- Training and support
- Install the FTU2 in the intranet

Demo of the system in SEFUW

Thank you for your attention

Q &A

Contacts:

aguirre@gie.esi.us.es
hipolito@gie.esi.us.es
jmmogollon@gie.esi.us.es
Luca.sterpone@polito.it
david.merodio.codinachs@esa.int