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

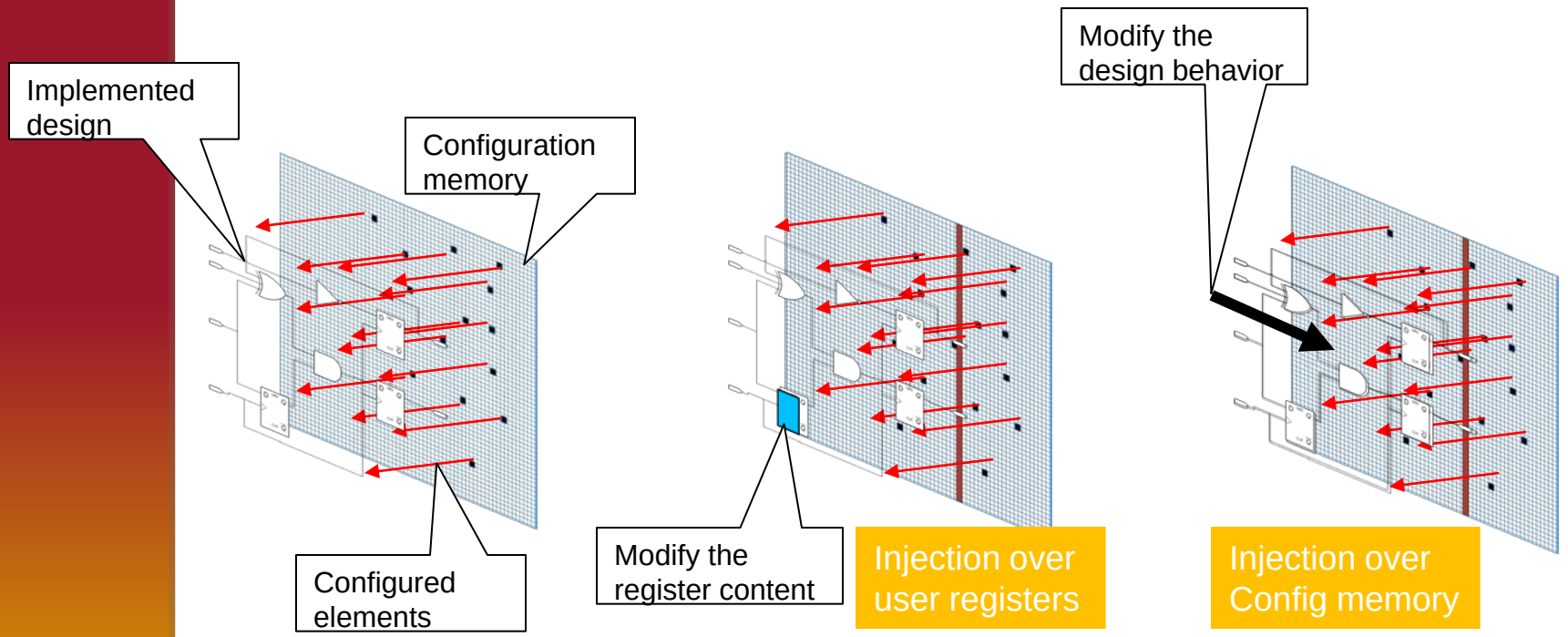
FT-UNSHADES

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 trough 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

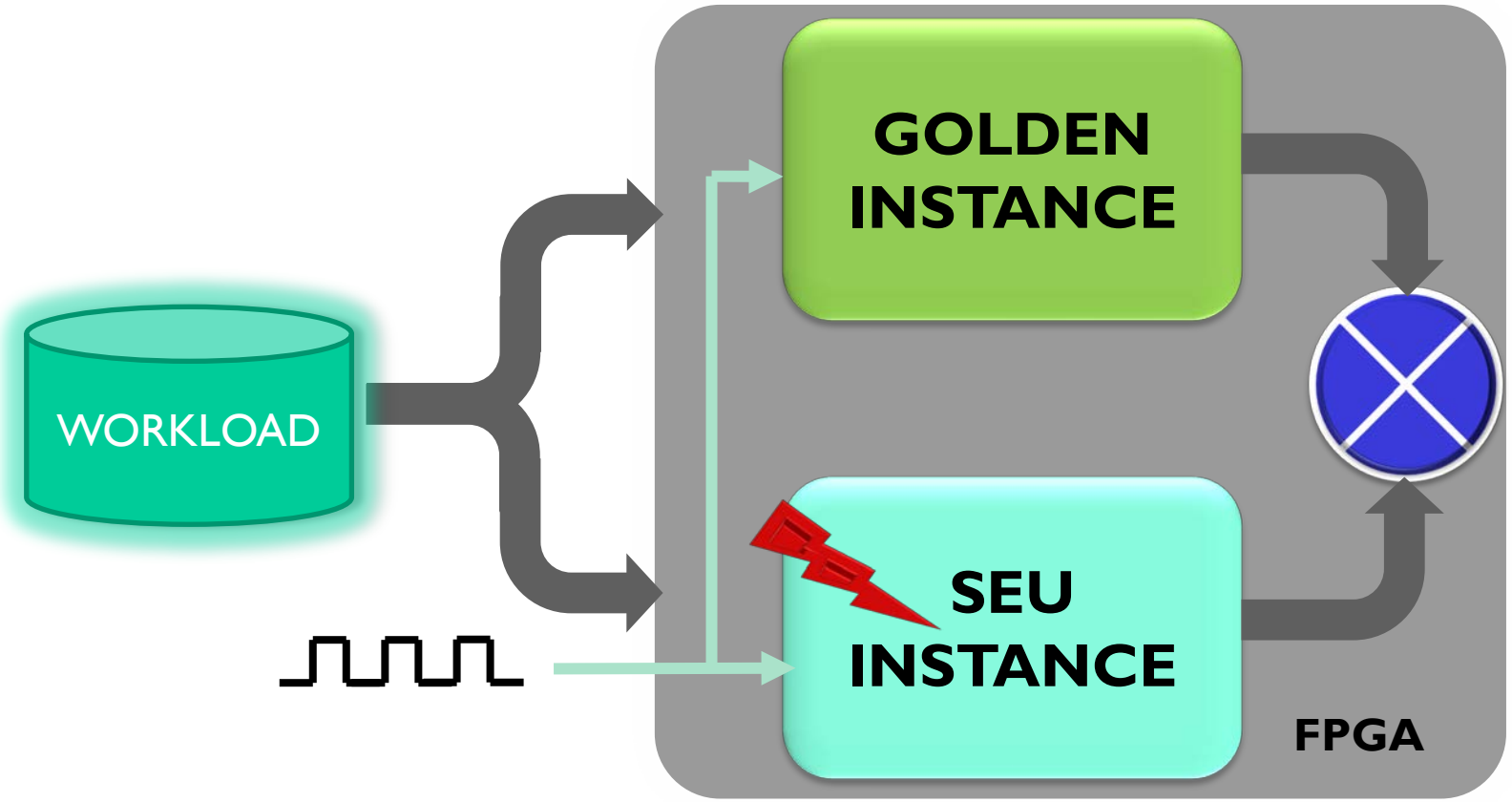


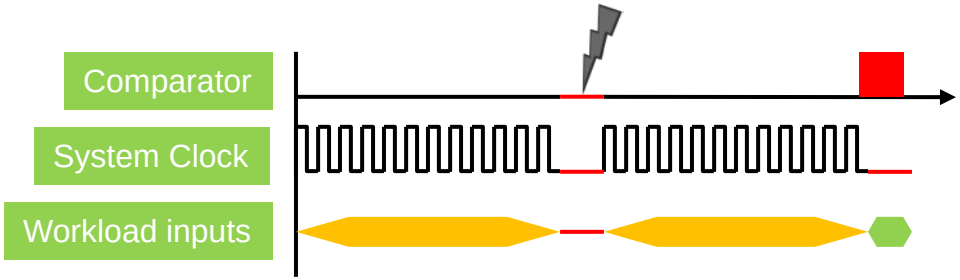
Quick Overview of FTUNSHADES

- 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

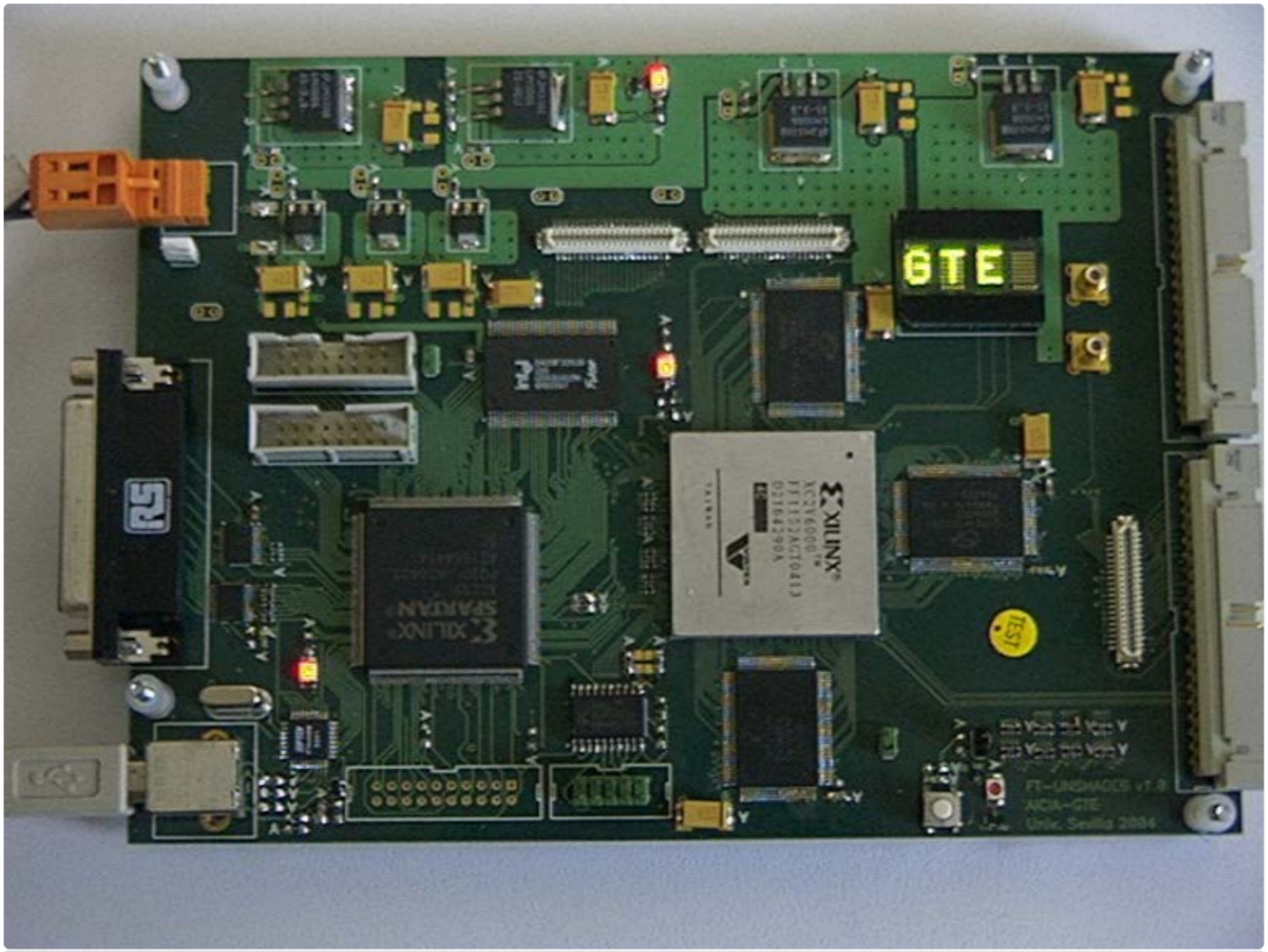
* in this context ,**observability** means accessing to the whole registers of the design at every clock cycle. **Controlability** is the capability to reproduce an event in the circuit at any time of the workload.

Quick View of FTUNSHADES

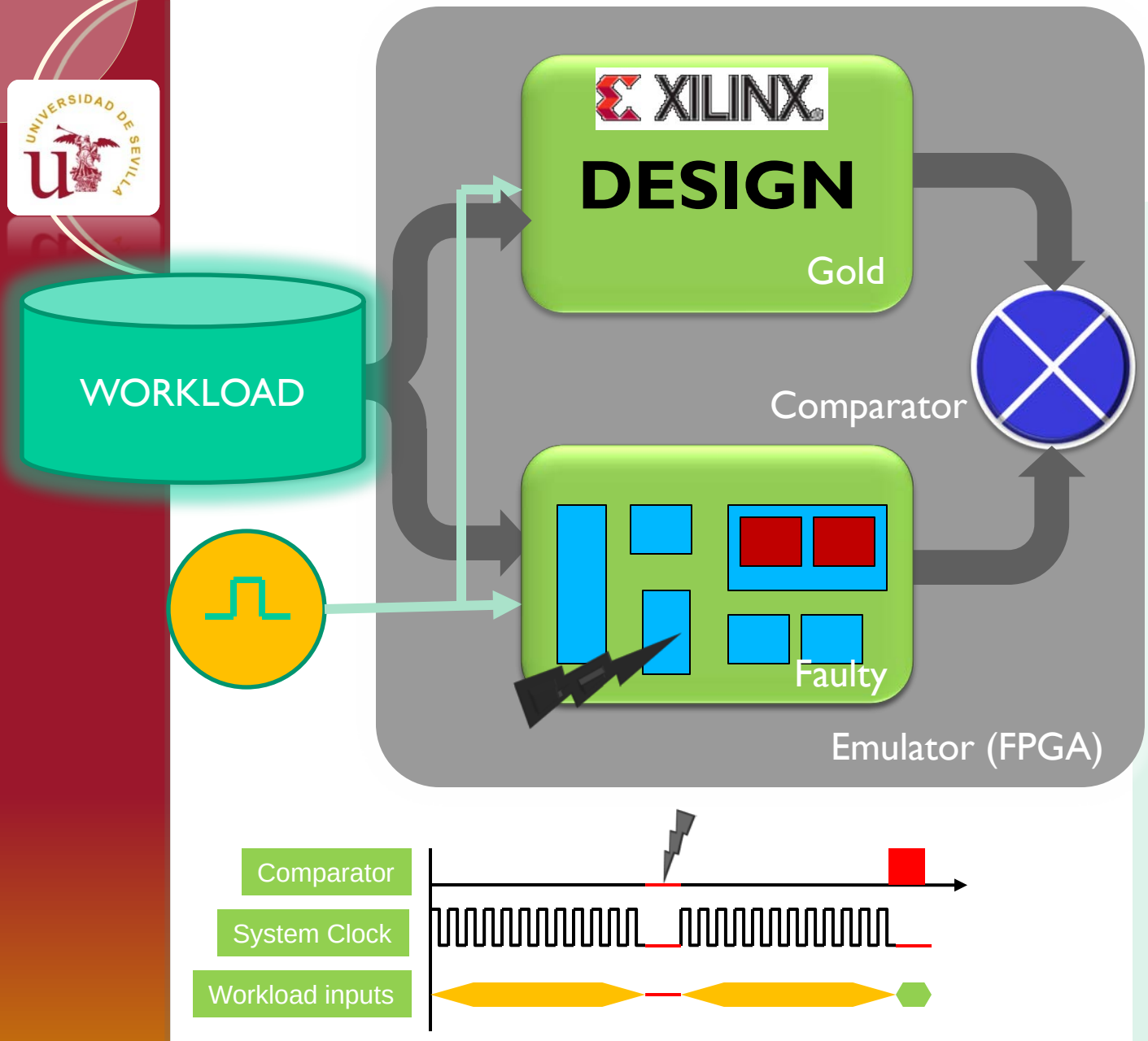




Quick View of FTUNSHADES



Standard exec model

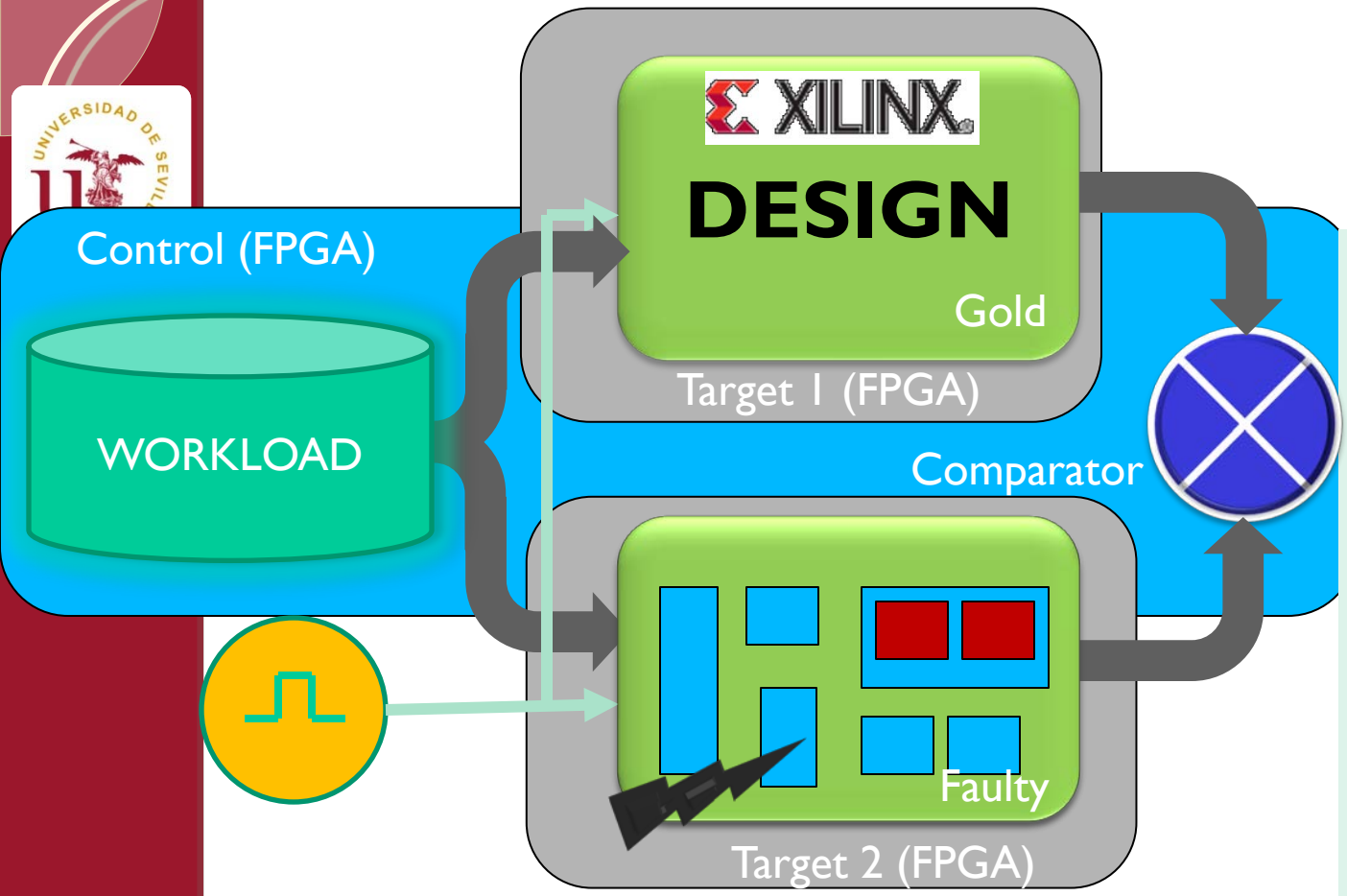


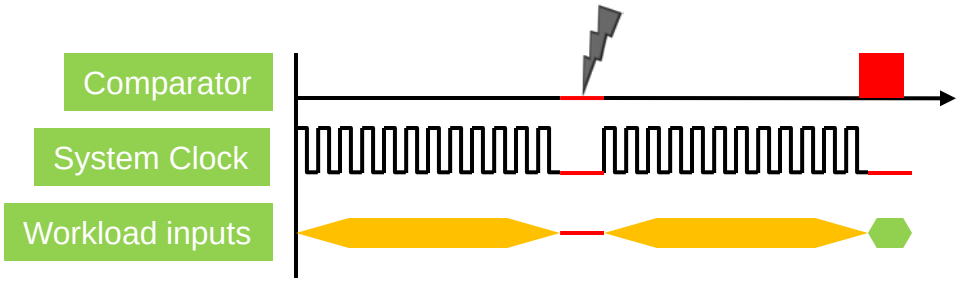
- This model is similar to a system in an accelerator.
- Two identical instances of the design are implemented in one FPGA
- The inputs are stored in external memories and the outputs are compared between both instances
- The system clock is common to each instance. Both instances work in parallel
- 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

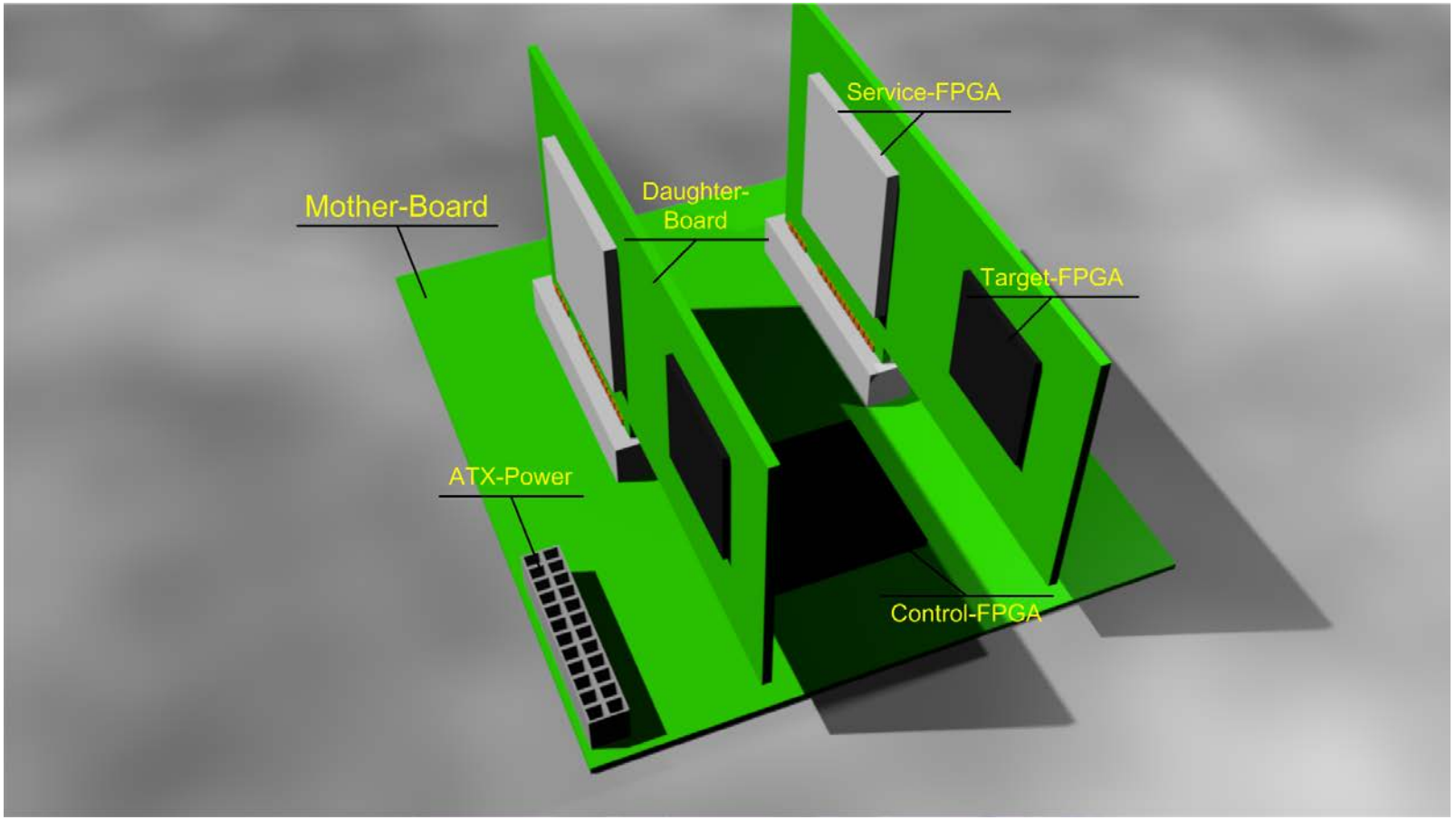
Standard model



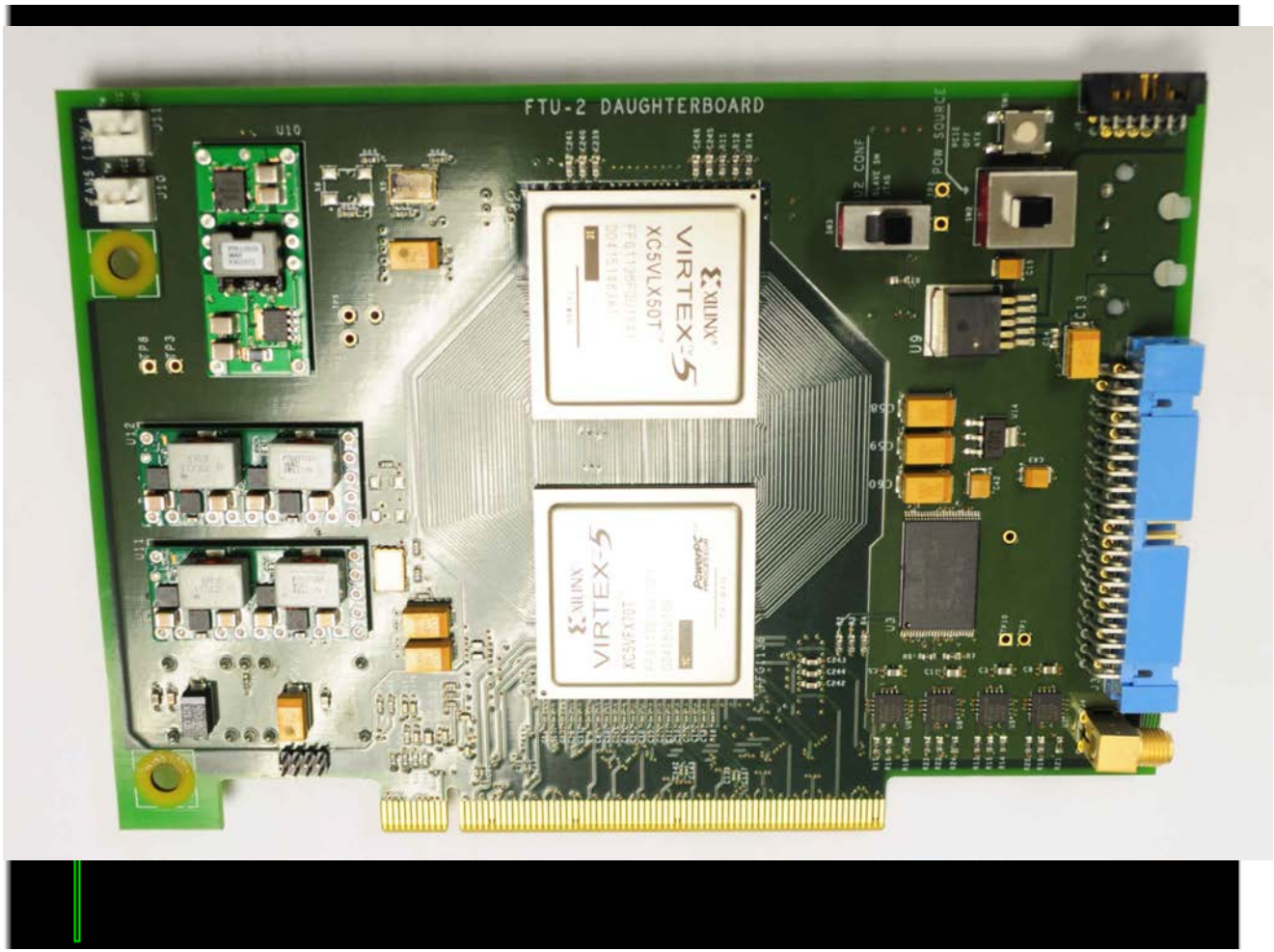


- This model is similar to a system in an accelerator.
- Two identical instances of the design are implemented in different FPGAs
- The inputs are stored in and external DDR external memory and the outputs are compared in the control FPGAs
- The injection is always to the faulty FPGA. The gold one works 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

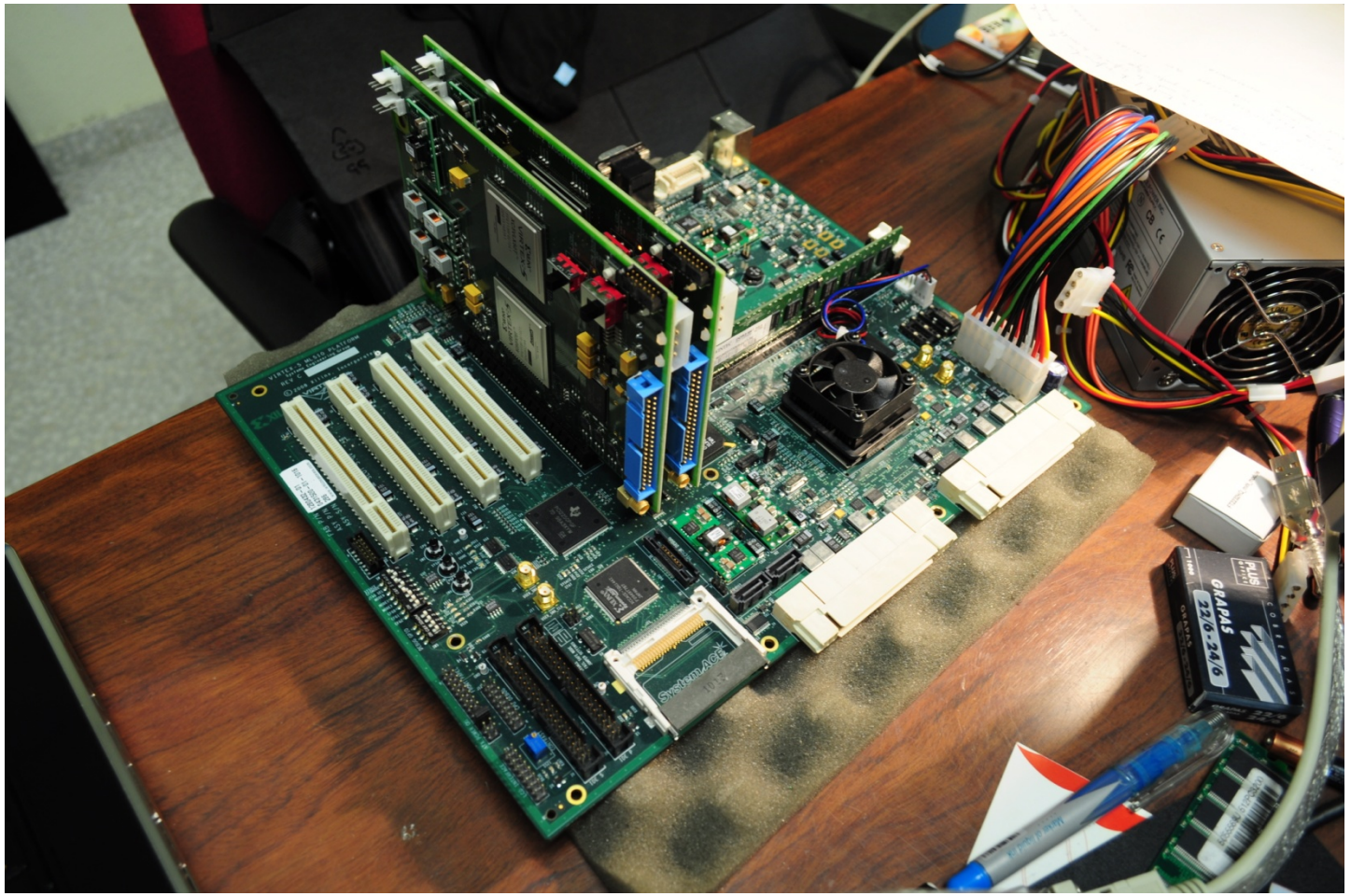
FTU2-DB Approach

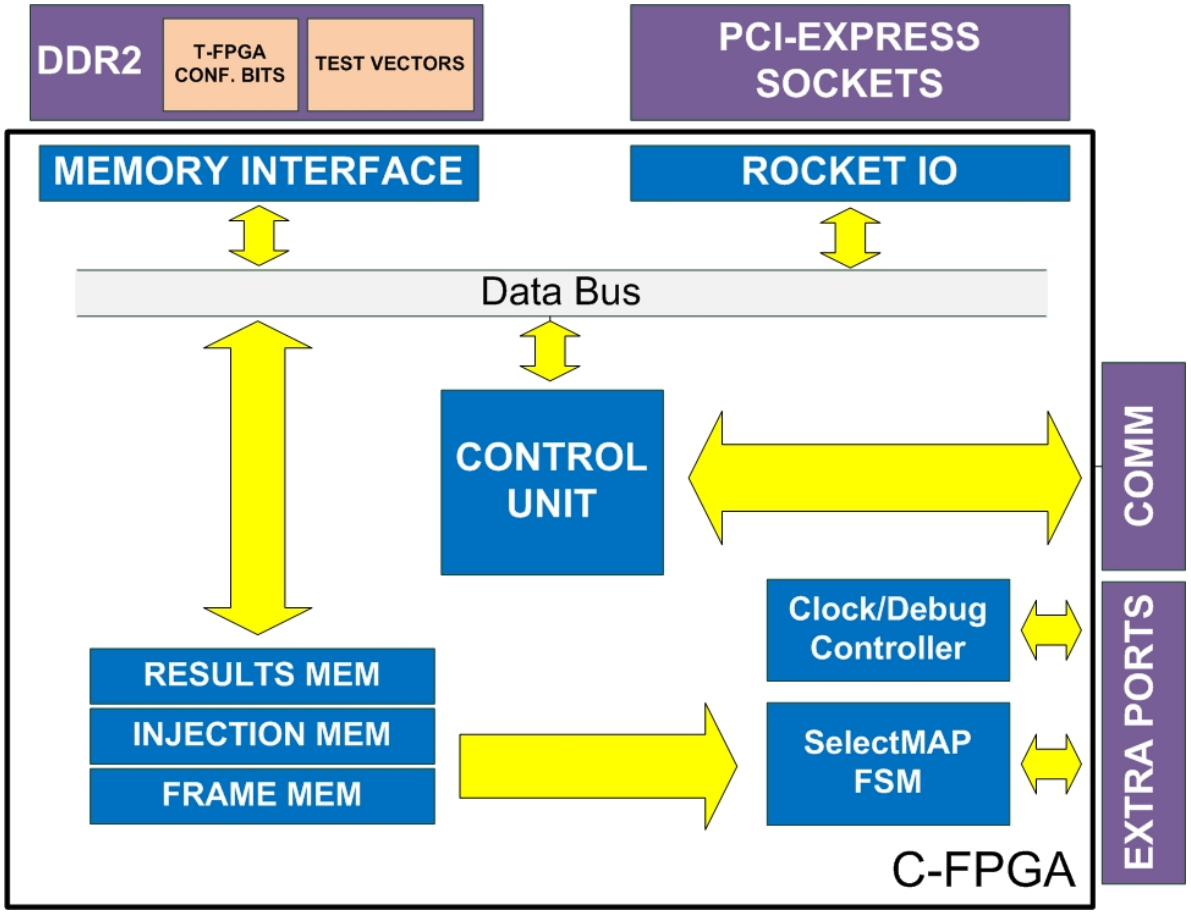


FTU2-DB Hardware Design Finished



FTU2-DB Hardware Design Finished




DDR2
T-FPGA CONF. BITS
TEST VECTORS
PCI-EXPRESS SOCKETS
MEMORY INTERFACE
ROCKET IO
Data Bus
CONTROL UNIT
COMM
Clock/Debug Controller
EXTRA PORTS
RESULTS MEM
INJECTION MEM
FRAME MEM
SelectMAP FSM
C-FPGA

FTU2 in Numbers

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

Max Signals	Configuration Interface
581	SelectMap x8
573	SelectMap x16
557	SelectMap x32

- Maximum DUT design size

	V5FX70T	V5SX95T	V5VLX155T
Slices	11200	14720	24320
Flip-Flops	44800	58880	97280
Block Rams	5.20Mb	8.58Mb	7.45Mb
Bitstream (bits)	27025408	35716096	43042304

FTU2 in Numbers

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

Maximum workload:

Nº Inputs * CLK cycles = 2 * 8 * 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]

X = Max. Number of Inputs or Outputs	Max. Frequency (MHz)
X<128	250
128<X<256	125
256<X<384	83.3
384<X<512	62.5
512<X<581	50

[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

$$T_{RUN} \leq T_{bit\_flip} + T_{Workload}$$

  - Time necessary to inject a fault.
    - By hardware => 2000 CLK cycles.
    - F=100MHz → T_{bit_flip}=20 µs

Workload	CLK (MHz)	T_{RUN}	Nº RUNS/s	$T_{bitflip}$(%)
500000	125	4020µs	248	0.5
250000	125	2020µs	495	1
125000	125	1020µs	980	2
50000	125	420µs	2380	4,7
125000	250	520µs	1923	3.84

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

3.75Gb/s

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

6.5Gb/s

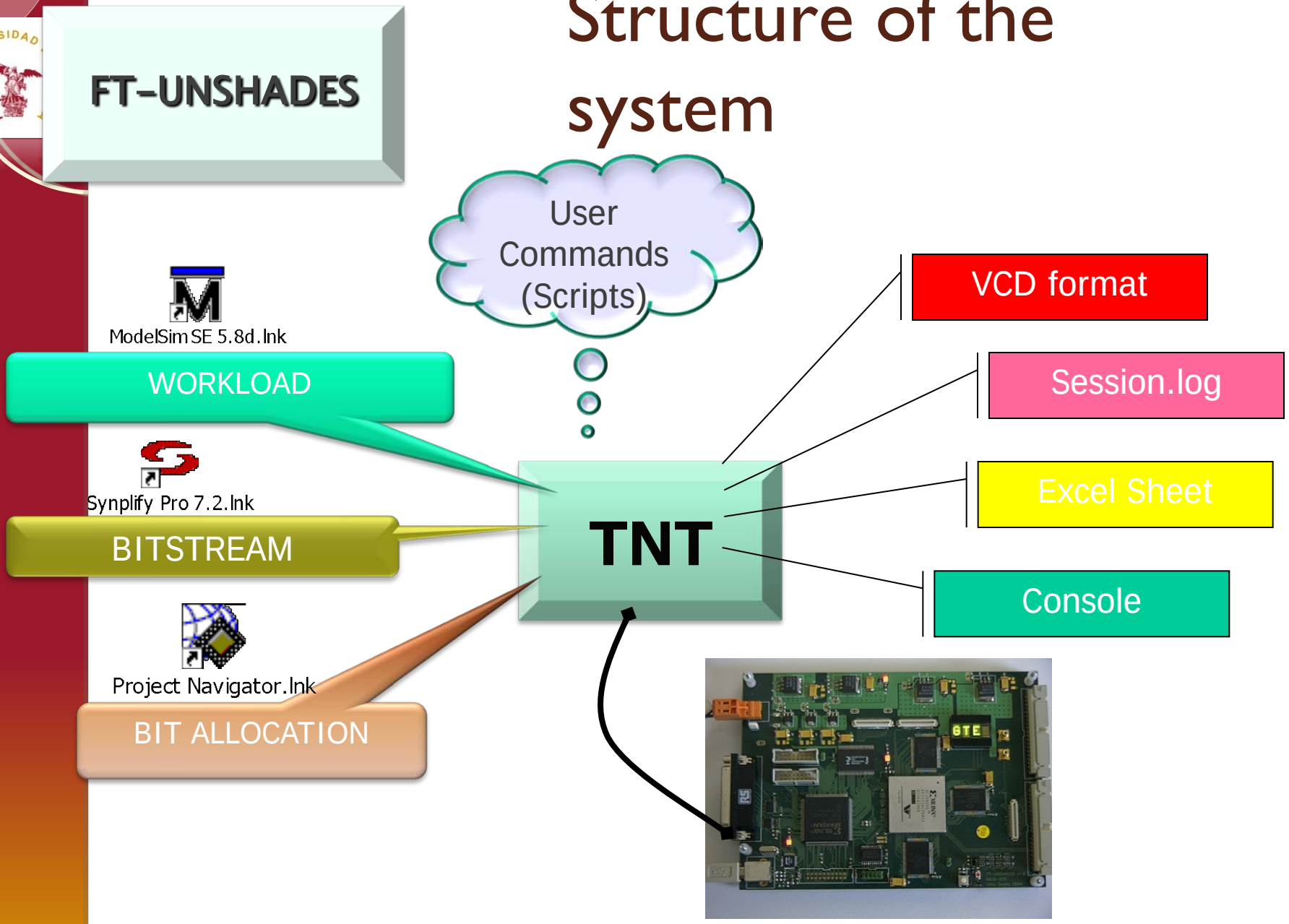
Data transfer via PCI-Express

- Experimental Results

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

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).

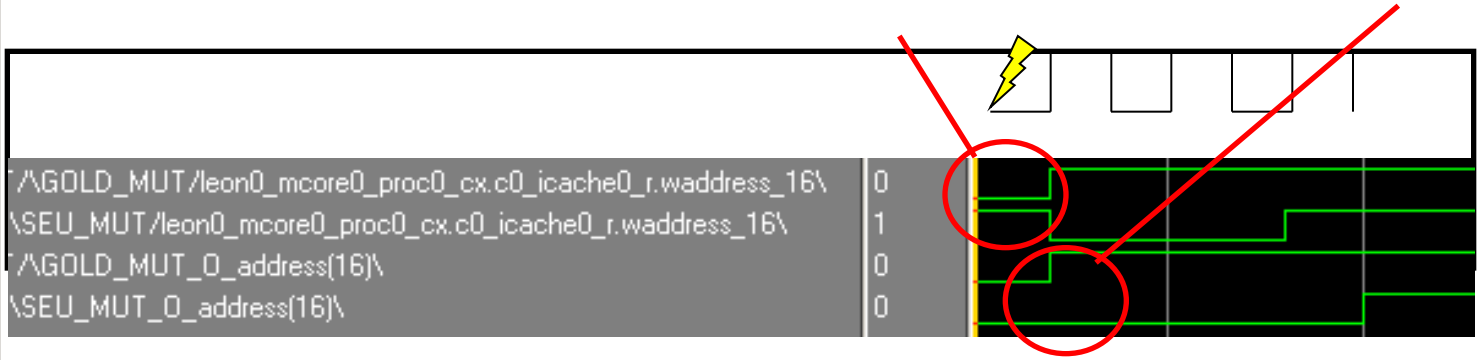
Structure of the system



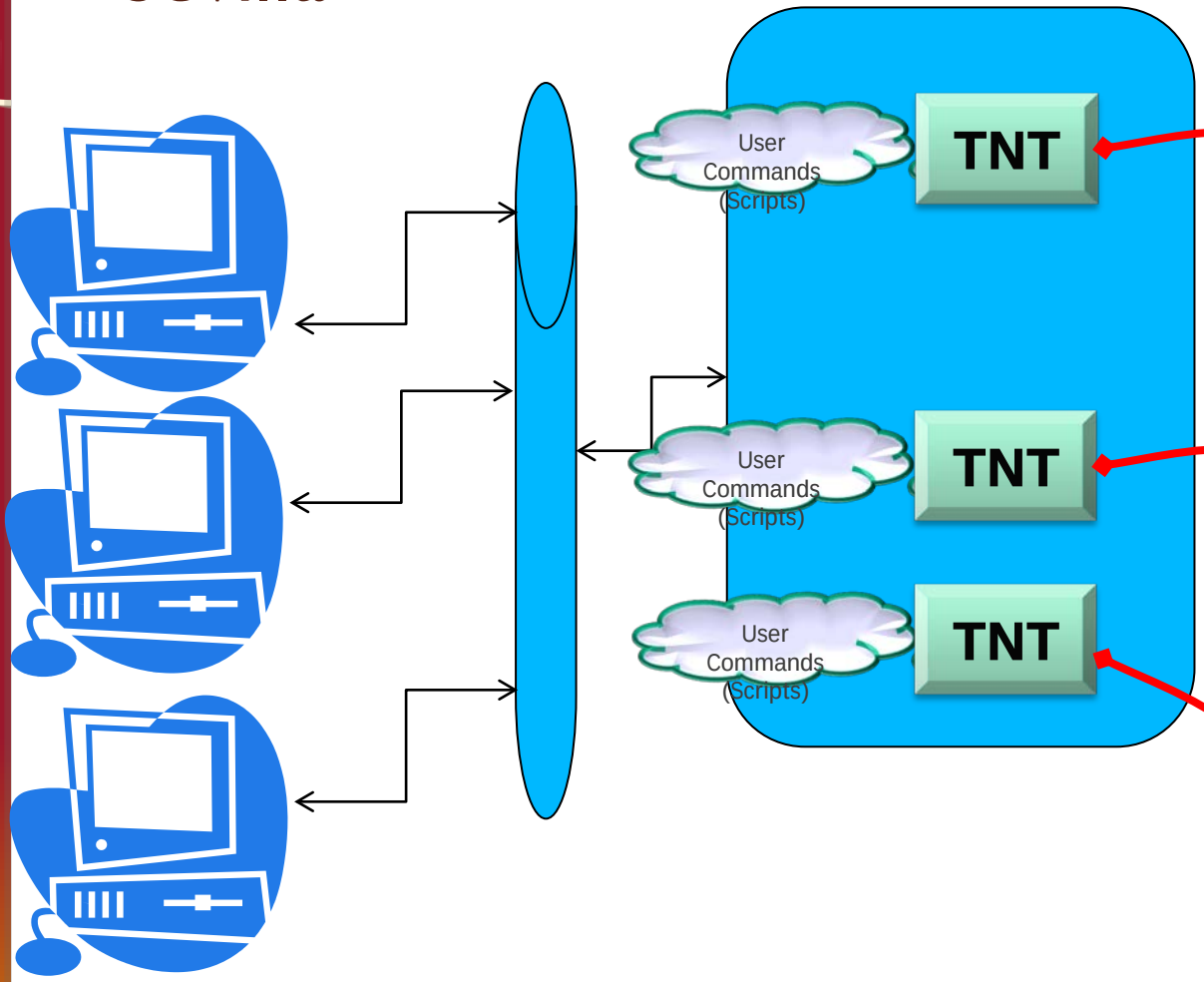
FT-UNSHADES Analysis Example

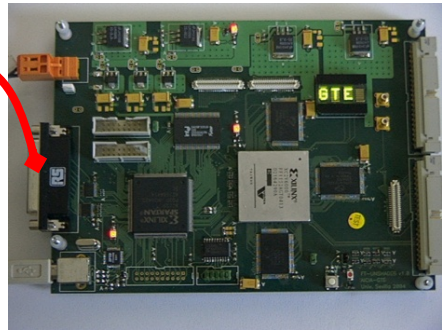
```
#RUN 1
Selected clk cycle for SEU insertion: 133937
Selected reg for SEU insertion: SEU_MUT/leon0_mcore0_proc0_cx.c0_icache0_
ess_16
OK
Output error detected in port: address
Damage detected 1 clk cycle after SEU insertion
Total elapsed time: 0.062501
Target size: 1, registers
Total FPGA Cycles: 0x0,00020B32 (133938)
```

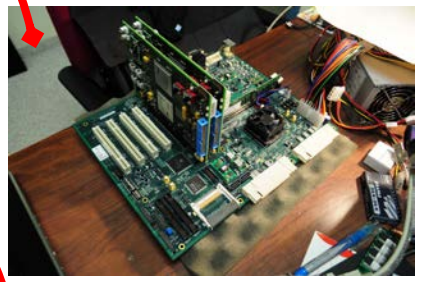
CLOCK: 133937
REGISTER:leon0_mcore0_proc0_cx.c0_icache0_r.waddress_16
DAMAGE DETECTED:YES
LATENCY: 1 CLK
PORT: address



FT-UNSHADES user center at University of Sevilla







User friendly interface

uff-TNT | Welcome, usuario | **Select device** | more

name	description	connected	available
XC2V 3000	Device description here	?	i134649152
XC2V 6000	Device description here	?	i134649152
XC2V 8000	Device description here	?	i134649152
XC5V LX50	Device description here	?	i134649152

uff-TNT | Welcome, usuario | XC2V 8000 | **Select design** | more

name	status	timestamp	size
awachifu	No vector file available.	[design timmestamp here]	[design size]
cordic	Ready to run.	[design timmestamp here]	[design size]
dte.8.leonhash	No vector file available.	[design timmestamp here]	[design size]
test.8.blockmem	No vector file available.	[design timmestamp here]	[design size]
test.8.distribmem	Ready to run.	[design timmestamp here]	[design size]

design name here | design description here | Create

uff-TNT | Welcome, usuario | XC2V 8000 | test.8.distribmem | Files | **Analysis** | more

Manage files

Device map: Load | Bit file: Load | Vectors: Load

Run

inject on SEU_MUT* | Go!

Repeat

1 | select campaign | full run | Go!

Step

1 | no VCD dump | Go!

Diff

Diff

Set

seusperrun	1	level	damage
regsim	onepass	timesim	onepass
seed	1	maxruns	1
stop@dam	no	timewins	1-74
freqsim	50000000	automask	0
masktoseu	yes	maxtime	0
clear@timeout	no	undo@end	no
clear@dam	no	restoremode	none
restorebus	none	debug	no
quiet	0		Update

Buses

Create bus: name | pattern | Create

```
[000 01 43.209348] Creating system buses...
[000 01 43.209654]  \_ Cycles: 32 registers, 32 addresses
[000 01 43.209761]  \_ FaultyRegs: 39 registers, 39 addresses
[000 01 43.209930]  \_ DbgEvents: 3 registers, 3 addresses
[000 01 43.210096]  \_ MonEvents: 1 registers, 1 addresses
[000 01 43.210264]  \_ EventMask: 3 registers, 3 addresses
[000 01 43.210370]  \_ GoldRegs: 39 registers, 39 addresses
[000 01 43.210535]  \_ MonMask: 1 registers, 1 addresses
[000 01 43.210560]  \_ FaultRegsOut: 1 registers, 1 addresses
[000 01 43.210578]  \_ GoldRegsOut: 1 registers, 1 addresses
[000 01 43.210767]  \_ hwm2s: 1 registers, 1 addresses
[000 01 43.211012]  \_ hwCycles: 32 registers, 32 addresses
[000 01 43.211027] Loading port names...
loadbit
loadvect
[000 01 46.559182] Loading test vectors
[000 01 46.559233] Number of test vectors: 74
runtest
[000 01 46.580143] AUTOMASK set to 0
[000 01 46.591114] Onepass mode: Maxruns auto-set to 2886
[000 01 46.591136] Running...
[000 04 31.702246] Total elapsed time: 165.121855
[000 04 31.702279] Target size: 39 registers
[000 04 31.702294] Total FPGA Cycles: 0x0,000259B8 (154040)
[000 04 31.702311] Total damage detected: 1984, cycles: mean 14.7 min 0 max 41
[000 04 31.702328]  \_ dout: 1984
```

>

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 trough 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

RADECS SEVILLA SPAIN SEPTEMBER 19/23 2011

11th EUROPEAN CONFERENCE on RADIATION and EFFECTS on COMPONENTS and SYSTEMS
SEVILLA, SPAIN - SEPTEMBER 19th-23th-2011

NSREC 2011