A Fault Injection Methodology and Infrastructure for Fast Single Event Upsets Emulation on Xilinx SRAM-based FPGAs

S. DI CARLO, P. PRINETTO, D. ROLFO, P. TROTTA
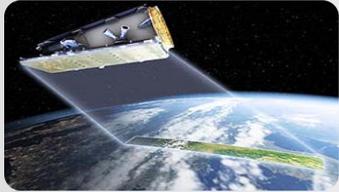
OUTLINE

- MOTIVATIONS

- STATE OF THE ART

- OPTIMIZED FAULT INJECTION ARCHITECTURE

- EXPERIMENTAL RESULTS

- CONCLUSIONS

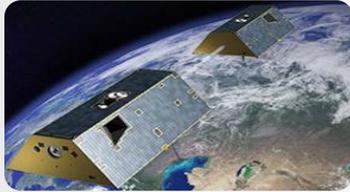# MOTIVATIONS

NovaSAR   Grace Follow-On   Formosat-5 (2015)   Glonass-K (2015)   Iridium Next (2015)
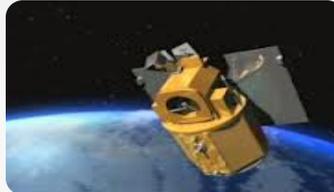
XILINX Space & Defense Devices

VIRTEX™

VIRTEX-II

VIRTEX V4

VIRTEX V5

ZYNQ 7000
ZYNQ 350 7000
ZYNQ 7045
ZYNQ 7000

SRAM FPGAs

Relative Performance

2000   2004   2008   2011
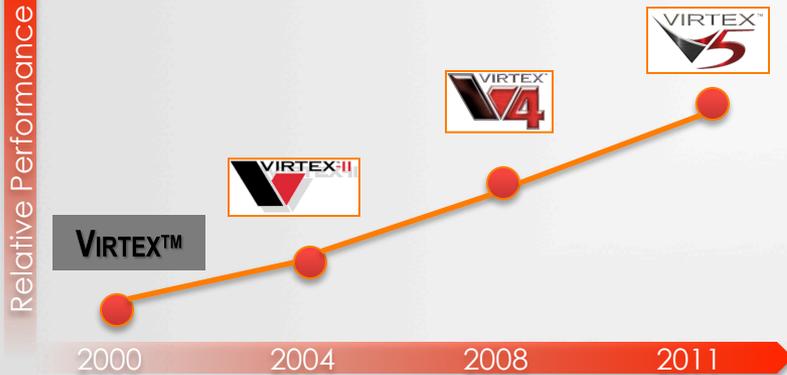
Increased use in **SAFETY CRITICAL** applications

Highly susceptible to **Single Event Upsets** (SEUs)

MOTIVATIONS

SEU for **Uosat-3** spacecraft

FPGA Design

FPGA Conf. Memory
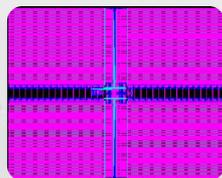
Faults in **FPGA Design**

- Errors in the content of the FPGA design memory elements (e.g., flip-flops, DRAMs, or RAMs)
- Classic Fault Injection techniques used to evaluate the reliability of the design
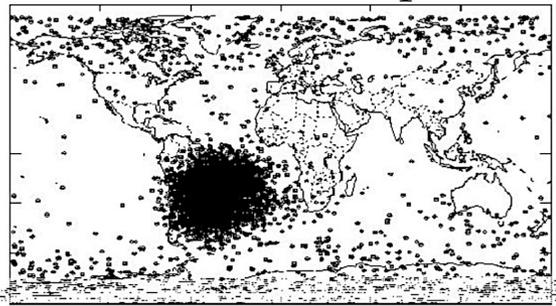
Faults in **FPGA Conf. Memory**

- A bit-flip in the configuration memory may permanently alter the functionality of the implemented circuit
- The configuration memory is very large therefore the probability of error is high
- Several faults in the configuration memory do not influence the design

MOTIVATIONS

Efficient **Fault Injection** required to:

1 Analyze the **effect** of SEUs in the **configuration memory**

2 Identify **weaknesses** of the circuit

3 Selectively and efficiently apply **fault detection** and/or **fault tolerance** techniques to harden the design.

STATE OF THE ART (DPR BASED FAULT INJECTION)

FPGA
- ICAP
- Fault Injection Infrastructure (FII)
- Fault List

Reconf. Module
- Circuit Under Test (CUT)

Read an FPGA Frame → Inject the fault in the frame → Reconfigure the frame with the faulty configuration → Execute the Workload → Restore the correct configuration

References
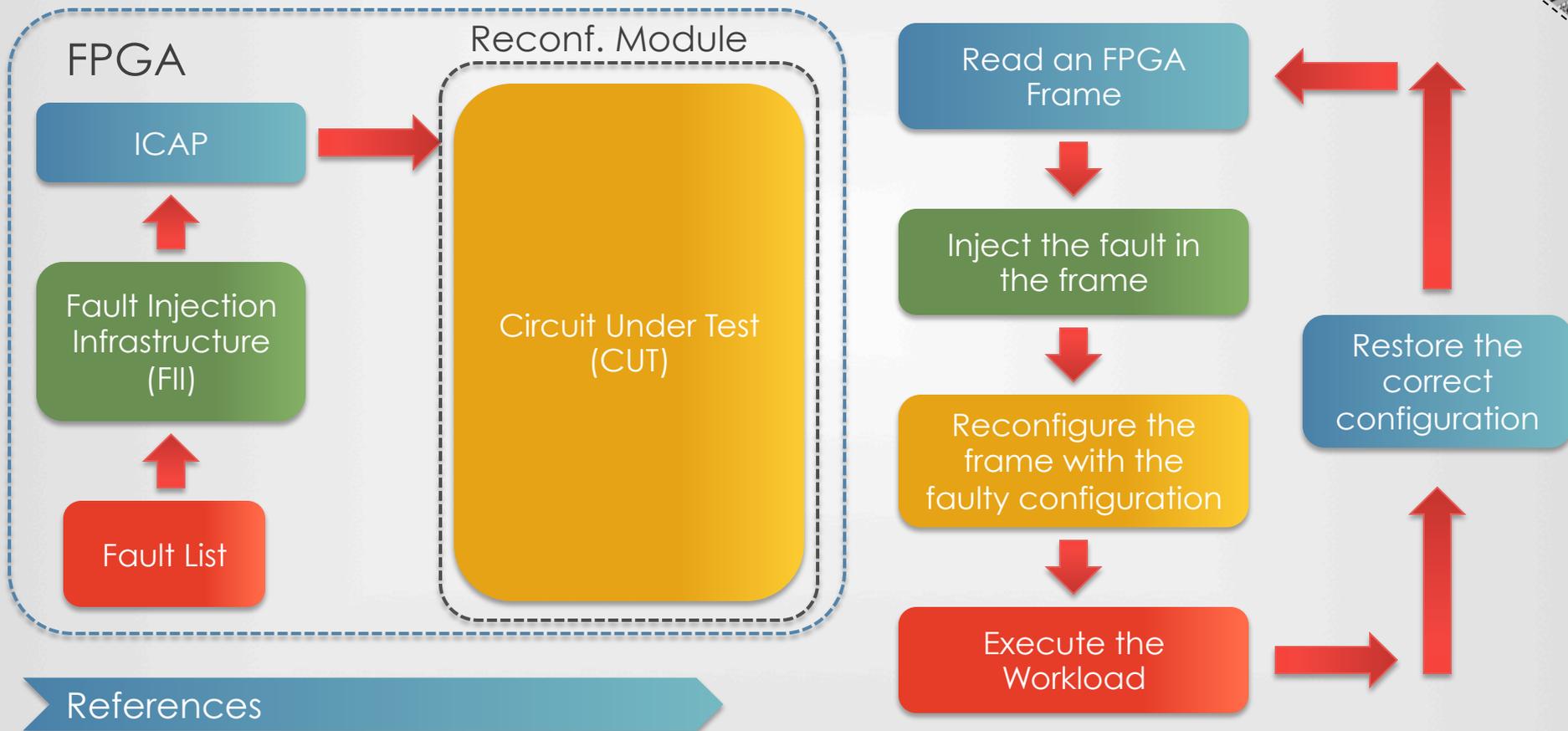
[Ibrahim et al. @ RAST'13], [Legat et al. @ DDECS'10],
[Nazar et al. @ DFTS'12], [Mogollone et al. @ RADECS'11],
[Sterpone et al. @ DFTS'07]

STATE OF THE ART (DPR BASED FAULT INJECTION)

Drawbacks and limitations

- Knowledge of the FPGA internal architecture (address of different frames)
- A flip in a frame can affect multiple frames (multiple injected faults, no restore is possible)
- Potential faults injected in the fault infrastructure (stall of the injection process, erroneous fault classification)
- High rate of no-effect faults

Proposed solutions

- Extremely speed up the fault injection process, injecting faults only on the sensitive memory configuration bits
  - The main novelty is a careful selection of the location in which faults must be injected using Xilinx Essential Bits technology
- Ensure the correct behavior of the fault injection infrastructure

OPTIMIZED FAULT INJECTION

Fault Injection optimization implemented in three steps:

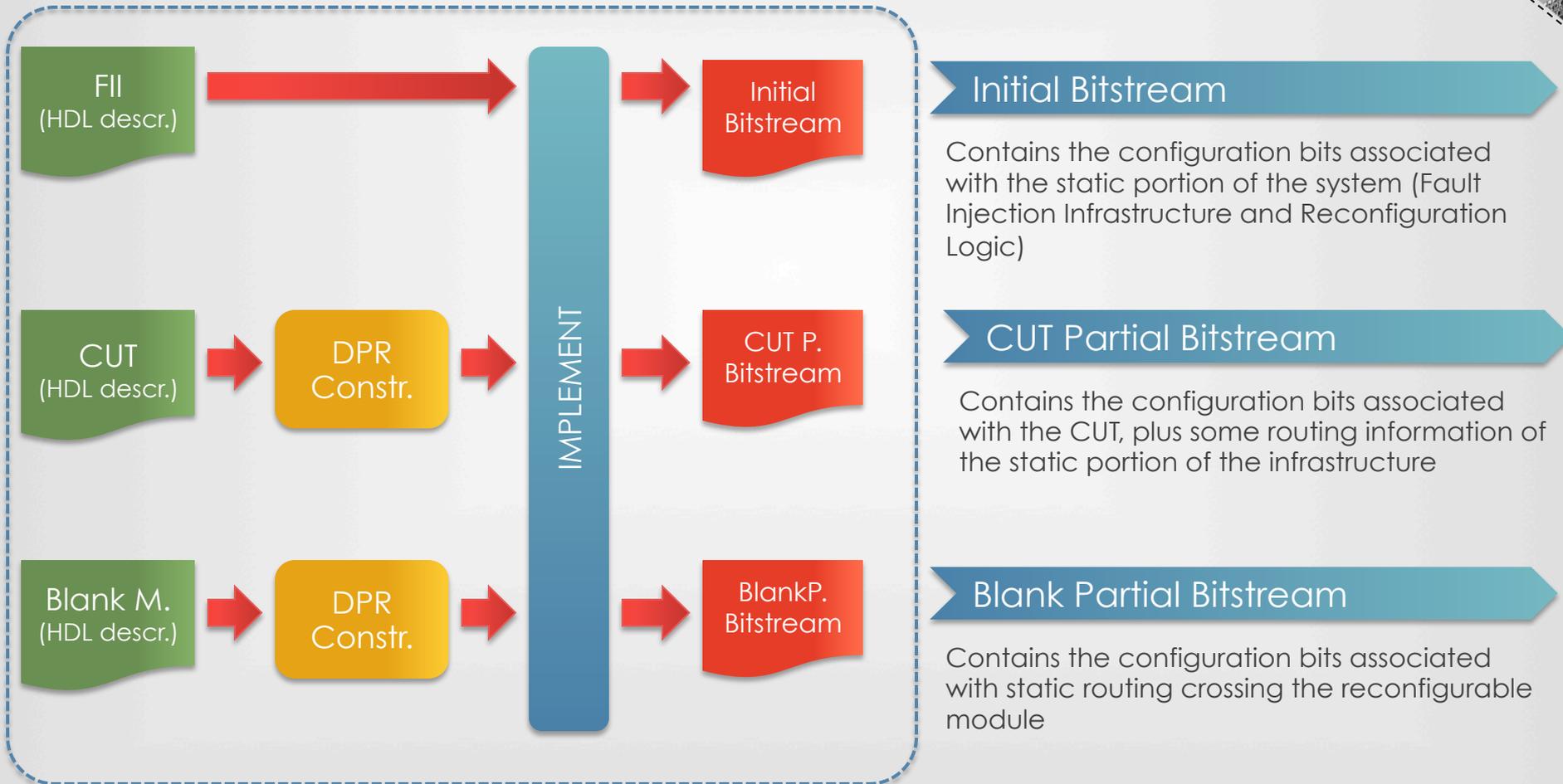**1**   **CUT Partial Bitstreams** generation

2   Identification of the **Essential Fault Locations (EFL)**
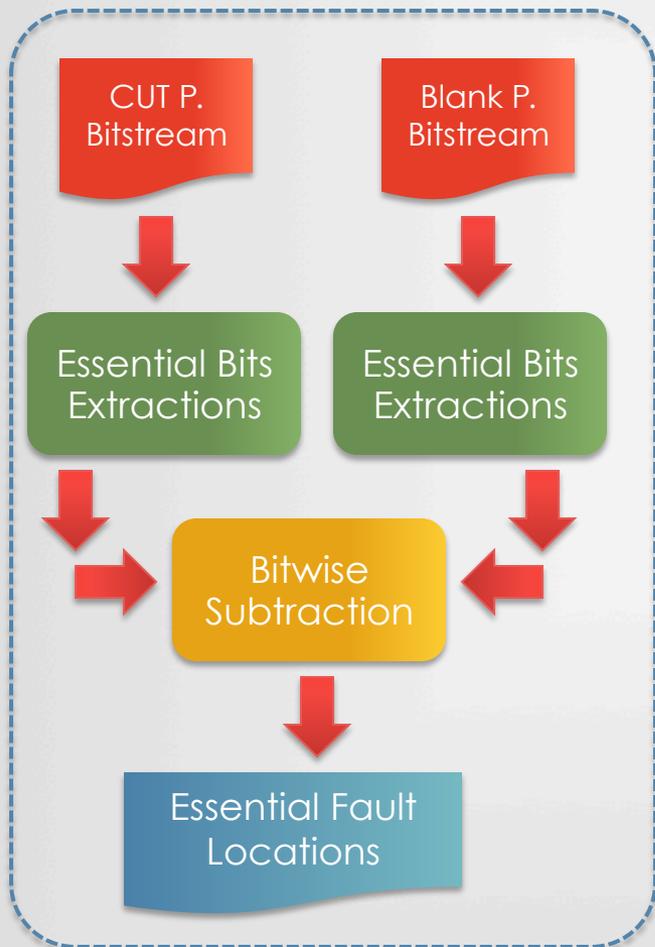
3   **Fault injection** and **fault effect characterization**

OPTIMIZED FAULT INJECTION (IDENTIFICATION OF ESSENTIAL FAULT LOC.)

FII (HDL descr.) → IMPLEMENT → Initial Bitstream

CUT (HDL descr.) → DPR Constr. → IMPLEMENT → CUT P. Bitstream

Blank M. (HDL descr.) → DPR Constr. → IMPLEMENT → BlankP. Bitstream

Initial Bitstream

Contains the configuration bits associated with the static portion of the system (Fault Injection Infrastructure and Reconfiguration Logic)

CUT Partial Bitstream

Contains the configuration bits associated with the CUT, plus some routing information of the static portion of the infrastructure

Blank Partial Bitstream

Contains the configuration bits associated with static routing crossing the reconfigurable module

OPTIMIZED FAULT INJECTION (IDENTIFICATION OF ESSENTIAL FAULT LOC.)

CUT P. Bitstream

Blank P. Bitstream

Essential Bits Extractions

Essential Bits Extractions

Bitwise Subtraction

Essential Fault Locations

Essential bits extractions

- Essential bits are the configurations bits that are essential for the functionality of the system
- These bits can be identified using the Xilinx BitGen tool

Bitwise subtraction

- Removes the essential bits associated with the static routing

Essential fault locations

- Contains the locations of the identified essential bits
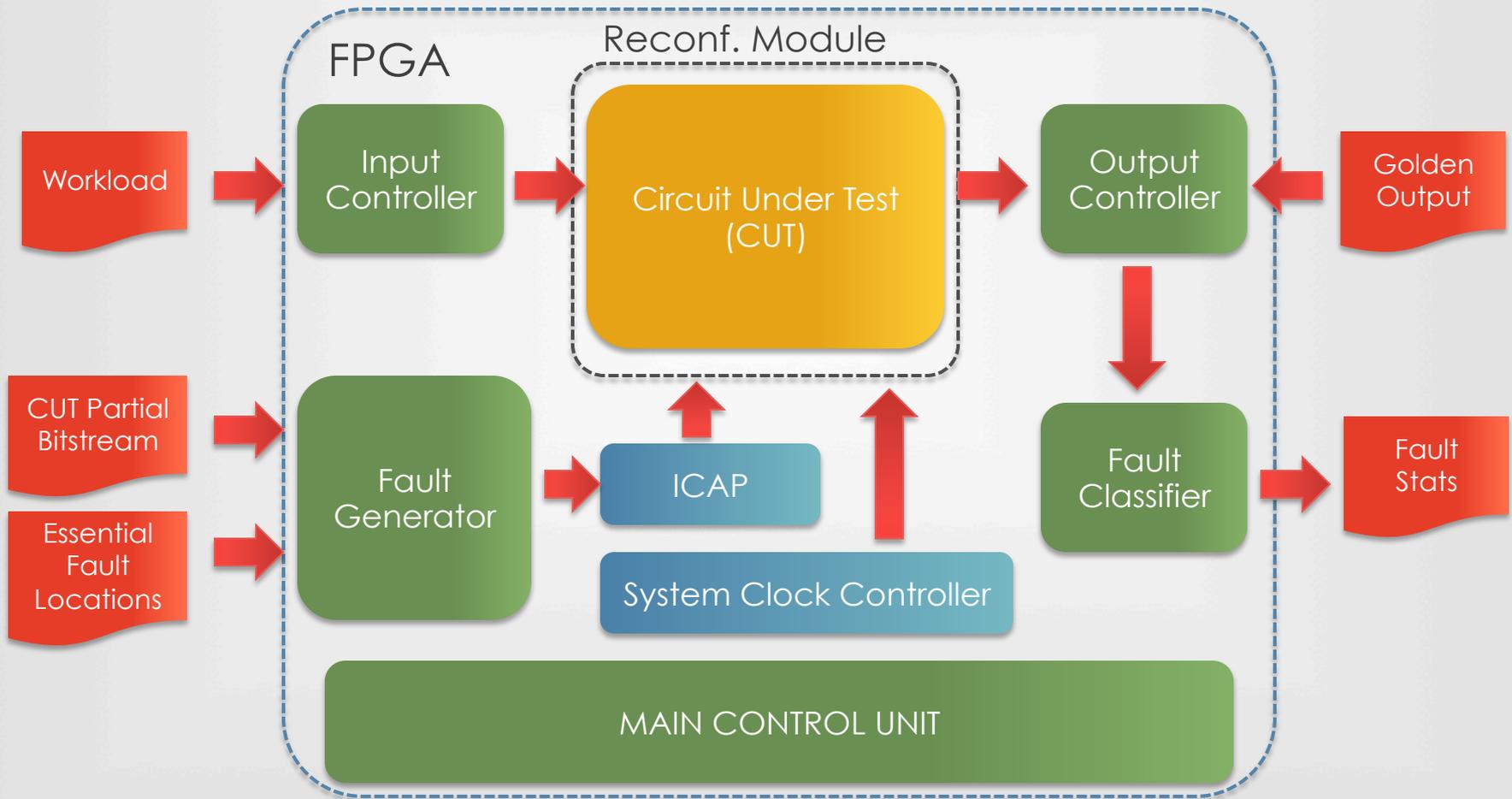
OPTIMIZED FAULT INJECTION (FAULT INJECTION)

Faults are injected in the **Essential Fault Locations** using FPGA DPR

- Essential bits are approximately 20% of the total bits of the memory configuration

- The fault injection process does not include the configuration associated with the fault injection infrastructure
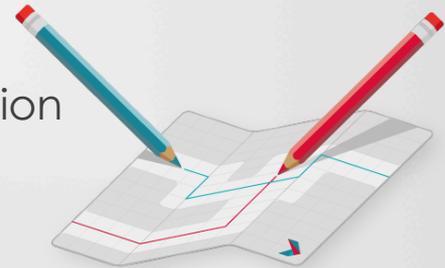
OPTIMIZED FAULT INJECTION (ARCHITECTURE)

EXPERIMENTAL RESULTS

XILINX ML605

Three case studies

1 **LEON3-based SoC** running several applications from the MiBench benchmark suite

2 2D convolution core for space image processing [Di Carlo et. Al IDT'11]
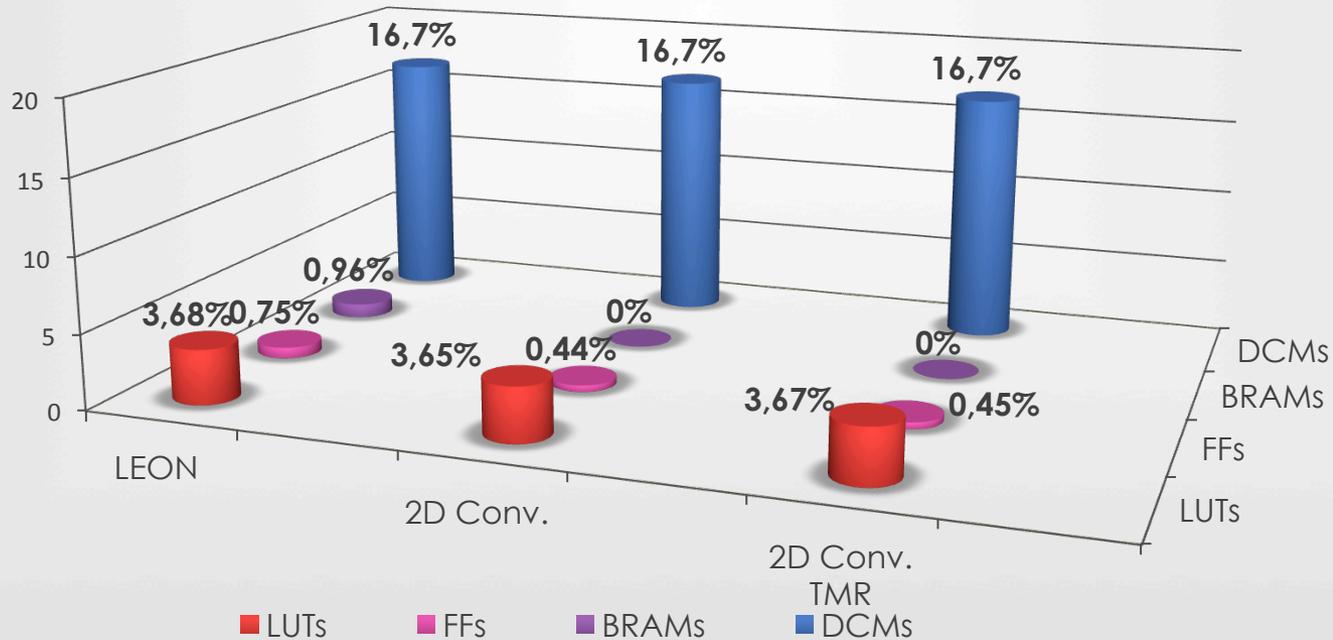
3 2D convolution core with TMR

EXPERIMENTAL RESULTS

EXPERIMENTAL RESULTS

CUTs Bitstream size (BS), percentage of Essential Bits (EB) , workload execution time (T_{run}), and total injection time (T_{inj})

CUT	BS[KB]	%EB	T_{run}[ms]	T_{inj}[h]
L3 Susan	755.6	16.2	37.91	11.5
L3 CRC32	755.6	16.2	20.94	6.82
L3 IFFT	755.6	16.2	395.65	109.8
2D Conv.	170.9	13.6	6.14	1,27
2D Conv. TMR	478.4	13.6	6.14	3,55

EXPERIMENTAL RESULTS

XILINX ML605

CUT	EFL
L3 Susan	6.18M
L3 CRC32	6.18M
L3 IFFT	6.18M
2D Conv.	1.39M
2D Conv. TMR	3.9M

Fault Injection classification results

Chart — Fault Injection classification results:

- L3 Susan: 90,2% %Non-Critical, 9,8% %Critical
- L3 CRC32: 92,7% %Non-Critical, 7,3% %Critical
- L3 IFFT: 86,4% %Non-Critical, 13,6% %Critical
- 2D Conv.: 87,6% %Non-Critical, 12,4% %Critical
- 2D Conv. TMR: 98,7% %Non-Critical, 1,3% %Critical

■ %Non-Critical   ■ %Critical
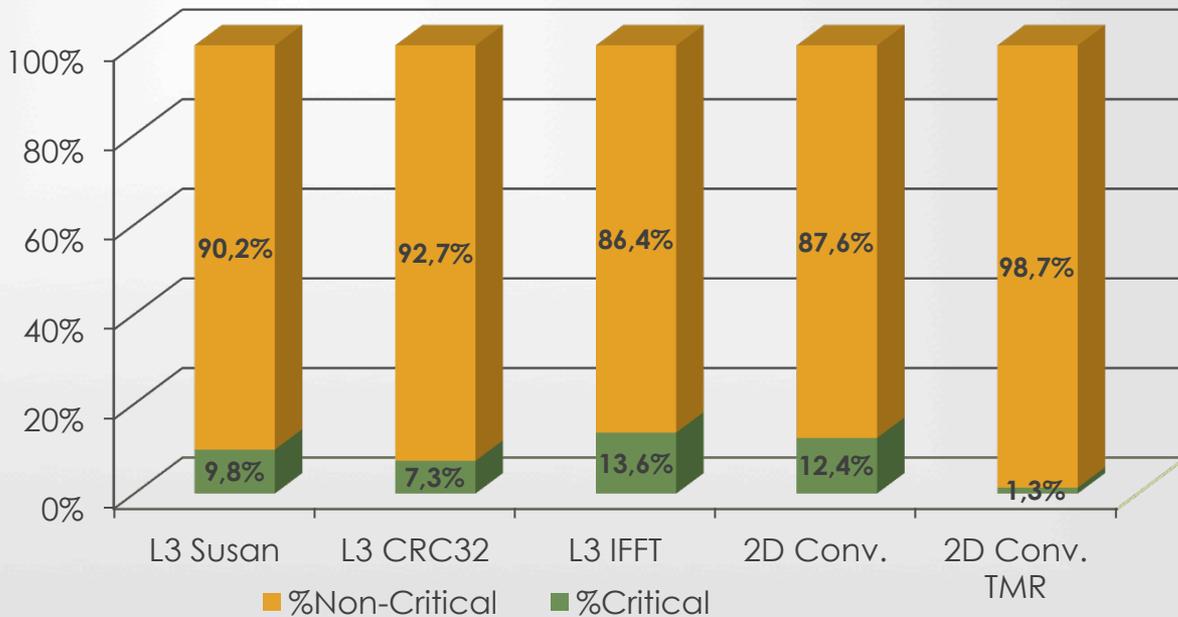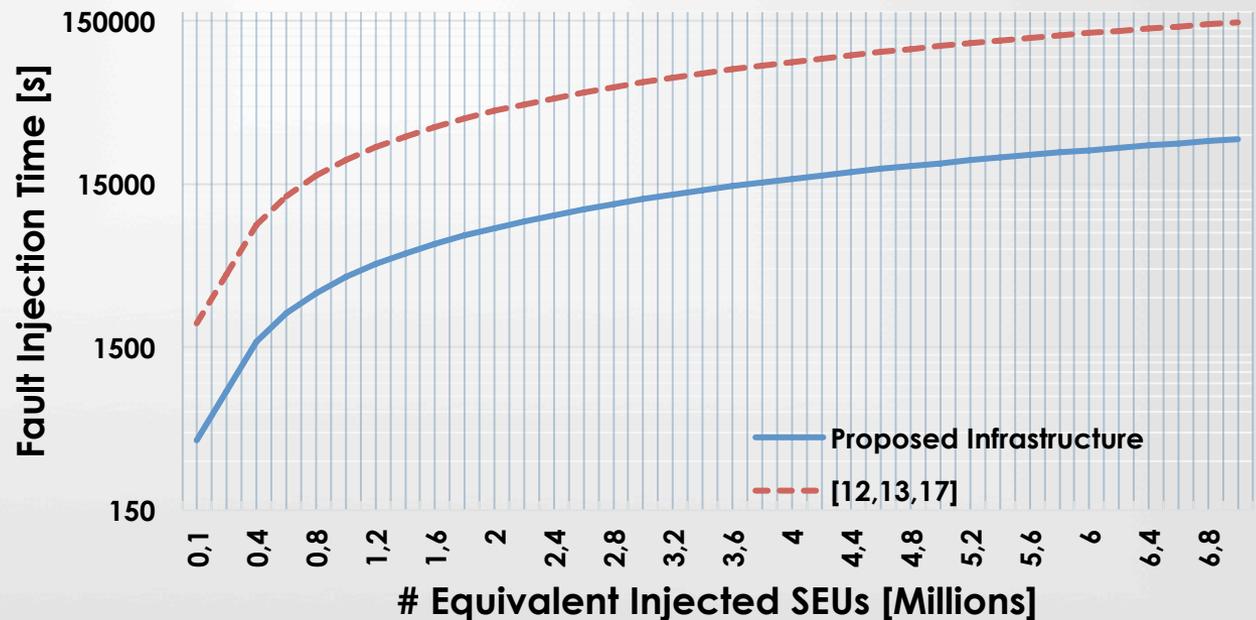
Performance improvement

- Testcase: LEON 3 running the CRC32 workload
- Comparison with [Sterpone et al. @ DFTS'07] [Nazar et al. @ DFTS'12] (implementation obtained from the paper)

CONCLUSIONS

DPR-based fault injection methodology and infrastructure for SEUs emulation in the configuration memory of Xilinx SRAM-based FPGAs

- Exploits the Xilinx Essential Bits technology to speed-up fault injection, ensuring the correctness of the infrastructure operations during the whole injection process
- Fault injection time speed-up of almost 10x for very high number of injected faults
  - Ability to evaluate reliability of FPGA designs of different complexity.

the proposed fault injection architecture can easily support Multiple Bits Upset fault model

- Future works will focus on the investigation of fault accumulation effects on FPGA-based designs

CONCLUSIONS