LA-UR-15-22303
Approved for public release; distribution is unlimited.

Title: The Use of Benchmarks for High-Reliability Systems

Author(s): Quinn, Heather Marie; Robinson, William; Rech, Paolo; Barnard, Arno; Aguirre, Miguel; Desogus, Marco; Entrena, Luis; Garcia-Valderas, Mario; Guertin, Steve Michael; Kaeli, David; Kastensmidt, Fernanda Lima; Kiddie, Bradley; Sanchez-Clemente, Antonio; Reorda, Matteo Sonza; Sterpone, Luca; Wirthlin, Michael

Intended for: SELSE, 2015-03-31 (Austin, Texas, United States)

Issued: 2015-03-31

VANDERBILT UNIVERSITY

The Use of Benchmarks for High-Reliability Systems

H. Quinn, **W. H. Robinson**, P. Rech, A. Barnard,
M. Aguirre, M. Desogus, L. Entrena, M. Garcia-Valderas,
S. M. Guertin, D. Kaeli, F. L. Kastensmidt,
B. T. Kiddie, A. Sanchez-Clemente, M. Sonza Reorda,
L. Sterpone, M. Wirthlin

Los Alamos NATIONAL LABORATORY EST. 1943

UFRGS UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

UNIVERSITEIT STELLENBOSCH UNIVERSITY

UNIVERSIDAD DE SEVILLA

POLITECNICO DI TORINO 1859 1906

Universidad Carlos III de Madrid

JPL Jet Propulsion Laboratory California Institute of Technology

NORTHEASTERN UNIVERSITY BOSTON, MASSACHUSETTS

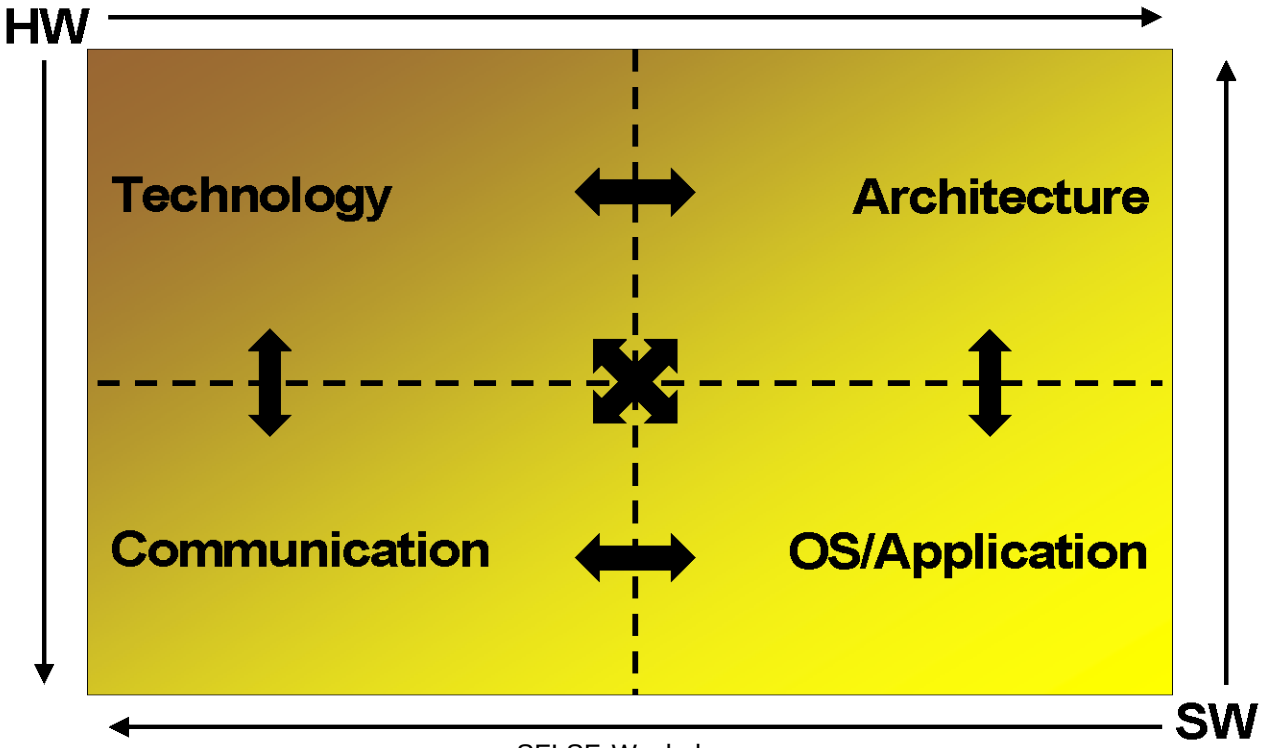
BRIGHAM YOUNG UNIVERSITY FOUNDED BYU 1875 PROVO, UTAH

Presentation Outline

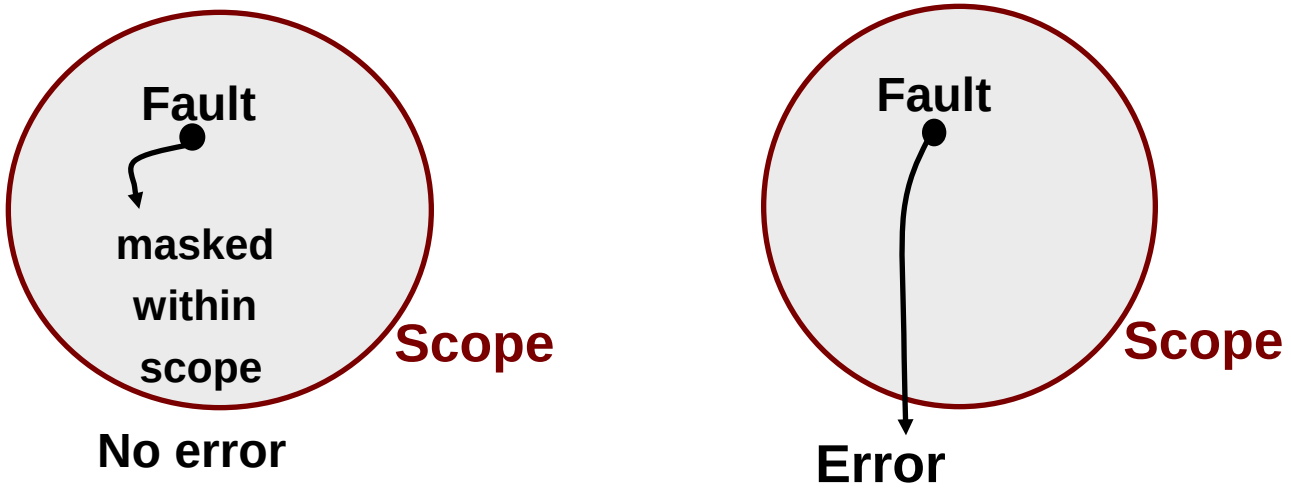
- Using benchmarks for radiation testing
- Field-programmable gate array (FPGA) benchmark
- Microprocessor software benchmark
- Preliminary test results for the benchmark
- Summary and future work

Complexity of System Design Space

- Viewing the system in terms of "**hardware**" and "**software**" is a **coarse-grained** analysis
- Understanding the **linkages** among the quadrants is critical for development of a **reliable, trusted** system



Fault vs. Error



- **Fault:**
  - Faults are underlying problems or defects in the hardware
  - Transient faults are ones that appear and disappear
- **Error:**
  - An error is a manifestation of a fault at a particular scope (e.g., chip boundary)
  - A soft error is caused by a transient fault
- **Failure:**
  - Error in outermost scope

Classification of Faults [1]

Faults experienced by semiconductor devices fall into three main categories:

- **Permanent**
  - Reflect irreversible physical changes
  - Example: Oxide wearout that causes a transistor malfunction
- **Intermittent**
  - Occurs repeatedly at the same location
  - Tend to occur in bursts when the fault is activated
  - Can be removed with replacement of the offending circuit
  - Example: Partial oxide wearout
- **Transient**
  - Occur because of temporary environmental conditions
  - Example: Radiation-induced bit flips

[1] C. Constantinescu, "Trends and challenges in VLSI circuit reliability," IEEE Micro, vol. 23, pp. 14-19, 2003.

Using Benchmarks for Testing

- **Benchmarks have been widely embraced by many researchers.**
- **Benchmarks allow designers to determine relative improvements caused by:**
  - Processing technology
  - Architecture
  - Circuit design
  - Software
- **A number of communities use benchmarks:**
  - Design for Test (DFT)
  - Automated Test Pattern Generation (ATPG)
  - Compiler
  - Supercomputing

Other Benchmarks in Use

- **Hardware benchmarks**
  - ISCAS 85/89
  - ITC'99, and
  - IWLS 2005
- **Software benchmarks**
  - Dhrystone/Whetstone
  - Linpack
  - Coremark
  - SPECint/fp
  - RODINIA
  - NAS

Benchmarks in High-Reliability Systems

- **Currently, there is no benchmark suite for reliability or radiation testing for either FPGAs or microprocessors**
- **The current state of the art is to use:**
  - Homemade, synthetic designs that represent worst-case scenarios,
  - Circuits from OpenCores, existing benchmarks, or Xilinx's CoreGen Tool, or
  - Designs that have been used previously by the researchers.
- **Codes/circuits used are not guaranteed to be consistent among research groups**

Advantages of Using Benchmarks for High-Reliability Systems

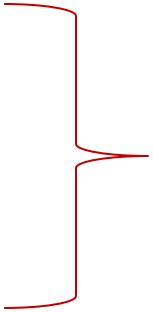
- **Assess relative reliability improvements between mitigated and unmitigated designs.**
- **Compare mitigation methods for effectiveness, performance, area, and power.**
- **Compare algorithms across architectures and process changes.**
- **Assess architectural effects on reliability, such as using or not using caches.**
- **Assess the effect of coding methods on reliability, such as iterative solvers.**
- **Compare test methodologies across organizations, including both radiation testing and fault injection.**

FPGA Benchmark

- **Using ITC'99 I99T for now**
  - Reasonably sized set of circuits which are small enough for mitigation
  - Comes with existing input vectors
- **For testing we have been focusing on B13**
  - Trying to correlate beam testing with fault injection data, so that we can test the entire suite using fault injection

Microprocessor Software Benchmark

- AES-128 with NIST test vectors
- Cache test with four memory test patterns
- CoreMark with internal inputs
- Matrix multiply
- FFT
- Hotspot
- Quicksort



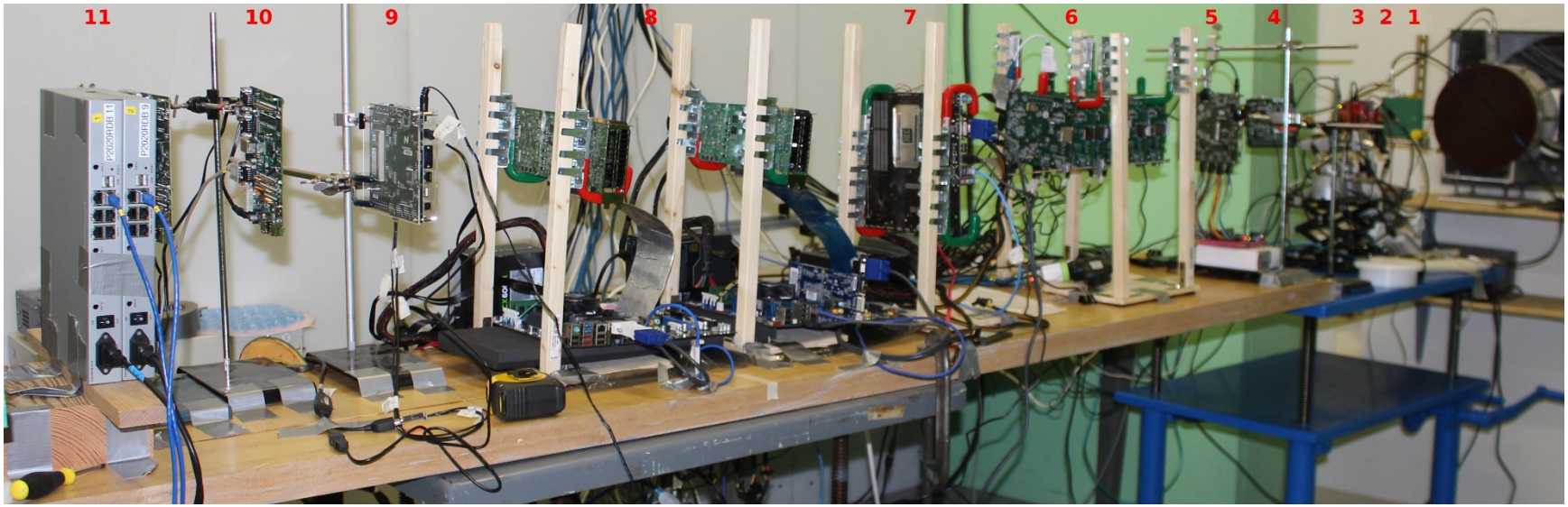
Randomly generated inputs

Neutron Beam Testing

- **Initial radiation tests were completed at the Los Alamos Neutron Science Center (LANSCE) in December 2014.**
- **Both benchmarks were tested on a number of different architectures.**
- **Dual purpose for the beam testing**
  - Individual mitigation approaches were evaluated against unmitigated implementations
  - Collective assessment of the benchmark composition

VANDERBILT UNIVERSITY

Test Setup from LANSCE



Experiments at LANSCE Test

	Benchmark	Org	Component
1	s/w[1]	LANL	2 x TI MSP430F2619
2	s/w[1]	LANL	2 x TI MSP430FR5739
3	s/w[1]	LANL	2 x TI Tiva
4	h/w[1]	Madrid	Xilinx Artix-7
5	h/w[1]	Torino	Xilinx Virtex-5
6	s/w[1]	UFRGS	6 x Xilinx Zynq
7	s/w[1]	UFRGS	Kaveri A10 Apu
8	s/w[1]	UFRGS	Tesla K20 GPUs, Xeon Phi
9	h/w[1]	UFRGS	Xilinx Virtex-5
11	s/w[1]	JPL	Freescale P2020
12	s/w[2] + h/w[1]	BYU	Xilinx Kintex-7
13	s/w[1]	BYU	Xilinx Zynq
14	custom[2]	Vanderbilt	

[1]Mitigated

[2]Unmitigated

Microcontroller Results

Code	Tiva	MSP430F2619	MSP430FR5739
AES	0.30 (0, 1.1)	0.38 (0.04, 1.37)	0.85 (0, 3.1)
AES TMR	0.31 (0, 1.1)	3 (1, 5)	2 (0, 7)
Cache	75 ± 10	8 ± 2	10 (6, 15)
Cache TMR	0.27 (0, 1.0)	0.21 (0, 0.76)	2 (0, 8)
Coremark	0.75 (0.15, 2.20)	1.27 (0.51, 2.61)	N/A
M x M	59 ± 13	4 (2, 6)	1 (0, 4)
M x M TMR	10 (7, 14)	0.27 (0, 1.0)	2 (0, 8)
Qsort	59 ± 13	3 (2, 5)	25 (16, 38)
Qsort TMR	0.34 (0, 1.27)	7 (4, 10)	2 (0, 7)

Software was mitigated using Trikaya software technique for s/w mitigation

- **All of these components are very small, which is why the FIT rate is small.**
- **These results show that AES-128 is naturally resistant to errors: very small amount of memory and processing.**
- **Many similarities in results are due to forcing similar amount of memory.**
- **These values are not normalized to the amount of work performed:**
  - Cache test makes the MSP430F2619 look like the most robust operation.
  - In reality, it is doing far less processing than the Tiva.
  - The slower processing in Coremark shows how the slower processing decreases resilience to errors.

NVIDIA K20 Results

Code	Config	SDC FIT	O'head
M x M	Unhard	(4:63 ± 0:80) x 10^2	1.0
	ECC	44:91 ± 9:94	1.01
	ABFT	8.34 ± 0.96	1.14
FFT	Unhard	(2.88 ± 0:39) x 10^3	1.0
	ECC	(4.14 ± 0.88) x 10^2	1.5
	ABFT	8.34 ± 0.96	1.18
Htspt	Unhard	(2.04 ± 0:31) x 10^3	1.0
	ECC	18.16 ± 2.01	1.0
	Spatial DWC	3.26 ± 0.45	2.45
	Temporal DWC	2.45 ± 0.34	1.90

- **Increase in overhead for ECC is modest, but sensitivity to SEFIs increases.**
  - ECC fails on multiple-bit upsets (MBUs)
- **For MxM ECC on GPUs has similar reliability improvement to LANL's Trikaya, but with less overhead.**
- **ABFT seems extremely efficient compared to other techniques.**

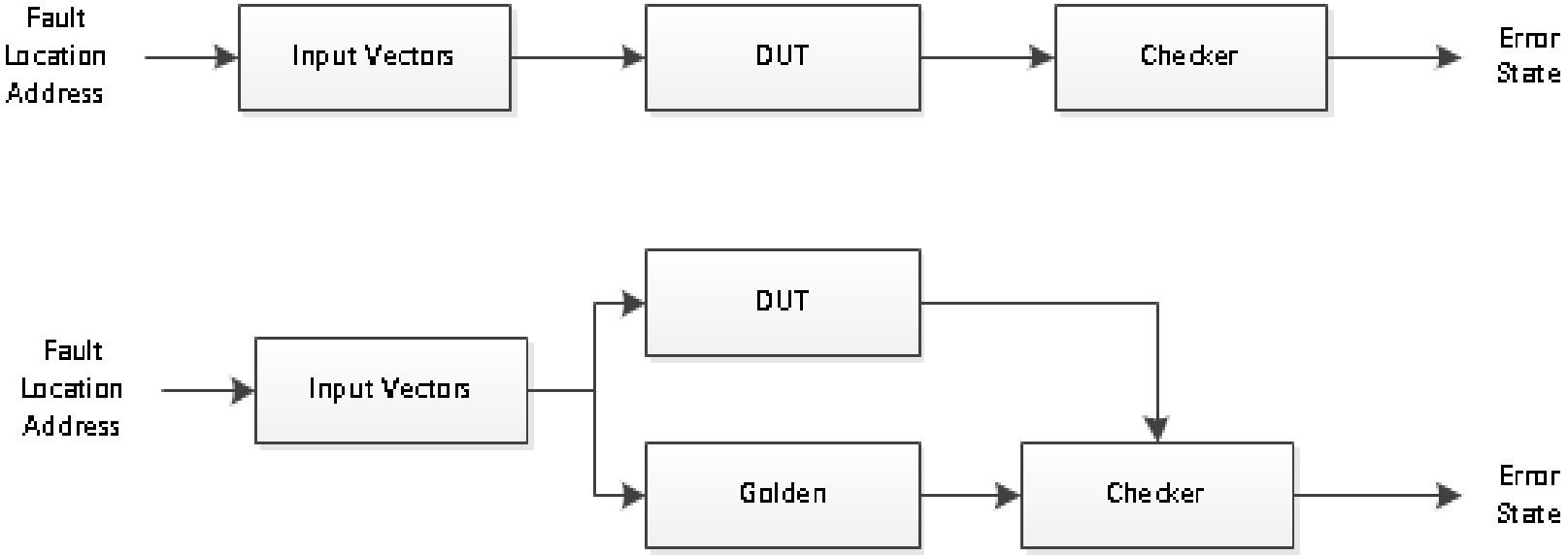
Virtex-5 LX50T Results for B13

Config	SDC FIT	O'head
Unhard	(2.10 ± 0:03) x 10^3	1.0
XTMR	(1.72 ± 0:01) x 10^3	4.56
VERI-Place	(1.34 ± 0:04) x 10^2	4.56

- **B13 has been implemented using the Xilinx TMR tool and the POLITO's software mitigation tool named VERI-Place.**
  - No scrubbing during test
- **The X-TMR and VERI-Place implementation take the same amount of FPGA resources.**
  - VERI-Place software hardens the circuit's physical netlist by acting on the logic placement position.
- **The results report the SDCs normalized to the unhardened version ones, while the overhead represents the increase of the circuit area.**
- **Focusing on validating the B13 results using fault emulation.**

Using Fault Injection [1]

- **Can perform fault injection to trigger malicious hardware within the design under test (DUT)**
  - May (or may not) be triggered by the test vector suite
- **FPGAs could accelerate this time-consuming process**



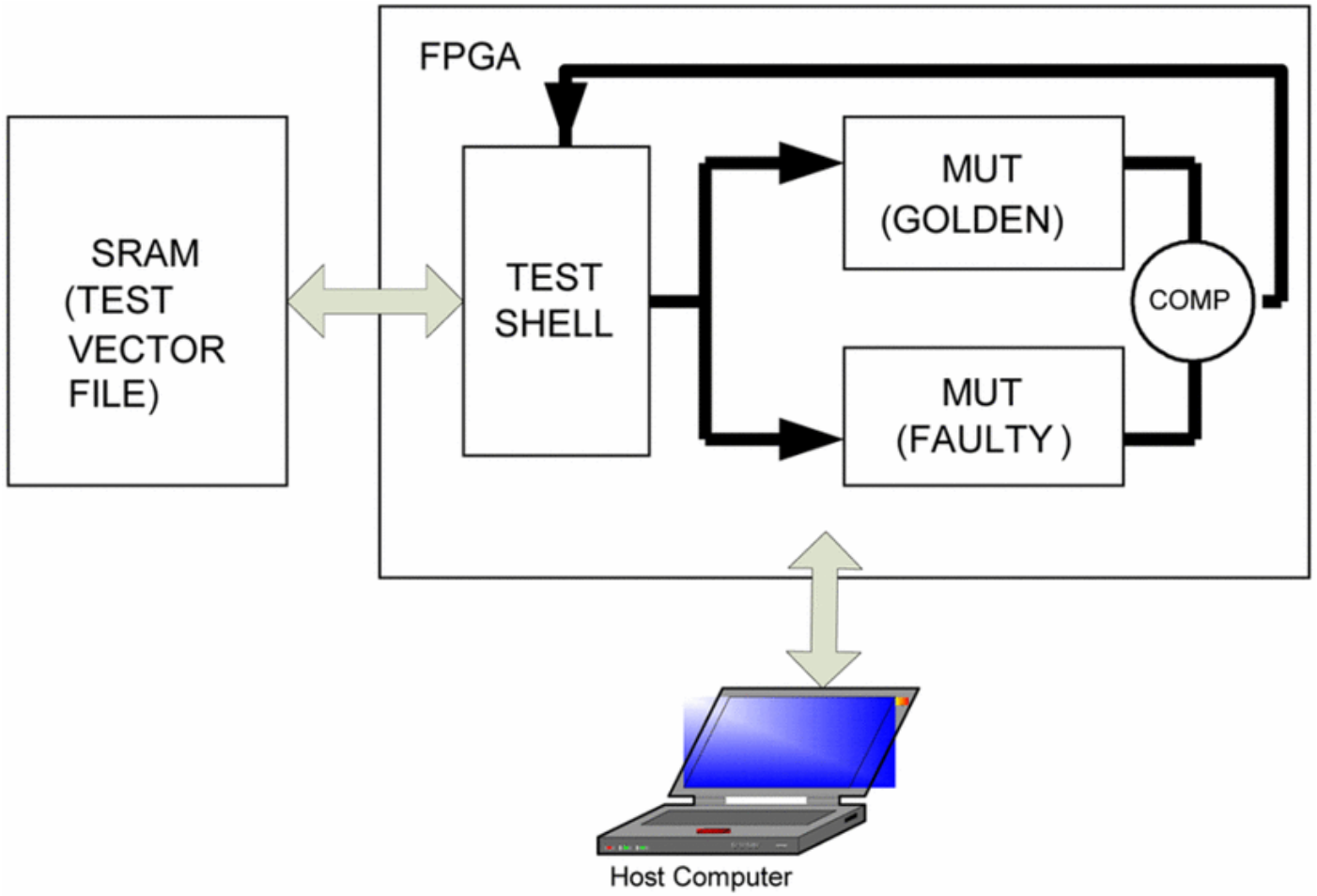
[1] H. M. Quinn, D. A. Black, W. H. Robinson, and S. P. Buchner, "Fault simulation and emulation tools to augment radiation-hardness assurance testing," IEEE Transactions on Nuclear Science, vol. 60, pp. 2119-2142, 2013.

Fault Simulation and Emulation Tools [1]

- **FPGAs as hardware accelerators**
  - Provide additional computational capability
- **SLAAC1-V SEU Emulator**
  - Los Alamos National Lab and Brigham Young University
  - One of the first fault emulation systems for FPGAs
- **FT-UNSHADES and FT-UNSHADES2**
  - University of Sevilla
  - Allows for emulation of radiation-hardened by design ASICs on an FPGA
  - System is available for on-line research purposes (search for “FT-UNSHADES2” for contact info)

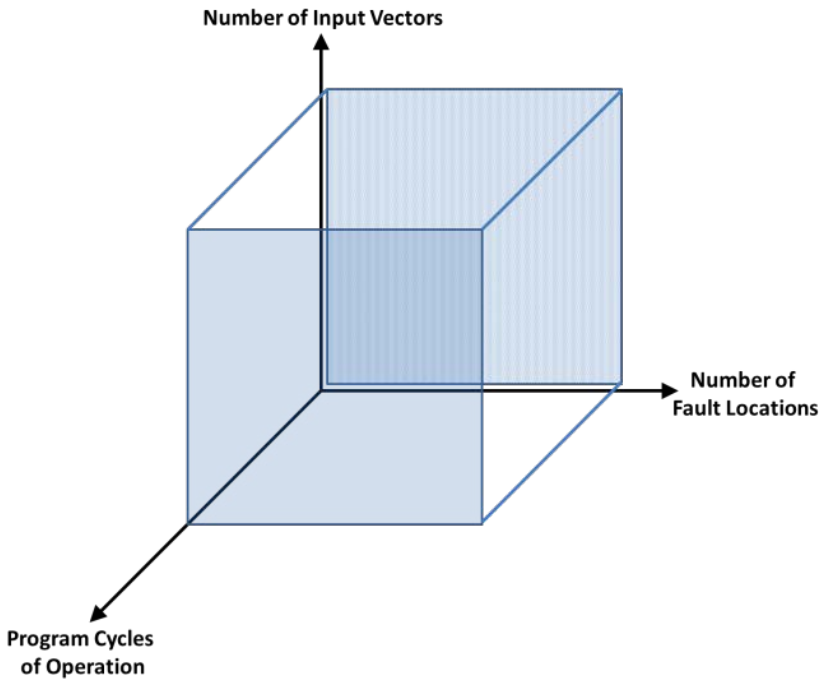
[1] H. M. Quinn, D. A. Black, W. H. Robinson, and S. P. Buchner, "Fault simulation and emulation tools to augment radiation-hardness assurance testing," IEEE Transactions on Nuclear Science, vol. 60, pp. 2119-2142, 2013.

FT-UNSHADES Fault Emulation Tool



Limitations of Fault Injection [1]

- **Trade-off between accuracy and testing time**
- **Effective sampling of the test space to detect (i.e., trigger) the malicious hardware**
- **In the case of FT-UNSHADES, the experimental setup is different**



[1] H. M. Quinn, D. A. Black, W. H. Robinson, and S. P. Buchner, "Fault simulation and emulation tools to augment radiation-hardness assurance testing," IEEE Transactions on Nuclear Science, vol. 60, pp. 2119-2142, 2013.

FT-UNSHADES Results

Design and conditions	Device	Essential Bits	Total Essential Bits
B13_X30_plain (Polito generation)	LX50T	322,399	11,006,638
B13_X30_plain (Polito circuit)	FX70T	327,850	18,936,096
B13_X30_plain (FTU2 circuit)	**FX70T**	**333,525**	**18,936,096**
B13_X30_XTMR (Polito generation)	LX50T	1,824,638	11,006,368
B13_X30_XTMR (Polito circuit)	FX70T	1,912,920	18,936,096
B13_X30_XTMR (FTU2's circuit)	**FX70T**	**1,922,272**	**18,936,096**
B13_X30_XTMR_veriplace_H	LX50T	1,809,635	11,006,368
B13_X30_XTMR_veriplace_L	LX50T	1,809,635	11,006,368

Summary And Future Work

- **We have attempted to create a common benchmark for comparing results across architectures, process technology, and mitigation schemes.**
  - Hardware: ITC'99 I99T
  - Software: AES, Cache, Coremark, FFT, Hotspot, M x M, Qsort
- **We have completed preliminary analysis of test results taken in Dec 2014**
- **There is still work to do:**
  - Is the software benchmark the right one? Might look at different software and input vector sets
  - Complete fault injection of the hardware benchmark

VANDERBILT UNIVERSITY

The Use of Benchmarks for High-Reliability Systems

H. Quinn, **W. H. Robinson**, P. Rech, A. Barnard,
M. Aguirre, M. Desogus, L. Entrena, M. Garcia-Valderas,
S. M. Guertin, D. Kaeli, F. L. Kastensmidt,
B. T. Kiddie, A. Sanchez-Clemente, M. Sonza Reorda,
L. Sterpone, M. Wirthlin

Los Alamos NATIONAL LABORATORY EST. 1943

UFRGS UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

UNIVERSITEIT STELLENBOSCH UNIVERSITY

UNIVERSIDAD DE SEVILLA

POLITECNICO DI TORINO 1859 1906

Universidad Carlos III de Madrid

JPL Jet Propulsion Laboratory California Institute of Technology

NORTHEASTERN UNIVERSITY BOSTON, MASSACHUSETTS

BRIGHAM YOUNG UNIVERSITY FOUNDED BYU 1875 PROVO, UTAH