Scuola Nazionale "Rivelatori ed Elettronica per Fisica delle Alte Energie, Astrofisica, Applicazioni Spaziali e Fisica Medica"

Simulare "Soft-error" in "SRAM-based FPGA": la piattaforma FLIPPER

M. Alderighi/F. Casini
monica@iasf-milano.inaf.it,
fcasini@iasf-milano.inaf.it

Goal

- Introduction of FLIPPER as a tool for simulating soft error in SRAM-FPGAs
- Presentation of case studies on FLIPPER usage

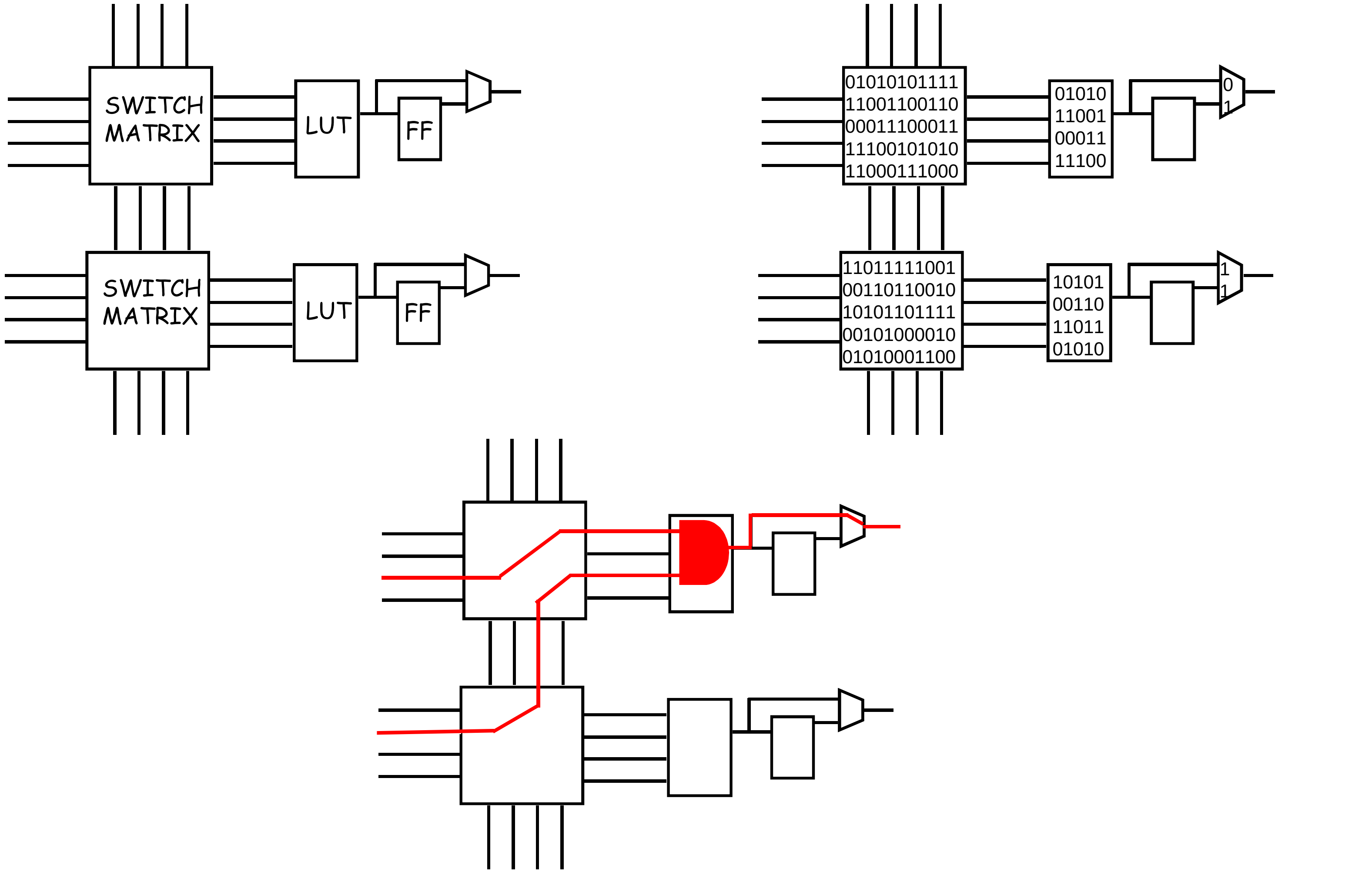
Outline

- Overview of FPGAs
- Radiation effects on FPGAs
- FLIPPER
- Examples

FPGA

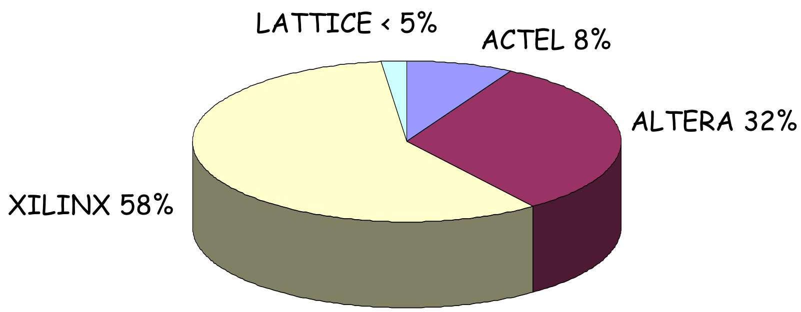
- Field Programmable Gate Array
  - Programmability
  - High integration density
  - High performance
  - Reduced development costs as compared to ASIC
- Applications
  - Telecom, Avionics, Space, Consumer Electronics, Automotive...
- Programmable logic elements and interconnections
- Hardware Description Language (VERILOG or VHDL) for configuration → CAD tool

Programmability/configurability



FPGA "Species"

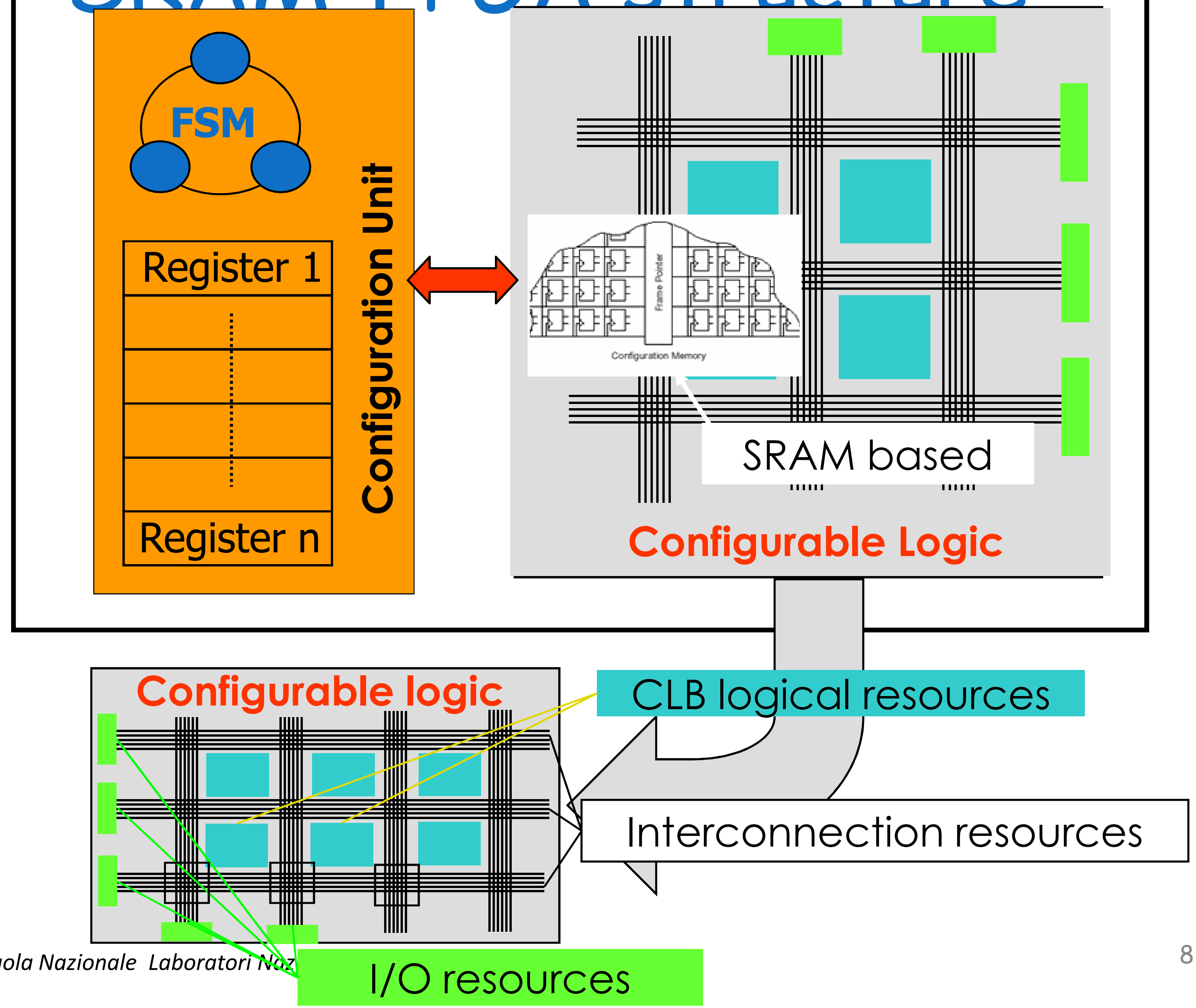
- Antifuse (ACTEL)
  - One time programmable
- FLASH (ACTEL)
  - Programmable
- SRAM (XILINX, ALTERA, LATTICE)
  - Programmable
  - Dynamic programmable



SRAM-FPGA

- Unlimited programmability!!!!
- High flexibility
  - A posteriori modifications of circuit functionalities
  - Fault reparation
- Dynamic programmable
  - Active partial reconfiguration (Xilinx)

SRAM-FPGA structure

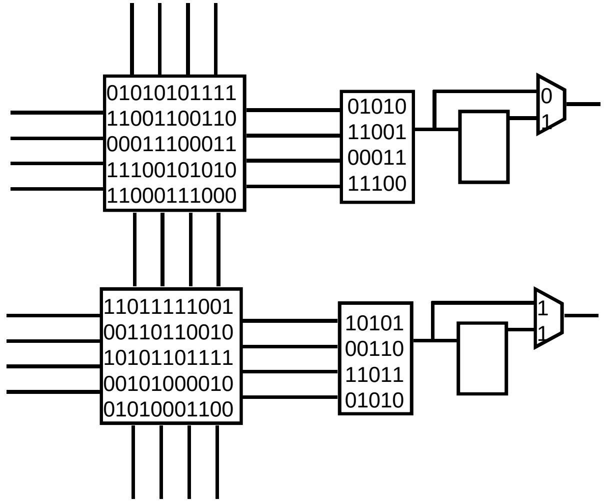


Interesting, but…

- Susceptibility to ionizing radiation (protons, heavy ions) and neutrons
- Effects
  - TID (Total Ionizing Dose)
  - SEE (Single Event Effect)
    - SEU (Single Event Upset)/MBU (Multiple Bit Upset)
    - SEL (Single Event Latch-up)
    - SET (Single Event Transient)
    - SEFI (<single Event Functional Interrupt)
- Mitigation
  - Manufacturing technology (TID)
  - Design Hardening
    - TMR tool, scrubbing

SEU in SRAM-FPGA

- Affect
  - Functions
  - Data
  - Interconnections
  - Configuration memory
  - Configuration logic



- Need suitable approach for space/avionic applications
  - Study and analysis of effects
  - Mitigation/protection techniques

Expected behavior & countermeasures

SRAM FPGAs

– SEUs in configuration memory and flip-flops/user memory, SEFI, SEL, and TID

SEU in configuration memory

- SEUs in configuration memory affect internal architecture and interconnections
- Mitigation is classically achieved by scrubbing the entire configuration memory content →the rate depends on the application and expected SEU rate
- Scrubbing is generally ruled by external circuit; for some devices it can be also performed by internal logic
- In some cases rewriting implies a device reset → it might provokes short service interruptions
- The time a device takes to reconfigure depends on the device size and allowed reconfiguration frequency

SEUs in flip flops/user memory

- The traditional approach for mitigating SEUs in flip flops is modular redundancy with voting scheme
- For SEUs affecting registers and user memory, error detecting and correcting codes can also be employed
- In case of data that are frequently rewritten, mitigation is easily obtained as new data overwrite old and possibly corrupted ones
- In case of data that do not vary often, scrubbing of registers and user memory can also be adopted for mitigating SEUs
- If dual port memories are employed, scrubbing can be performed in parallel with data access
- Modular redundancy with voting can be applied at resource, i.e. flips flops, as well as device level

SEFI

- An SEFI is a condition in which an SEU occurs in the device's control circuitry which prevents any further configurations

- As a countermeasure in case of a SEFI, a device reset is usually performed

- If that does not work, the device is power cycled

- Short service interruptions might occur

TID

- Countermeasures for TID are the choice of appropriate devices (technology), if possible, and adequate shielding

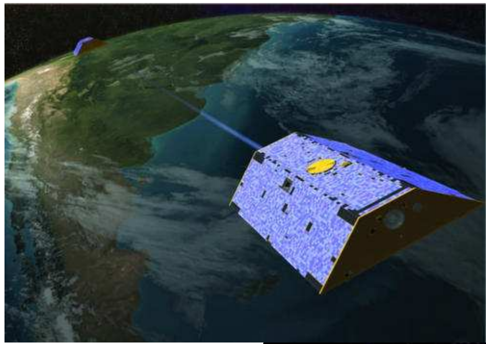
SEL

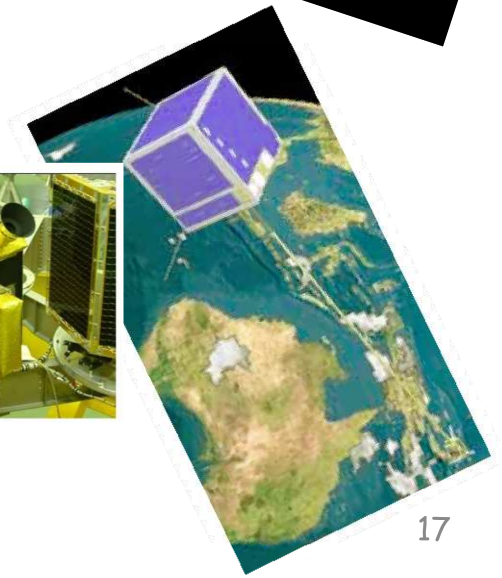
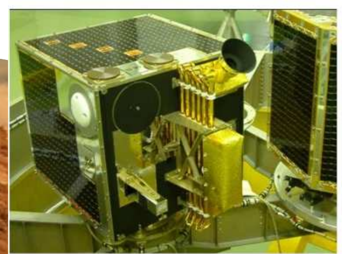
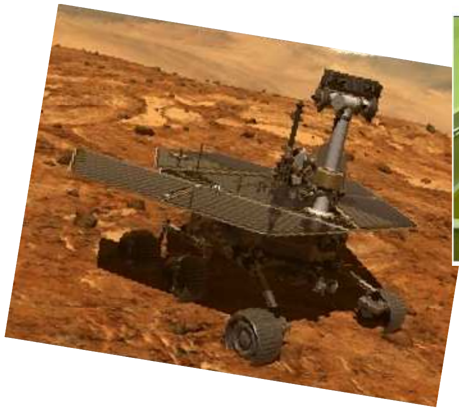
- Countermeasures for SELs are the same as for TID

- Ad hoc circuitry can also be developed which is able to detect progressively increase of current absorption and possibly switch off the device

SRAM-FPGA in space

- Venus Express
- Mars Reconnaissance Orbiter
- Mars Landers (pyro control)
- Mars Rovers (motor control)
- GRACE
- FedSat
- OPTUS Signal Processing
- TACSAT2
- CIBOLA



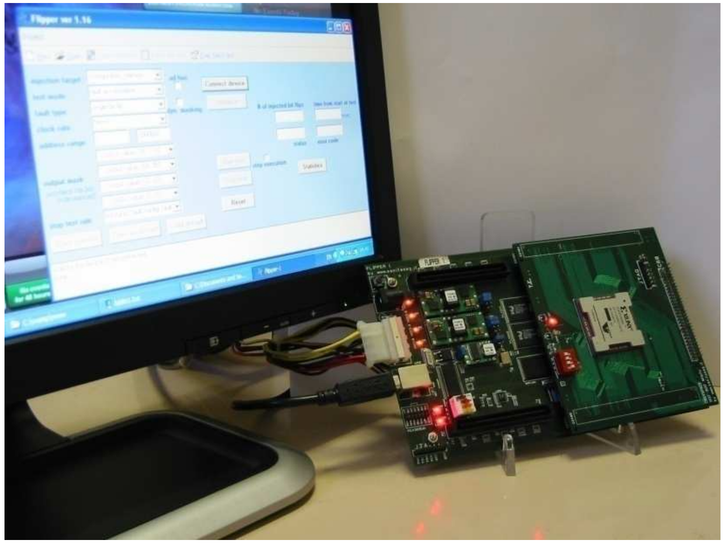




Evaluation of SEU effects

- Radiation ground testing
  - The higher the energy beams the better
  - Complex experimental set-up
  - Expensive!
- Simulation
  - Slow
- Fault emulation
  - Faster than simulation and nominal operation speed
- Static analysis
  - Independence from test vectors
  - STAR, RoRA by Politecnico di Torino

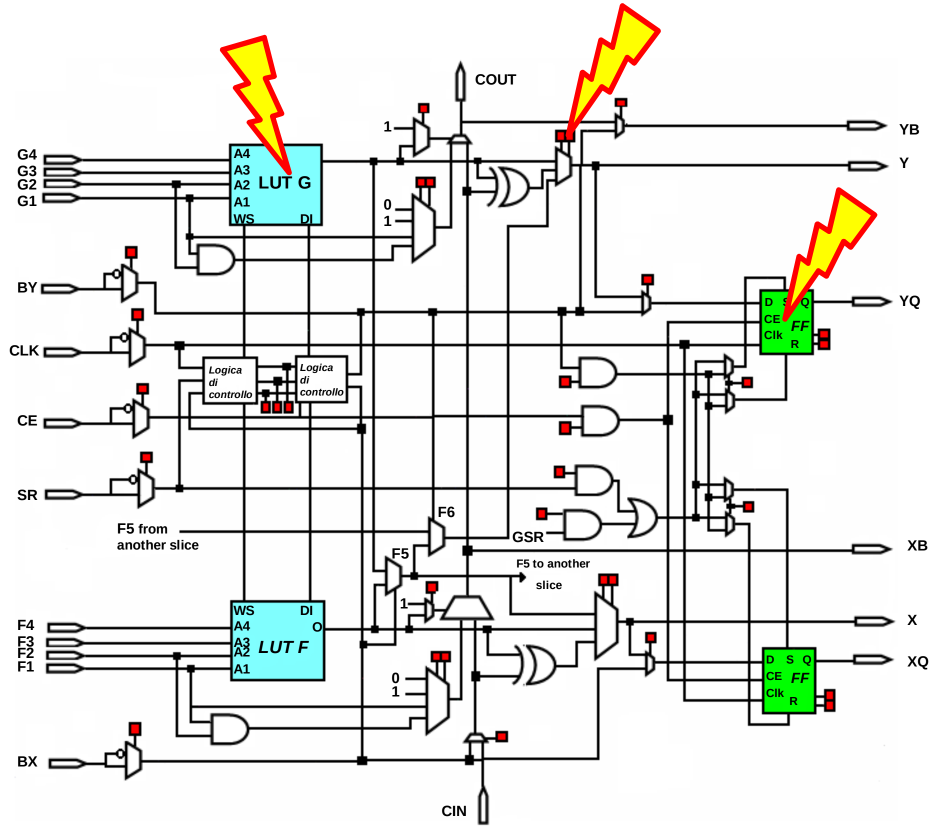
FLIPPER

- FLIPPER is funded by ESA
  - http://www.esa.int/TEC/Microelectronics/SEMP0NU681F_0.html
- FLIPPER injects bit-flips within the FPGA configuration memory by means of partial re-configuration
- The system consists of a hardware platform and a software application running on a PC
- DUT device is an XQR2V6000 hosted on a piggy-back board
  - TID tolerance up to 200 krad(Si)
  - SEL immunity LET > 160 MeV•cm2/mg
- Test vectors and reference values for the functional test of implemented designs are imported by the software application from an external HDL simulator



Fault Model

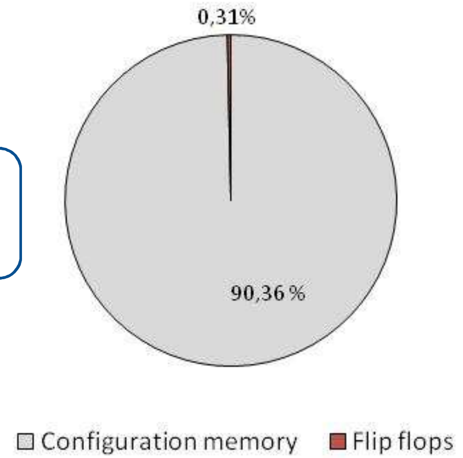
- Bit-flip of configuration memory cells
- Bit-flips in configuration memory may affect
  - Logic functions
  - Circuit topology



Fault Model

- Bit-flip of configuration memory cells
- Bit-flips in configuration memory may affect
  - Logic functions
  - Circuit topology
- Configuration memory represents the majority of device
- Accelerator validation of FLIPPER at PSI, November 2008

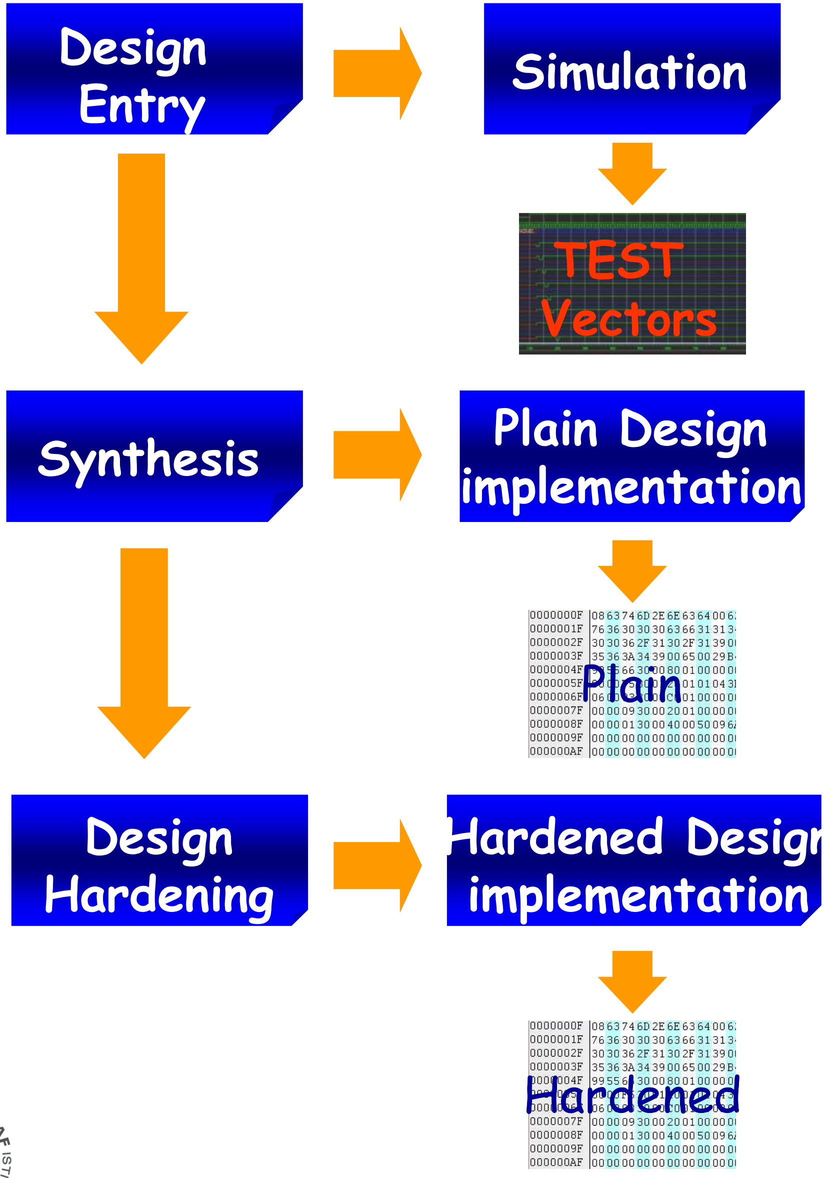
XQR2V6000

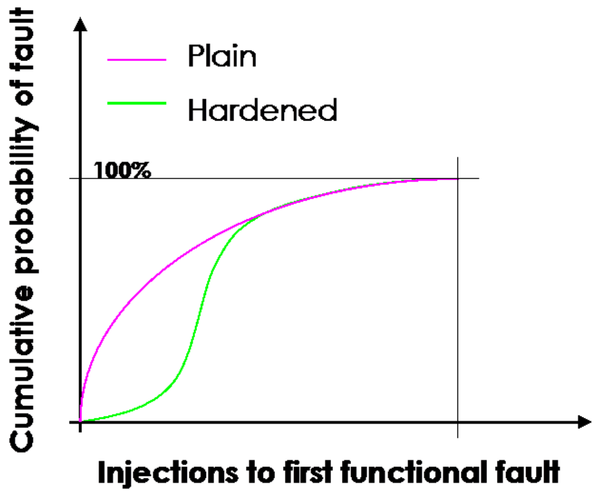


What's FLIPPER for

- Quantitative characterization of design robustness
- Workload dependant analysis of sensitive bits
- Comparison of design hardening techniques
- Tuning of design redundancy and protection
- Optimization of radiation ground testing

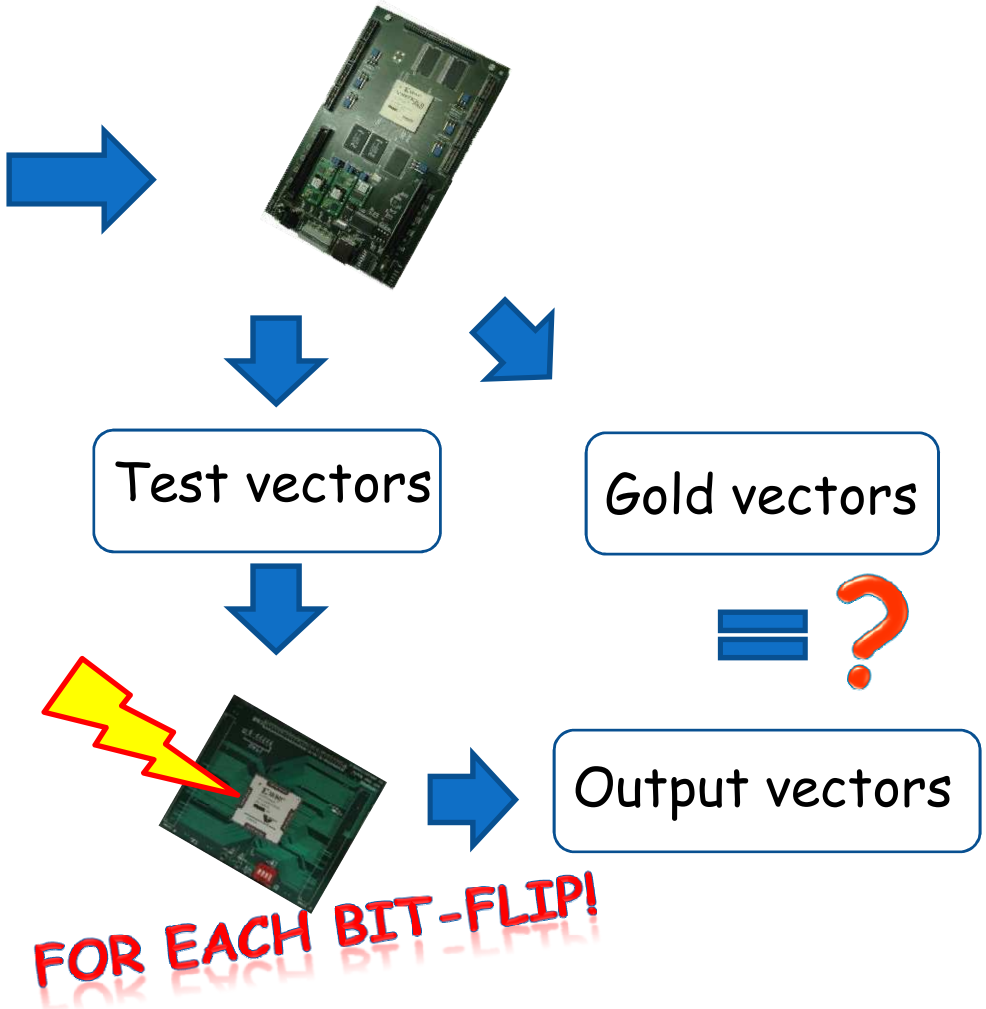
An Example


Design Entry
Simulation
TEST Vectors
Synthesis
Plain Design implementation
Plain
Design Hardening
Hardened Design implementation
Hardened


Cumulative probability of fault
Plain
Hardened
100%
Injections to first functional fault

Basic Test Procedure

Bit flip address set
Test and gold vectors
DUT bitstream



- Main steps:
  - Control Board initialization
  - DUT configuration
  - Bit-flip injection by partial configuration
  - Functional test
  - Failure log file analysis

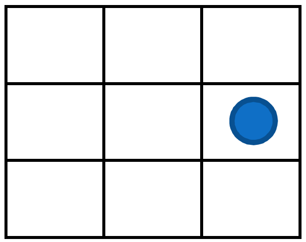
FI Campaigns

How configuration memory locations
to flip are chosen ?

- Systematic
  - identifies the design sensitive bits with respect to the applied set of test vectors
- Random
  - mimics the irradiation experiment (bit-flip accumulation)
- Specific
  - evaluates the impact of critical bits for a given workload

1	2	3
4	5	6
7	8	9

1		4
3	6	5
		2

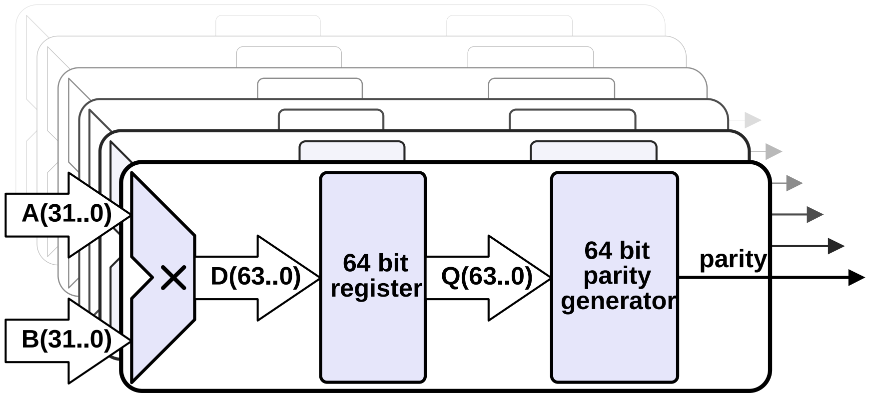


Systematic FI Campaign

- Identifies the *design critical bits* with respect to the applied set of test vectors
  - each and every configuration memory bit is addressed and flipped
  - the **altered bit is restored** before the successive injection is performed
- Results
  - list of critical bits (i.e. bits that, when flipped, cause a failure)

$$\sigma_{app\text{-}pseudo\text{-}stat} = \#\text{critical bits} \cdot \sigma_{bit}$$

Systematic FI Campaign: an example



1200 Test vector @10 MHz

Resources	MULT_36	MULT_18
IOBs [#]	102 (12%)	84 (10%)
LUTs [#]	40,957 (60%)	20,478 (30%)
FFs [#]	2,304 (3%)	1,152 (1%)

Host Device **XQR2V6000**

configuration cells = **19.742.976**

MULT_36

#critical bits = 1,799,480

$\sigma_{app\text{-}pseudo\text{-}stat} = 5.96 \cdot 10^{-8}$ cm^2

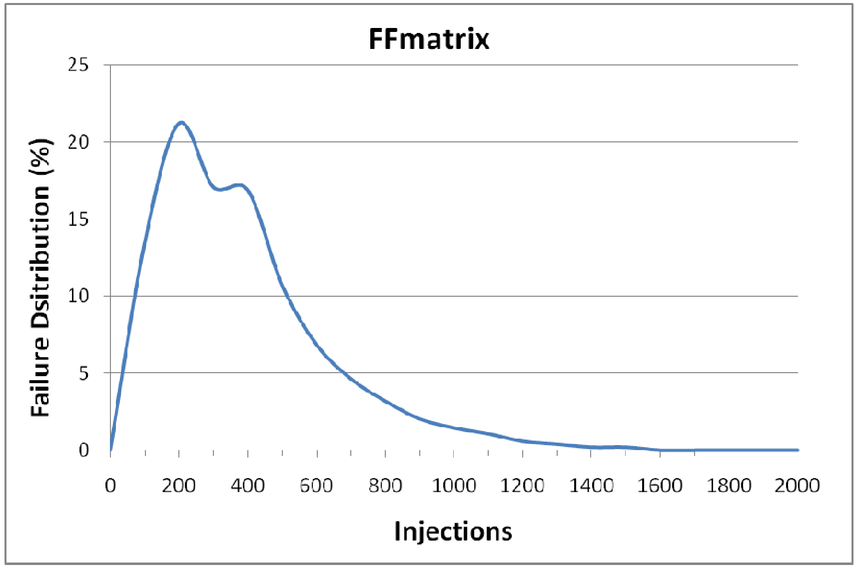
MULT_18

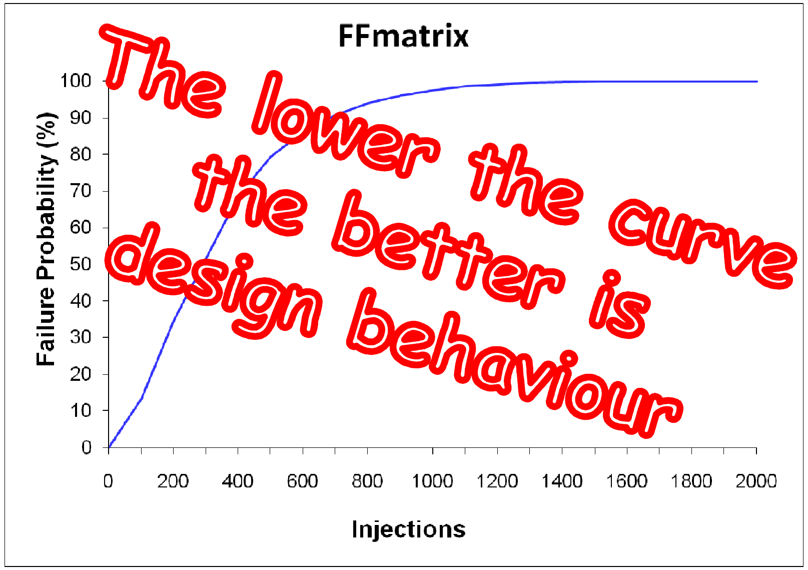
#critical bits = 874.599

$\sigma_{app\text{-}pseudo\text{-}stat} = 2.89 \cdot 10^{-8}$ cm^2

Random FI Campaign

- Random: mimics the irradiation experiment
  - configuration memory bits to be flipped are randomly addressed
  - the **altered bit is NOT** restored before the successive injection is performed
  - several injection RUNs
  - RUN -> injection procedure iterates until a predefined number of injection is reached or design failure occurs
- Results: Injections to failure distribution





How to define a Random FI campaign

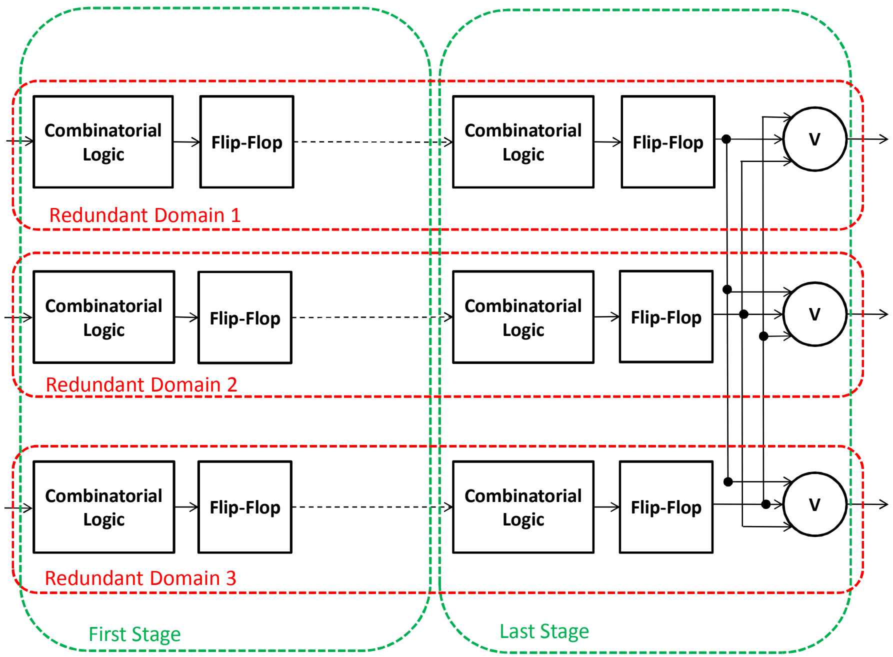
- Maximum number of injections per RUN?
  - "High" -> accumulation effects are highlighted
  - "Low" -> realistic application case, upsets into configuration memory should not be allowed to accumulate
- Whole circuit or "per module" analysis?
  - Modules are defined by output partitions
  - Injections are always performed into the whole configuration memory
  - Failed modules are ruled out (by dynamically masking their output)
  - SEU sensitivity analysis of different design parts can be easily accomplished

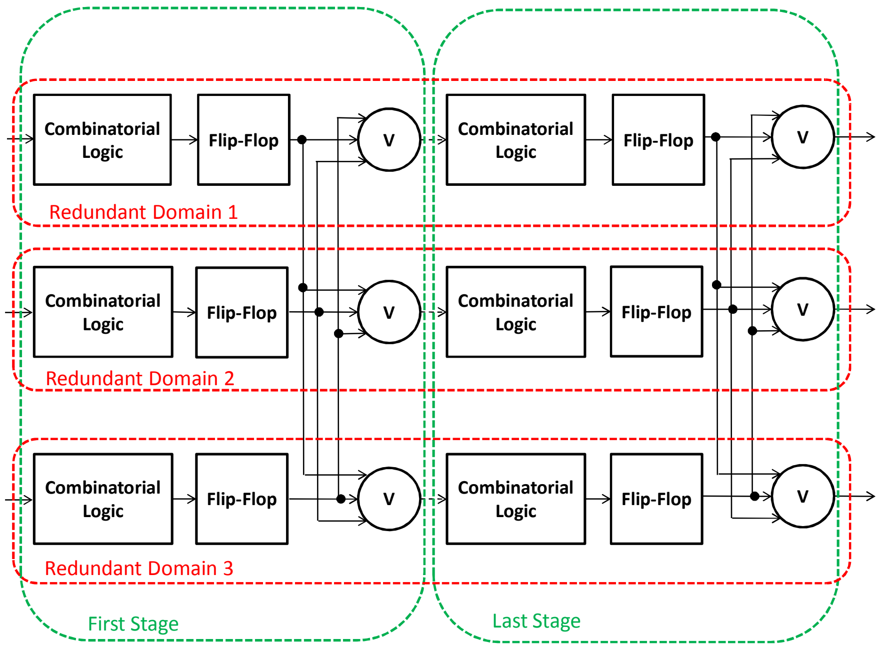
Random FI Campaign: an example

- ESA benchmark design consisting of modules
  - FFT: Fourier Transform of a data matrix
  - MULT16_LUT: 2-stage 16x16 bit multiplier instantiated twice
  - MULT16_MULT18: 10-stage 16x16 bit multiplier instantiated twice (embedded)
  - FFmatrix: two identical copies of a shift register chain (480 bits each)
  - ROMff: two copies of a shift register (256 bit each); the former is loaded and holds the stored values, the latter reads the values stored by the former

Example: V1 and V2 variants

- V1 and V2 design variants
  - V1 is a TMR version of the plain design, voters are inserted only in the last stage and after flip-flops with feedback paths.
  - In V2 voter are inserted after EACH flip-flop





Example: resource usage

- FPGA resource usage

Host Device **XQR2V6000**

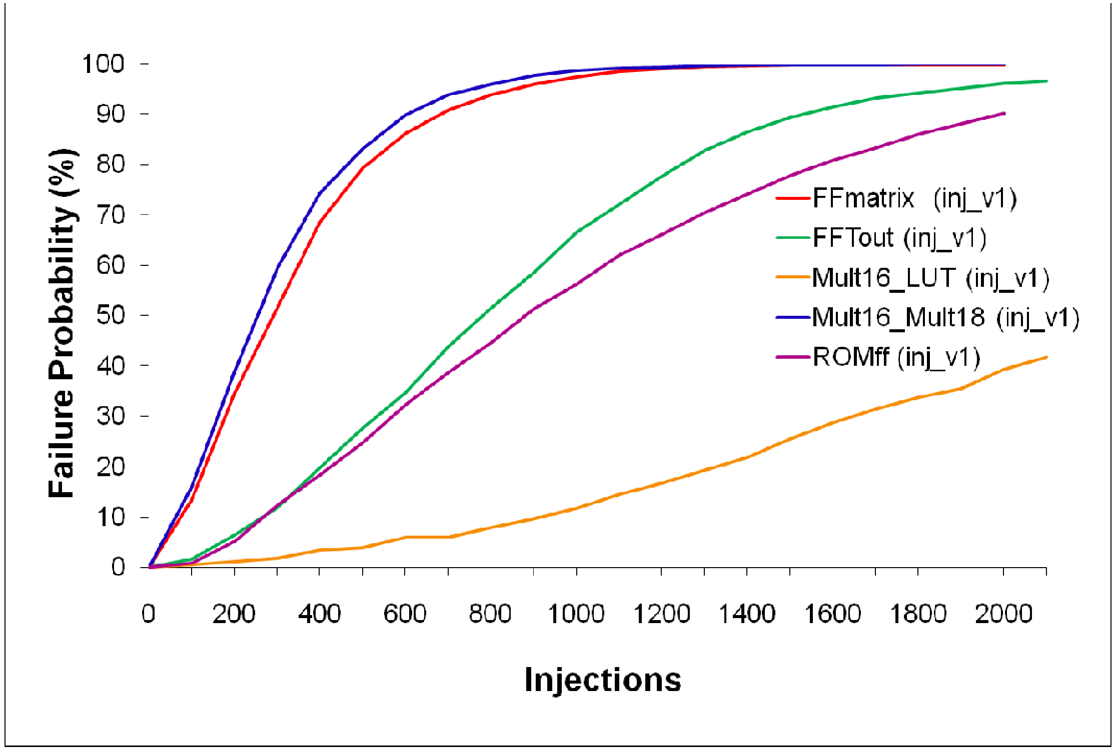
configuration cells

19.742.976

Resource	Plain	V1	V2
FF	2,926 (4%)	8,778 (12%)	8,778 (12%)
LUT	3,806 (5%)	13,437 (19%)	29,217 (43%)
IOB	87 (10%)	264 (32%)	267 (32%)
MULT 18x18	32 (22%)	96 (66%)	96 (66%)
GCLK	1 (6%)	3 18%)	3 (18%)

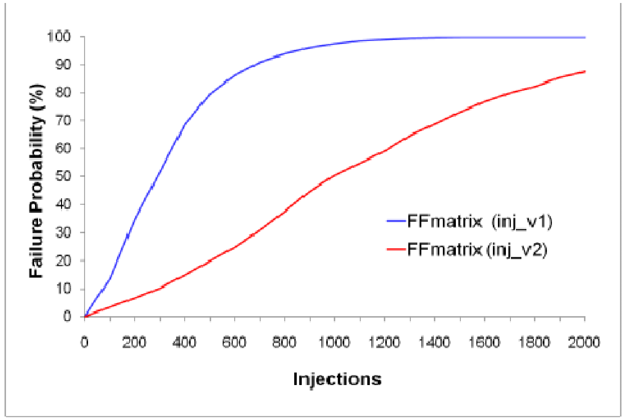
	FFmatrix	
	V1	**V2**
FF (DFF)	3,313	3,313
LUT (FG)	813	7,437
MULT 18x18	0	0
	Mult16_LUT	
	V1	**V2**
FF (DFF)	579	579
LUT (FG)	3543	4701
MULT 18x18	0	0
	FFTout	
	V1	**V2**
FF (DFF)	1080	1080
LUT (FG)	5382	6468
MULT 18x18	36	36
	Mult16_Mult18	
	V1	**V2**
FF (DFF)	2139	2139
LUT (FG)	507	4785
MULT 18x18	60	60
	ROMff	
	V1	**V2**
FF (DFF)	1572	1572
LUT (FG)	2421	4758
MULT 18x18	0	0

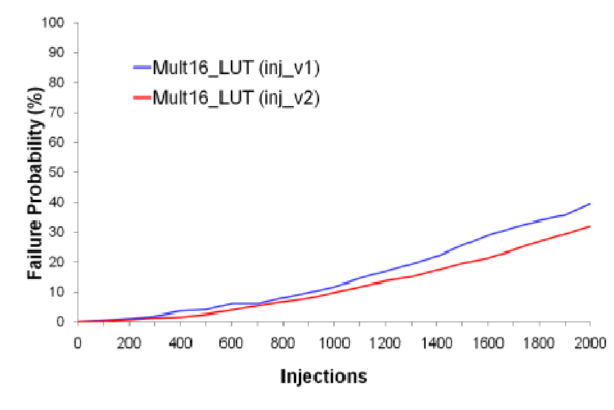
Example: per module results

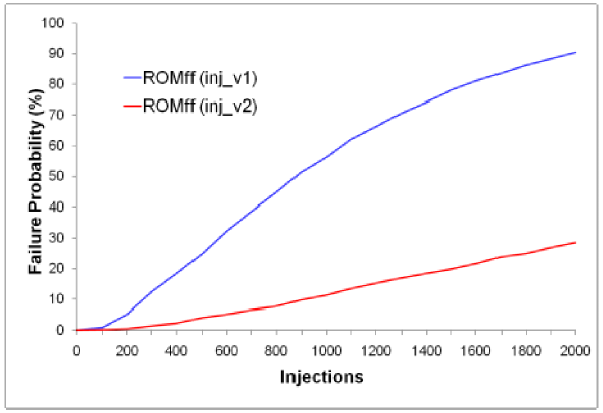


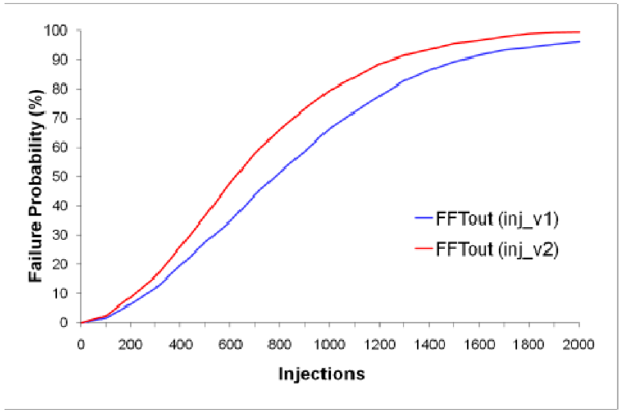
- Max 100k injections per RUN
- 28000 test vector @10 MHz
- Per module analysis
  - MOST sensitive "module" -> Mult16_Mult18
  - LESS sensitive "module" -> Mult16_LUT

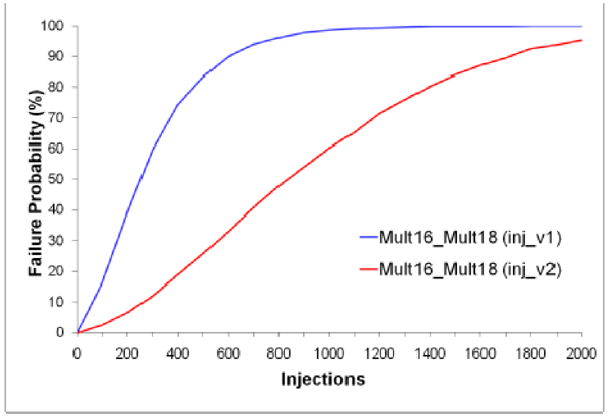
Example: V1 and V2 results











- General behaviour
  - V2 better than V1 with different grade depending on module
- Exception
  - FFT, not completely surprising -> bit-flips accumulation invalidates the redundant domains independence

Specific FI Campaign

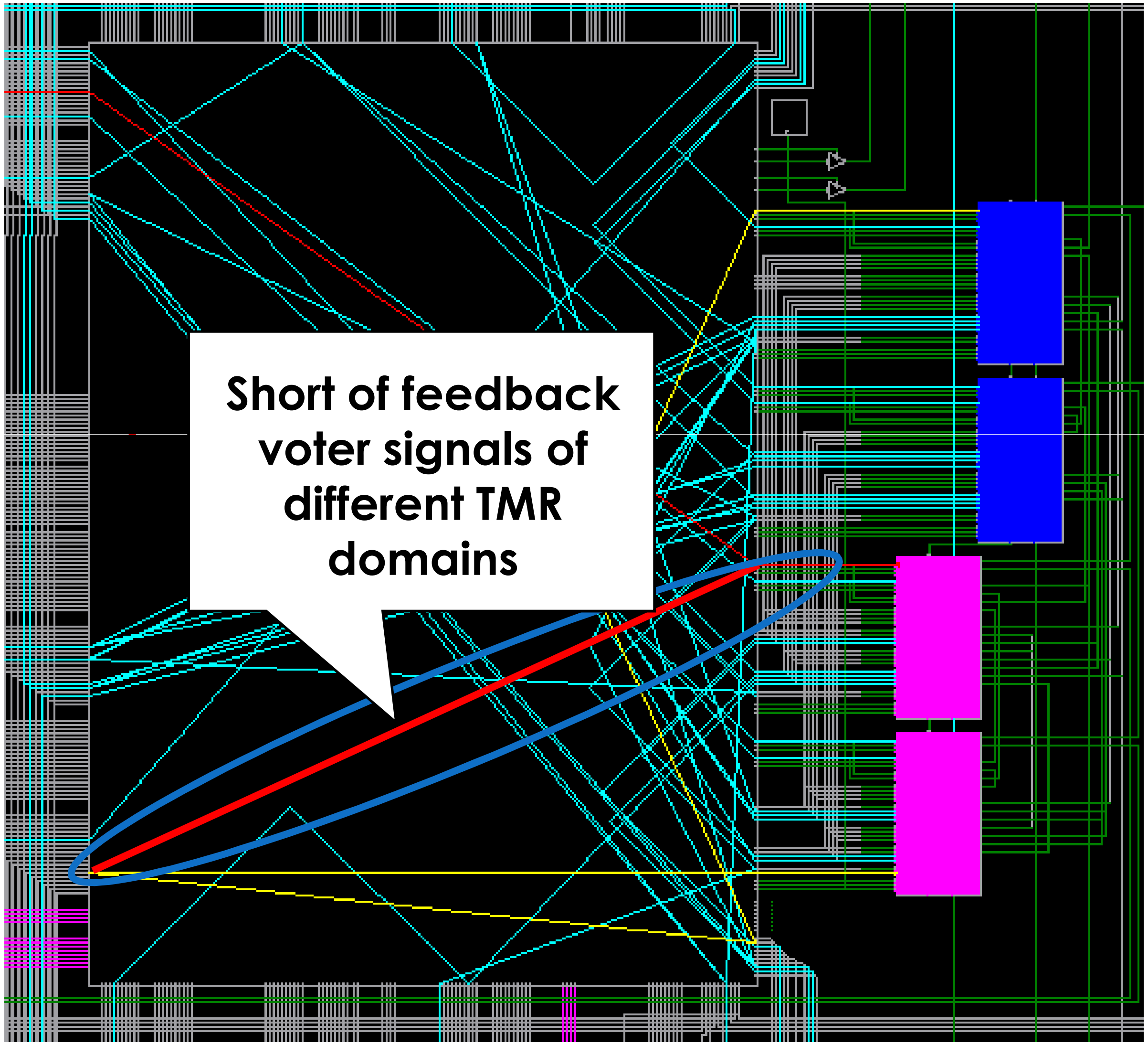
- Evaluates the impact of critical bits for a given workload
  - selected bits in the configuration memory are injected
  - the altered bit is restored before the successive injection is performed
- Results: list of sensitive bits w.r.t. a selected workload
- Example
  - Simple 8 bit counter protected by TMR
  - List of critical bit idenfied by STAR (Static Analysis Tool by Politecnico di Torino)

X-TMR Circuit	CLB [#]	IOBs [#]	Slices [#]	LUTs [#]	FFs [#]
COUNT8	33	90	130	144	120

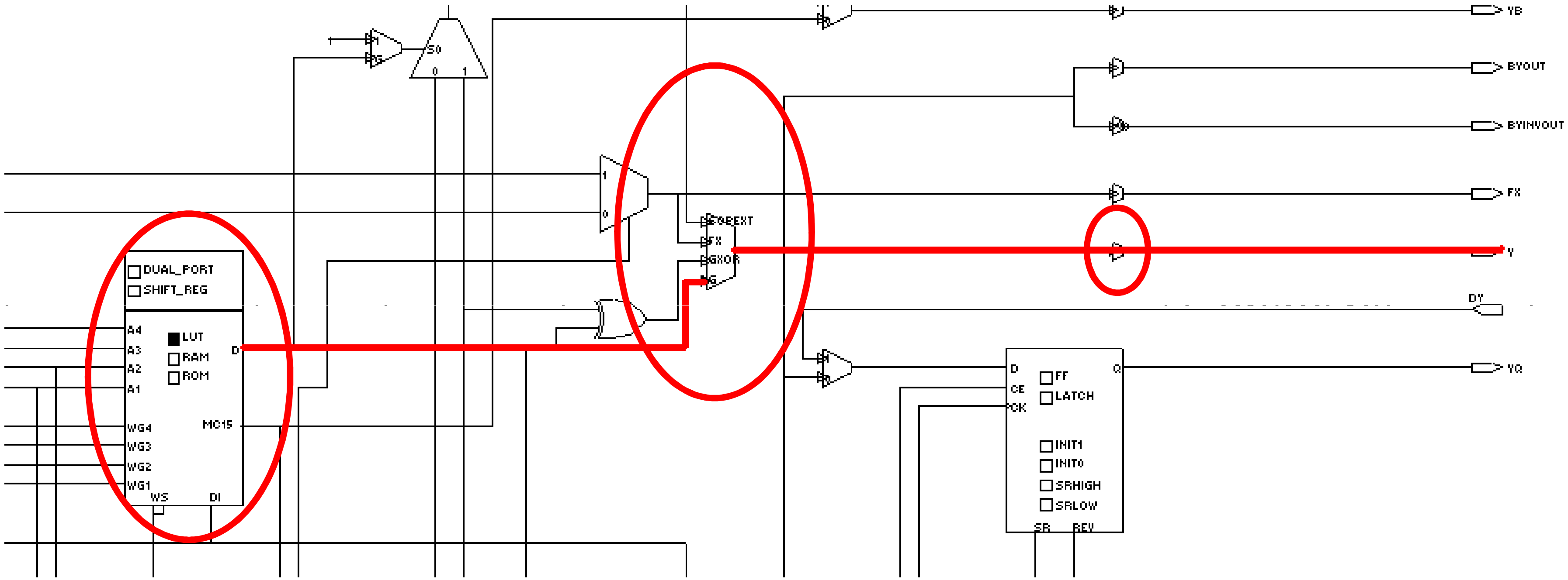
Results - COUNT8

Bit position	STAR	FLIPPER	Resource	CLB coordinates	Fault type
10,510,980	x		SM	R[15]C[60]	Short PIP OMUX14 ↔ XQ0
10,637,088	x	x	LUT	R[17]C[61]	LUT first bit upset
10,629,222	x	x	MUX Y	R[17]C[61]	Control bit upset
10,629,230	x	x	MUX OUT	R[17]C[61]	Control bit upset

Results - COUNT8



Results - COUNT8



Future of the work

- Improve system performances
- Upgrade FLIPPER to further device families and accelerator validation
- Improve the integrated FLIPPER/STAR-RoRA flow for SEU susceptibility analysis (ESA)

References

- C.R. Yount, D.P. Sieworek, "A methodology for the rapid injection of transient hardware errors", IEEE Trans. on Computers, vol. 45. n.8, August 1996, pp. 881-891.
- P.E. Dodd and L.W. Massengill, "Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics", IEEE Trans. on Nucl. Sci., vol. 50, n. 3, pp 583-602, June 2003.
- M. Caffrey, P. Graham, E. Johnson and M. Wirthlin,, "Single-Event Upsets in SRAM FPGAs", in Proc. of the Military and Aerospace Applications of Programmable Devices Int'l Conference (MAPLD), September 2002.
- M. Alderighi, F. Casini, S. D'Angelo, M. Mancini, S. Pastore, G.R. Sechi, and R. Weigand, "Evaluation of Single Event Upset Mitigation Schemes for SRAM based FPGAs using the FLIPPER Fault Injection Platform", Proc. of the 22th IEEE Int'l Symp. on Defect and Fault Tolerance in VLSI Systems, Rome, Italy, pp. 105-113, Sept. 2007.
- M. Alderighi, F. Casini, S. D'Angelo, M. Mancini, S. Pastore, L. Sterpone, and M. Violante, "Soft errors in SRAM-based FPGAs: a comparison of two complementary approaches", IEEE Trans. on Nucl. Sci., vol. 55, n. 4, August 2008, pp. 2267-2273.
- M. Alderighi, F. Casini, M. Citterio, S. D'Angelo, M. Mancini, S. Pastore, G.R. Sechi, G. Sorrenti, "Using FLIPPER to Predict Irradiation Results for VIRTEX 2 Devices", Proceedings of the 2008 European Workshop on Radiation Effects on Components and Systems, Jyväskylä, Finland, Sept. 10-12, 2008.
- M. Alderighi, F. Casini, S. D'Angelo, M. Mancini, D. Merodio Codinachs, S. Pastore, G. Sorrenti, L. Sterpone, R. Weigand, and M. Violante, "Robustness analysis of soft error accumulation in SRAM-FPGAs using FLIPPER and STAR/RoRA", Proceedings of the 2008 European Workshop on Radiation Effects on Components and Systems, Jyväskylä, Finland, Sept. 10-12, 2008.
- M. Alderighi, F. Casini, M. Citterio, S. D'Angelo, M. Mancini, S. Pastore, G.R. Sechi, G. Sorrenti, "Using FLIPPER to Predict Irradiation Results for VIRTEX 2 Devices", Proceedings of the 2008 European Workshop on Radiation Effects on Components and Systems, Jyväskylä, Finland, Sept. 10-12, 2008.
- M. Alderighi, F. Casini, M. Citterio, S. D'Angelo, M. Mancini, S. Pastore, G.R. Sechi, G. Sorrenti,"Using FLIPPER to Predict Proton Irradiation Results for VIRTEX 2 Devices: a Case Study", IEEE Trans. on Nucl. Sci., in print.