Towards reliable FPGA-based satellite systems - the RUSH experiment

Ediz Cetin
Department of Engineering

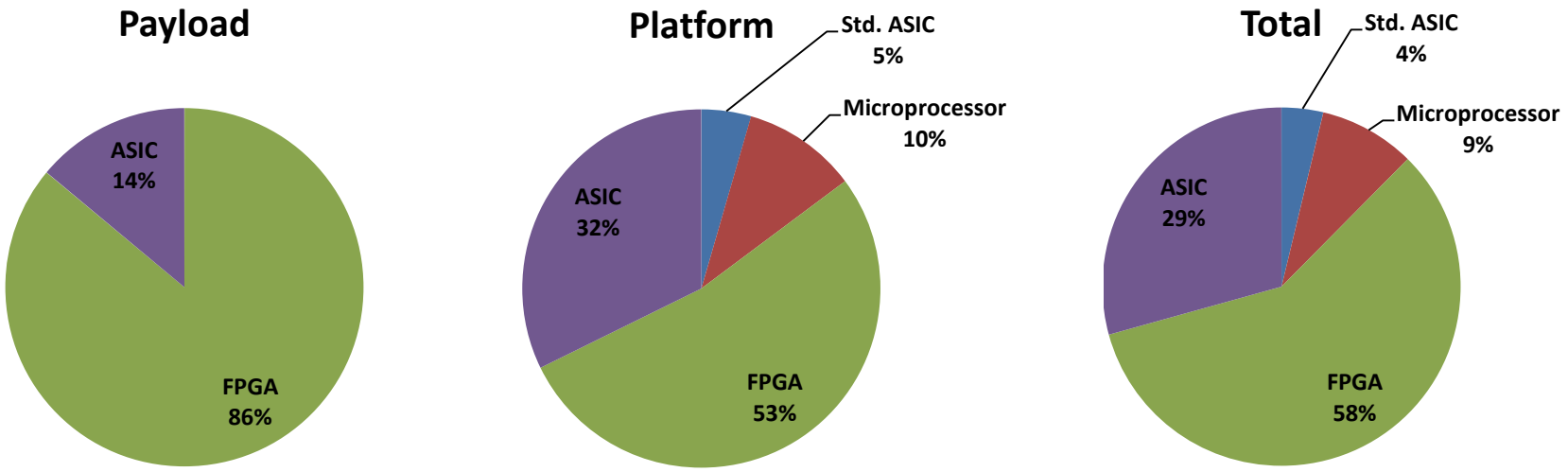
Email: ediz.cetin@mq.edu.au

19 April 2017

COTS FPGAs in Space

- The processing speed, cost and flexibility requirements of future satellite-based applications cannot be satisfied with conventional radiation-hardened processors or custom integrated circuits.
- Growing international interest in the development of space missions based on low-cost nano-/microsatellites demands new approaches to the design of reliable, low-cost, reconfigurable digital processing platforms
- SRAM-based Field Programmable Gate Arrays (FPGAs) provide an opportunity for meeting these requirements with off-the-shelf hardware.
- SRAM-FPGAs ride Moore's law (unlike radiation-hardened counterparts)
- As a group we are undertaking research into designing highly reliable COTS SRAM-FPGA based reconfigurable systems for space applications

ESA Sentinel 2 Mission

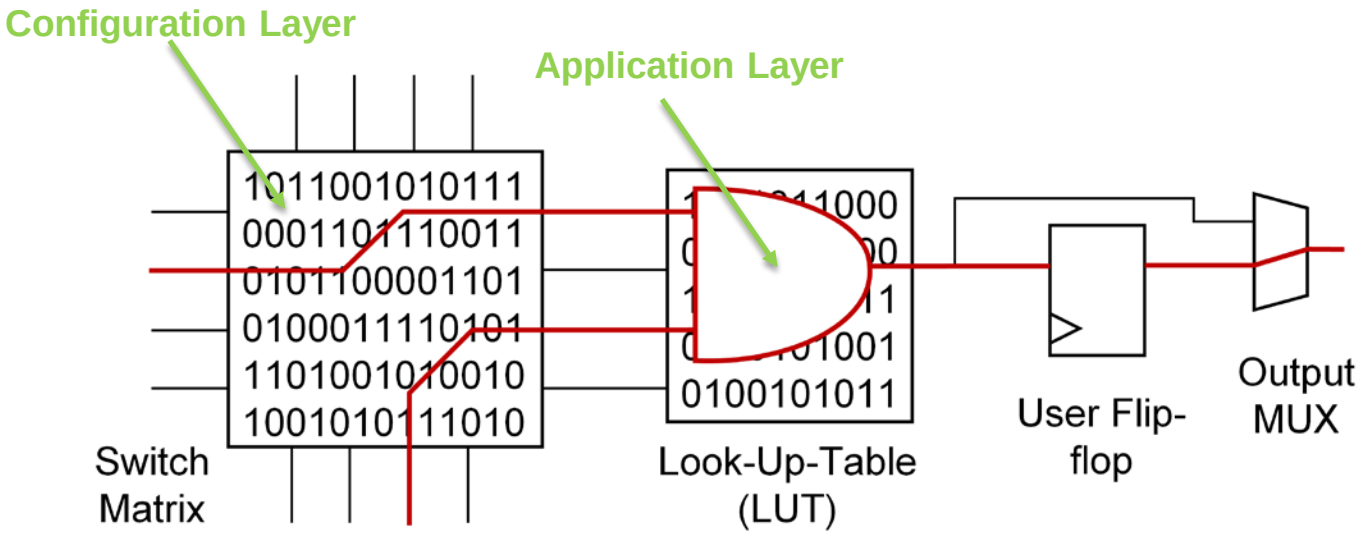


IC Type	Quantity Platform	Quantity Payload
FPGA	**118**	**37**
Custom ASIC	72	6
Microprocessor	23	0
Standard ASIC	10	0

* **Data from:** "Trends and patterns in ASIC and FPGA use in space missions and impact in technology roadmaps of the European Space Agency", Roger Boada Gardenyes, Master Thesis, T. U. Delft and ESA, 15th August 2012

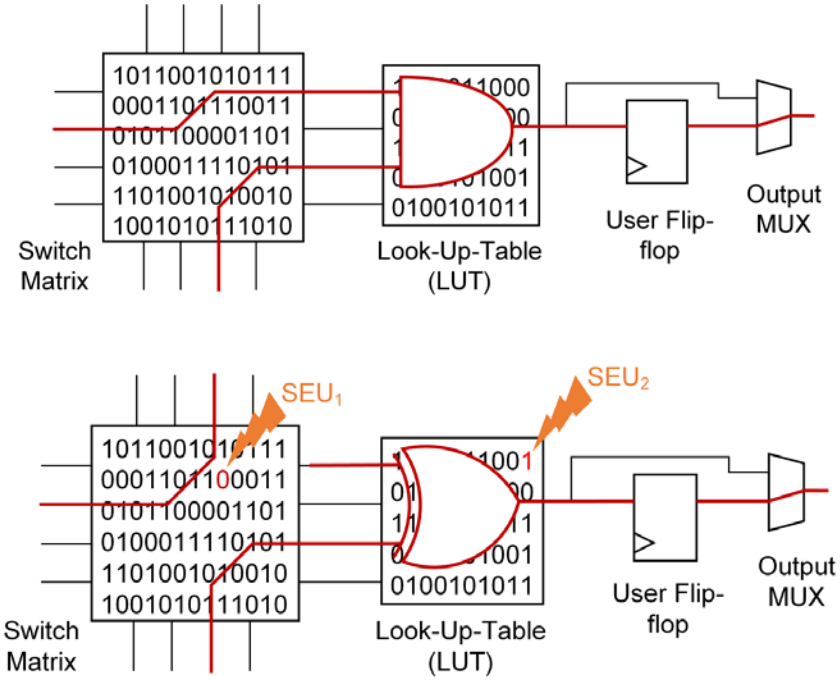
FPGA Fabric and the SEUs

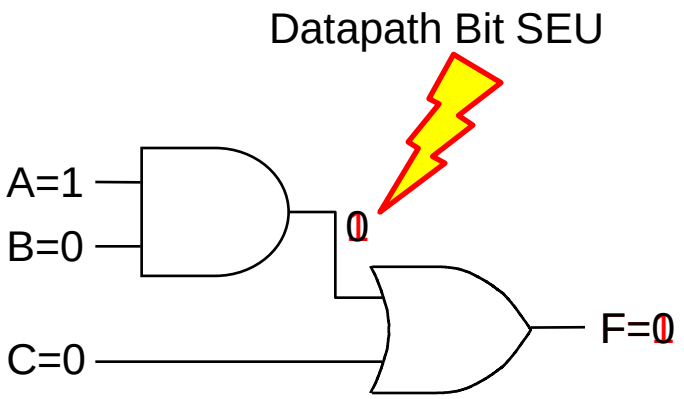
- SRAM-based FPGA can be thought of consisting of two layers:
  - **an application layer:** Where user circuits live and operate
  - **a configuration layer:** Where specific information is written or "downloaded" into the SRAM-based configuration memory to configure the FPGA to implement user applications (in the application layer).



FPGA Fabric and the SEUs

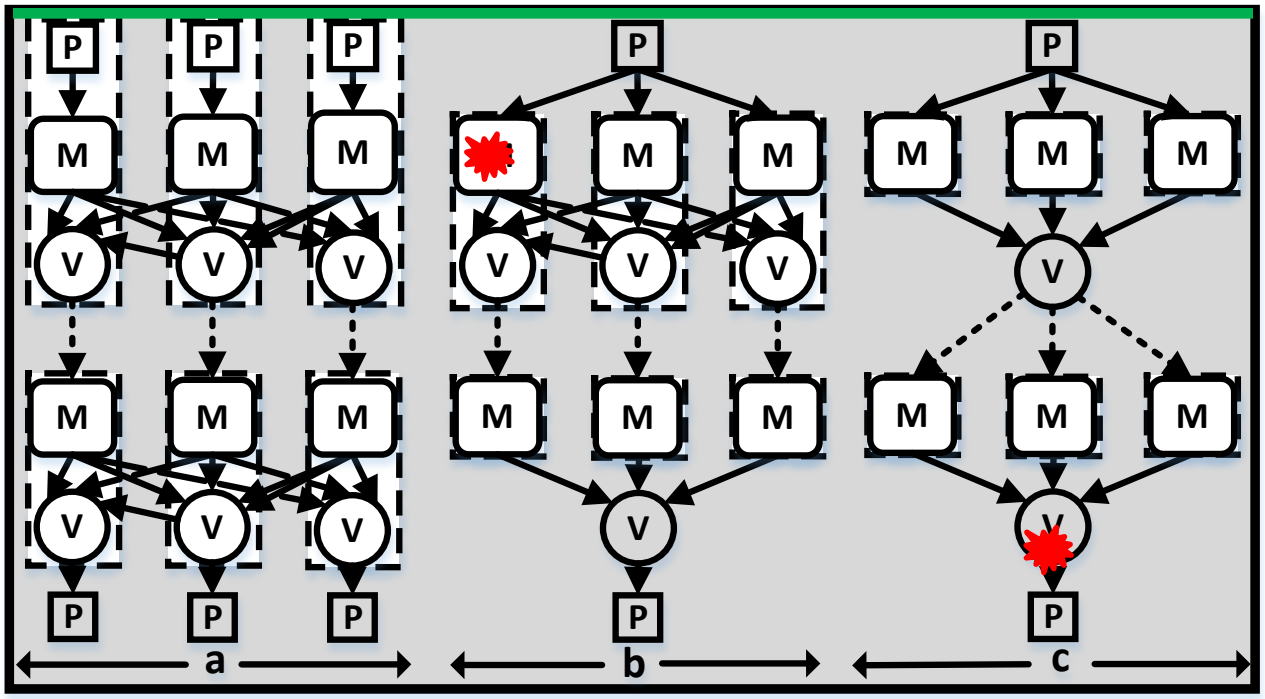
- FPGAs, however, are particularly susceptible to radiation-induced Single Event Upsets (SEUs):
  - Deposited charge causes a change of state in dynamic circuit elements
  - Affects both datapath (application layer) and configuration memory (configuration layer)





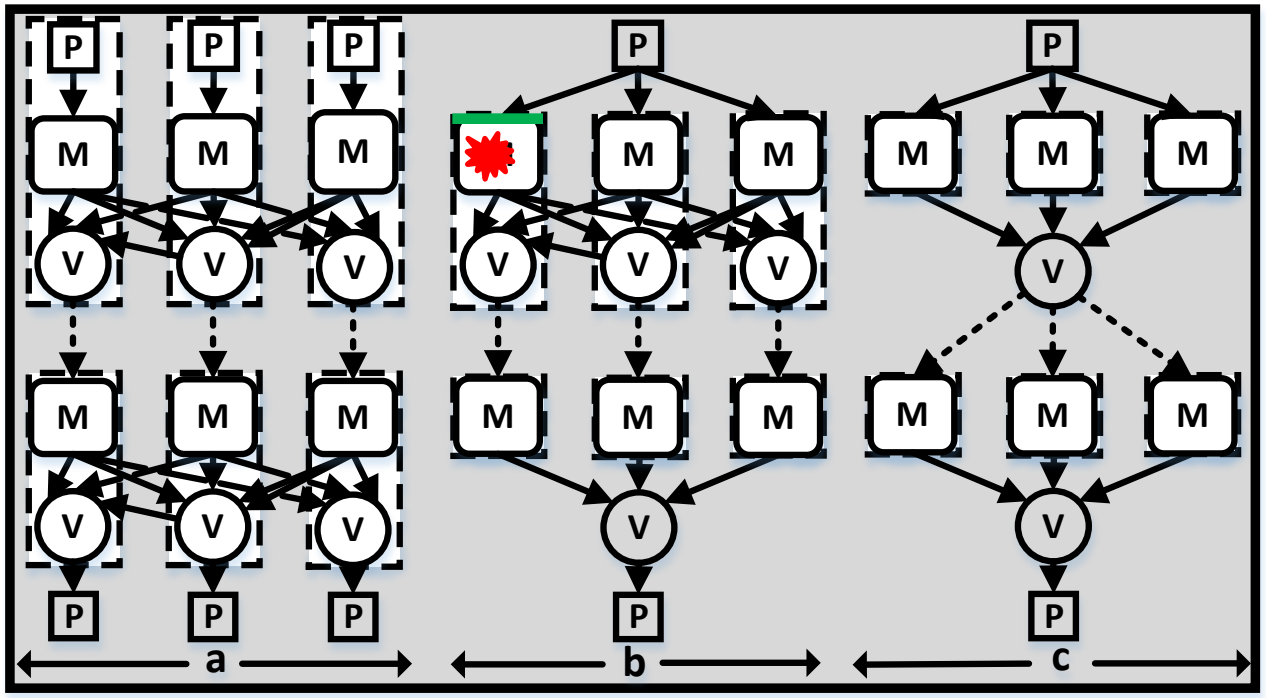
Configuration Memory Error Recovery - Scrubbing

- Energy wasted since the system is “scrubbed” even if there are no errors
- Increased Mean Time to Detect (MTTD) Errors
- Robust error recovery technique



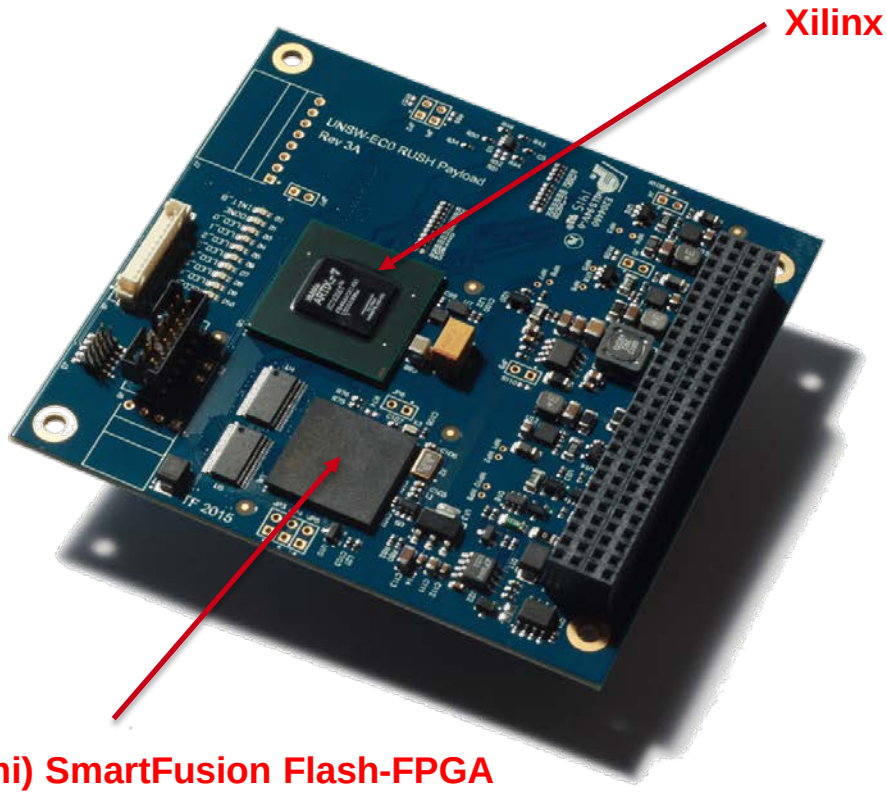
Configuration Memory Error Recovery – Module Based

- Low energy consumption
- Rapid MTTD
- No error recovery in the configuration bits of the shaded region



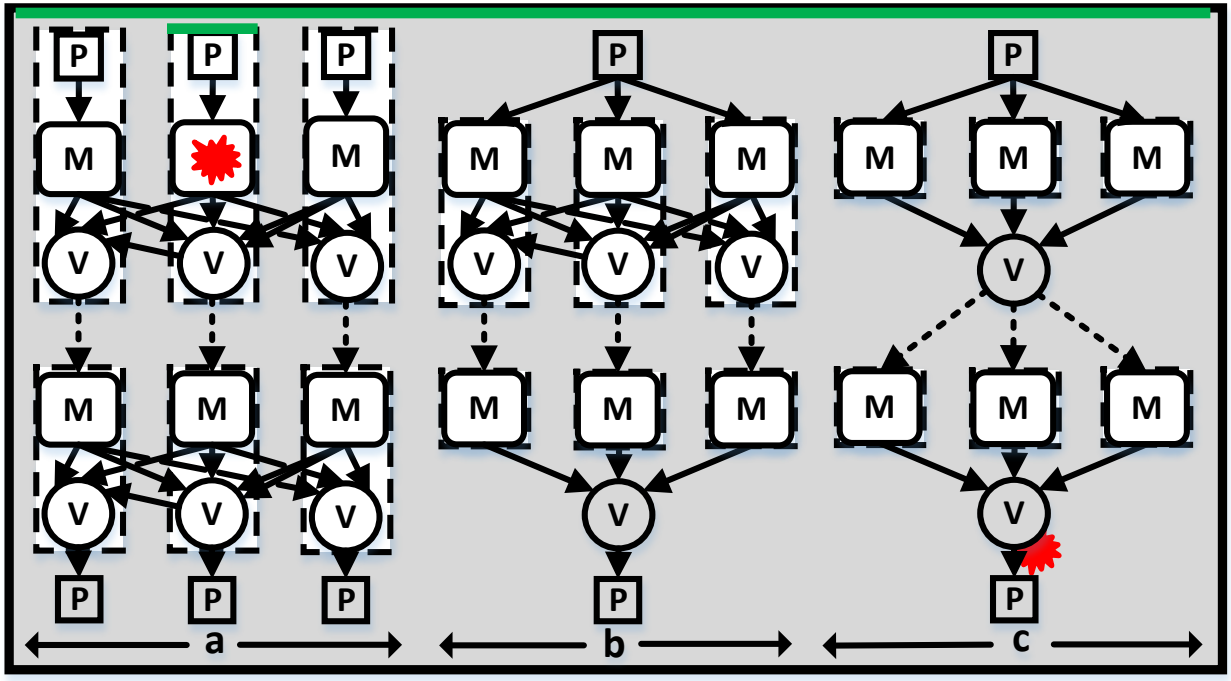
RUSH Payload

- Will test and compare traditional “scrubbing” approach with Module Based Error Recovery (MER) approach.



Research Questions Post RUSH Payload

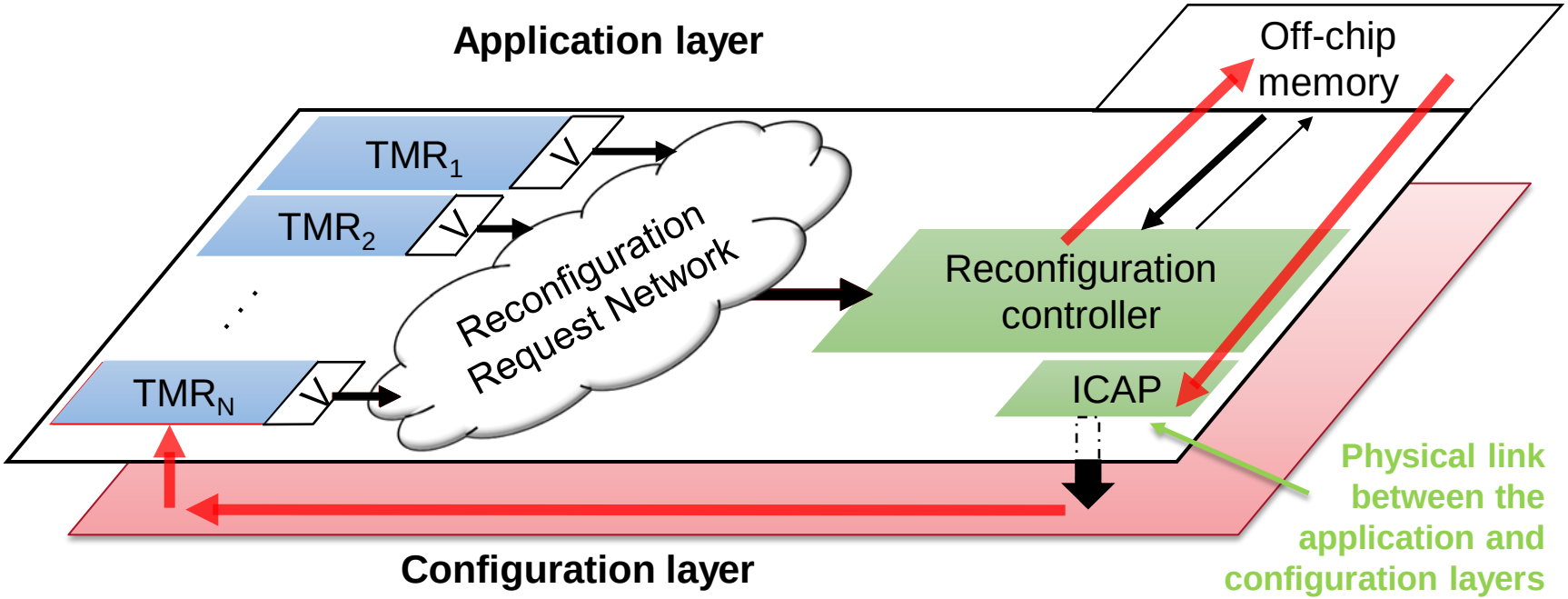
- How do you deal with non-triplicated resources (or resources that cannot be triplicated)?
  - Use **FMER**: A hybrid error recovery technique (best of both worlds)



See: Agiakatsikas *et. al*, "FMER: A Hybrid Configuration Memory Error Recovery Scheme for Highly Reliable FPGA SoCs. 26th International Conference on Field-Programmable Logic and Applications (FPL'16), 2016.

Research Questions Post RUSH Payload

- In which **order** and **how** should one check the TMR voters?



See: Nguyen *et.al.* "Dynamic Scheduling of Voter Checks in FPGA-based TMR Systems", International Conference on Field-Programmable Technology (FPT'16),pp.169–172.

Zhao *et. al.*, "Fine-grained Module-based Error Recovery in FPGA-based TMR Systems", International Conference on Field-Programmable Technology (FPT'16), pp 101 – 108.

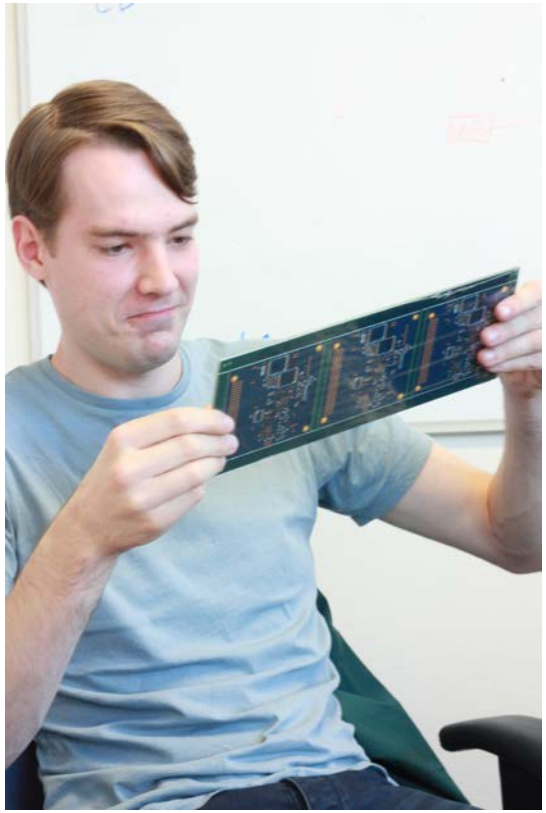
Agiakatsikas *et. al.*, "Reconfiguration Control Networks for TMR Systems with Module-based Recovery", IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'16), pages 88 – 91, 2016.

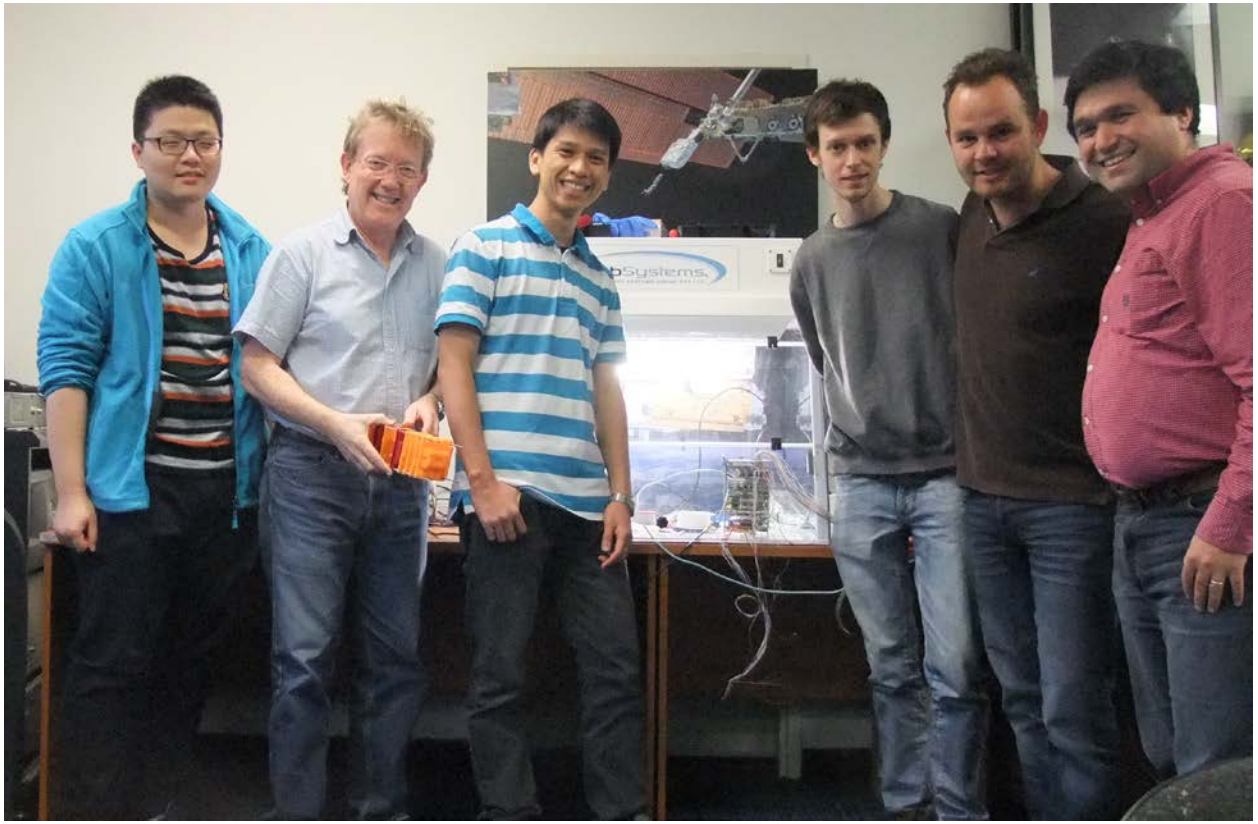
Do I need to be an FPGA and Reliability Expert to Implement Fault-tolerant Circuits on FPGAs?

- Yes but not for long ! (We hope ☺)
- Introducing **TLegUp** – Given high level C code, generates low level highly reliable FPGA implementations that are already triplicated and ready to deploy on FPGAs.

See: Lee *et. al*, “TLegUp: A TMR Code Generation Tool for SRAM-Based FPGA Applications Using HLS”, To appear *IEEE Symposium on Field-Programmable Custom Computing Machines* (FCCM’17), May 2017.

Team Members (Partial)





Funding

- Research work supported by the Australian Research Council:
  - ARC Linkage Grant (LP140100328)
  - ARC Discovery Grant (DP150103866)

- Partners:

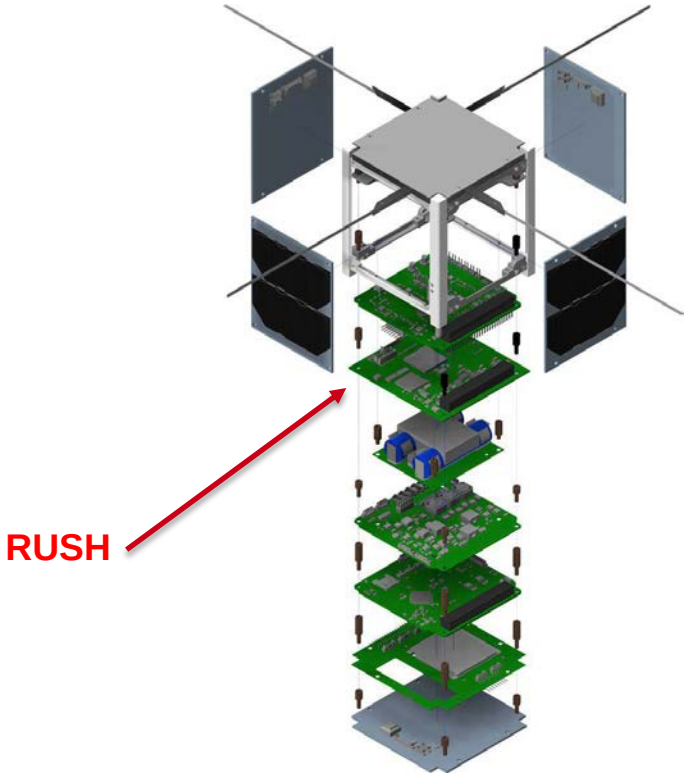
Australian Government
Australian Research Council

ThalesAlenia Space
A Thales / Finmeccanica Company

GENERAL DYNAMICS

What's Next for RUSH

- Payload on FloripaSat (Late 2017) in Collaboration with Federal University of Santa Catarina (UFSC) Brazil



- **RUSH-SDR** : Dual RF channel 70 MHz – 6 GHz Software Defined Radio (SDR) with on-board FPGA-based processing (under development)
- **Research:** Looking for use cases from the CubeSat Community