MAPLD 2008

Annapolis, Maryland

Hardening-by-design techniques using residue number system in SRAM-based FPGAs: an experiment on a FIR filter

S. Pontarelli, G.C. Cardarilli, A. Salsano
Università di Roma "Tor Vergata", Roma, Italy

S. Gerardin, A. Manuzzato, A. Paccagnella
DEI - Università di Padova, Padova, Italy

Outline

- **Introduction**
- **Traditional hardening approach**
  - basic idea and drawbacks
- **Residue Number System**
  - definition and properties
  - error detection and correction capabilities
- **Case study: hardening a FIR filter**
  - hardened circuit
  - irradiation experiments
- **Conclusions**

Introduction

THE PROBLEM: the use of SRAM-based FPGAs in harsh **radiation** environment is limited by the **susceptibility** of such devices to radiation effects

Single Event Upsets affecting the configuration memory can alter the implemented circuit functionality

"**THE SOLUTION**": if we want to use commercial SRAM-based FPGAs in radiation environments, **hardening-by-design** techniques are mandatory to preserve the correct circuit functionality

Traditional Approach: TMR

A widely used approach to improve the system reliability is the **Triple Modular Redundancy** (TMR) technique

The idea: all the logic is tripled and a majority voter chooses the circuit outputs, masking errors to the outside world

TMR drawbacks and issues

This approach is very area expensive!

- **area** increases more than 3 times
- **power** consumption increases (tripled logic, tripled clock distribution…)
- needs triple **I/Os** (plus voltage references)
- degradation in **timing performance**

Problems affecting this hardening technique:

- **Multiple Bit Upsets** (MBUs) can alter simultaneously two redundant domains [*Quinn et al. TNS Dec. 2007*]
- In the configuration memory there are bits controlling multiple resources → **single point of failure** (due to the FPGA architecture) [*Sterpone et al*. TNS Aug. 2006]

Residue Number System

A **Residue Number System** is defined by a set of relatively prime integers

*In this work we present a hardening technique based on the **Residue Number System** to implement FIR filters with error detection and correction capabilities*

Binary →RNS

$$x \xrightarrow{RNS} \begin{cases} x_1 = x \bmod m_1 \\ x_2 = x \bmod m_2 \\ \vdots \\ x_p = x \bmod m_p \end{cases}$$

$$x \in [0, M-1], M = \prod_1^P m_i$$

Operations in the RNS domain

$$Z = \mathrm{X}\ \mathrm{op}\ \mathrm{Y} \xrightarrow{\mathrm{RNS}} \begin{cases} Z \bmod m_1 = (X_{m_1}\ op\ Y_{m_1}) \bmod m_1 \\ \vdots \\ Z \bmod m_p = (X_{m_p}\ op\ Y_{m_p}) \bmod m_p \end{cases}$$

Modular organization, no hierarchy, reduced length

RNS → Binary

Conversion by the Chinese Remainder Theorem (CRT)

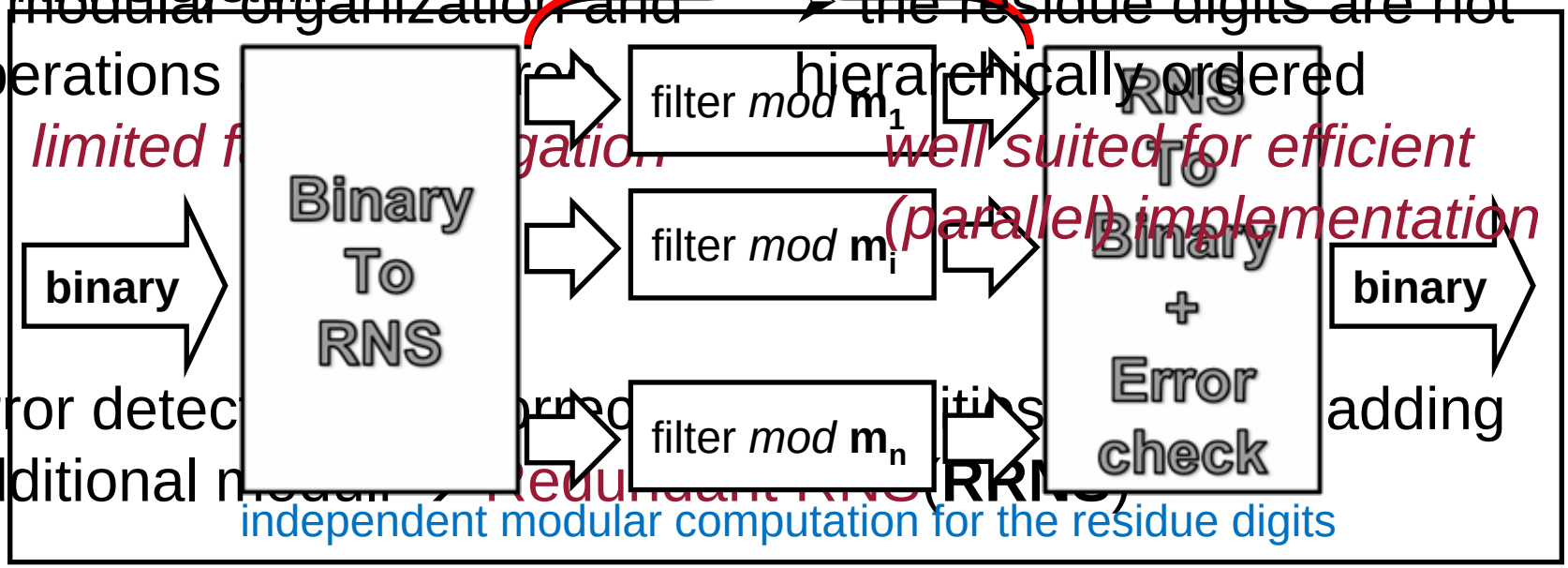
Why RNS ?

Operations in the RNS domain :

$$Z = X \text{ op } Y \xrightarrow{\text{RNS}} \begin{cases} Z \bmod m_1 = (X_{m_1} op\ Y_{m_1}) \bmod m_1 \\ \vdots \\ Z \bmod m_p = (X_{m_p} op\ Y_{m_p}) \bmod m_p \end{cases}$$

RNS domain

- RRNS system
  - modular organization and operations
  - the residue digits are not hierarchically ordered
- *limited f… gation*
- *well suited for efficient (parallel) implementation*



Error detec… adding additional m… → Redundant RNS (RRNS)

RNS background

RRNS moduli : $\{m_1, m_2, \ldots, m_k, m_{k+1}, \ldots, m_{k+r}\}$

normal moduli — redundant moduli

*Minimum to correct 1 error : **3** moduli + **2** redundant moduli*

The normal moduli define the system dynamic range M:

$$M = \prod_1^k m_i \Rightarrow x \in [0, M-1]$$

While the product of all the moduli defines the total range M_T

$$M_T = \prod_{i=1}^{k+r} m_i$$

$$[0, \ldots, M-1, M, \ldots, M_T - 1]$$

legitimate range — illegitimate range

Error detection and correction

Important properties stand for the ***m_i-projection***, defined as:

$$X_{mi} = X \bmod \left(\frac{M_T}{m_i} \right) = CRT\left(\underbrace{x_{m_1}, x_{m_2}, \ldots, x_{m_i\text{-}1}, x_{m_{i+1}}, \ldots, x_{m_{k+r}}}_{\text{residue vector representation of X with the residue digit } \boldsymbol{i} \text{ deleted}} \right)$$

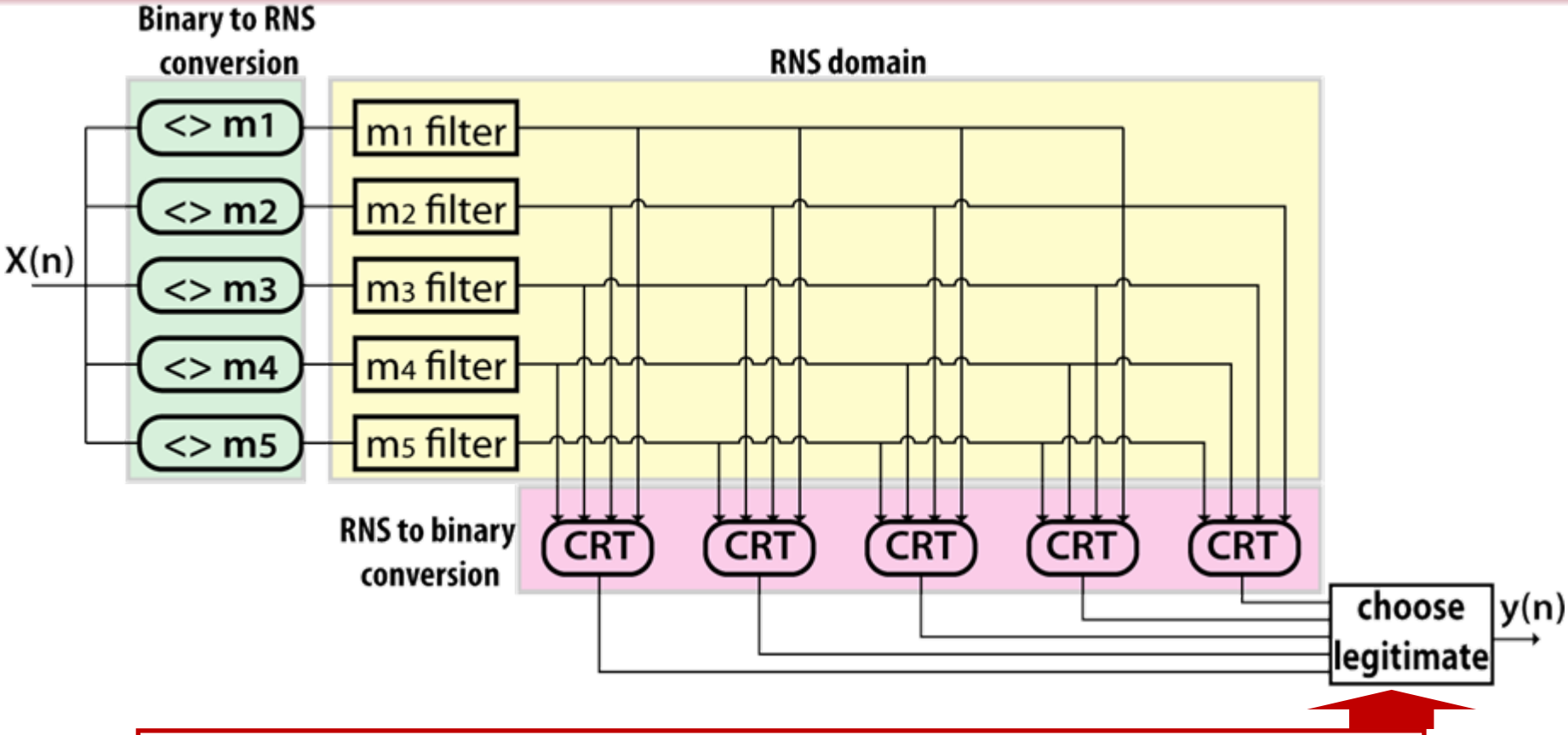
Error detection and localization

If an error affects the module ***i***
then
projection ***i*** falls in the *legitimate* range
and the others in the *illegitimate* range

Error correction

The correct output value can be obtained performing the reverse conversion of the m_i projection (X_{mi})

RRNS FIR filter

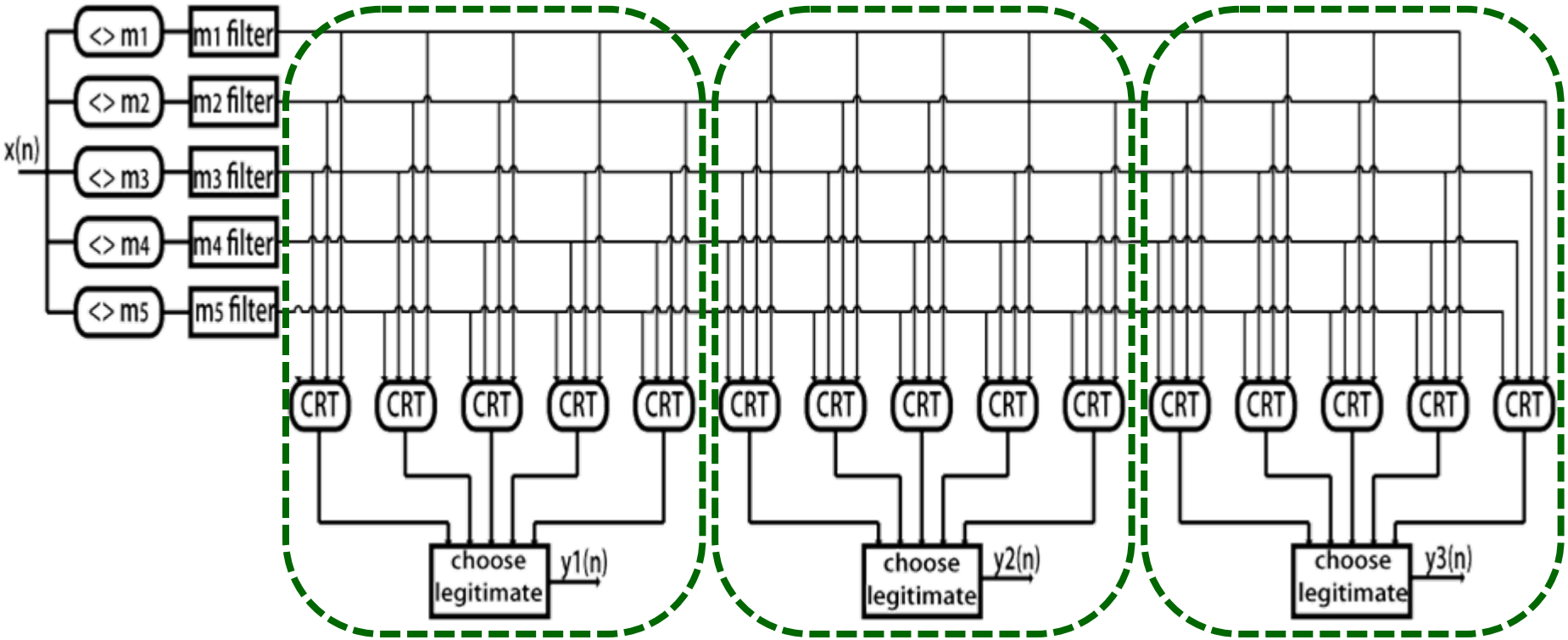


The error detection and correction block selects the legitimate value

This scheme does not cover faults in the CRT converters and in the “choose legitimate” block

TMR Hardened RRNS FIR filter

Possible solution: protection of the conversion blocks (CRTs) and the “choose legitimate” block with TMR

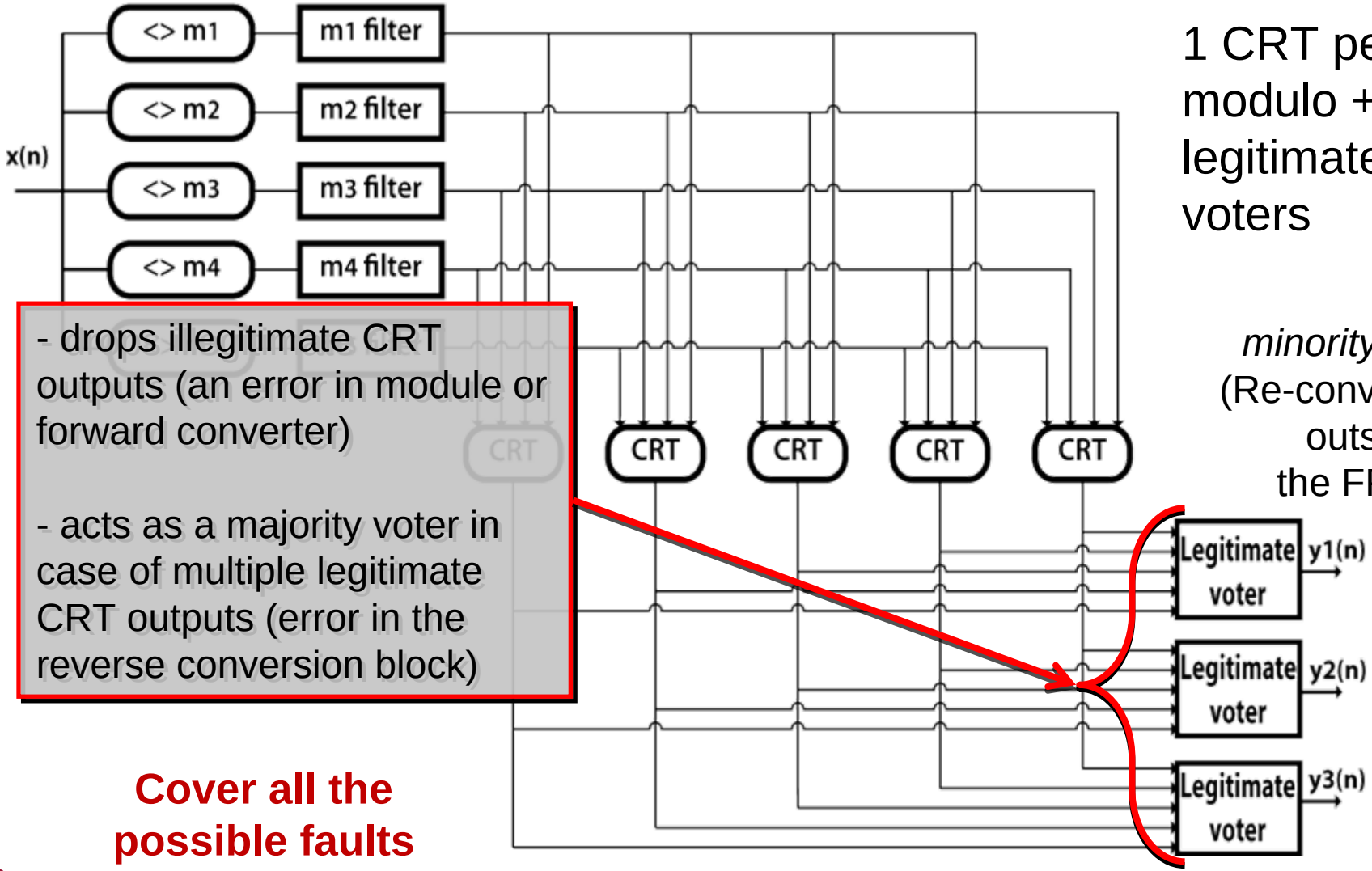


1 CRT per modulo X 3 + 3 “choose legitimate” blocks

Area occupation !!

Our RRNS hardened implementation

Our solution: create a block "***Legitimate voter***"



1 CRT per modulo + 3 legitimate voters

- drops illegitimate CRT outputs (an error in module or forward converter)

- acts as a majority voter in case of multiple legitimate CRT outputs (error in the reverse conversion block)

minority voters (Re-convergence outside the FPGA)

Cover all the possible faults

Overhead comparison

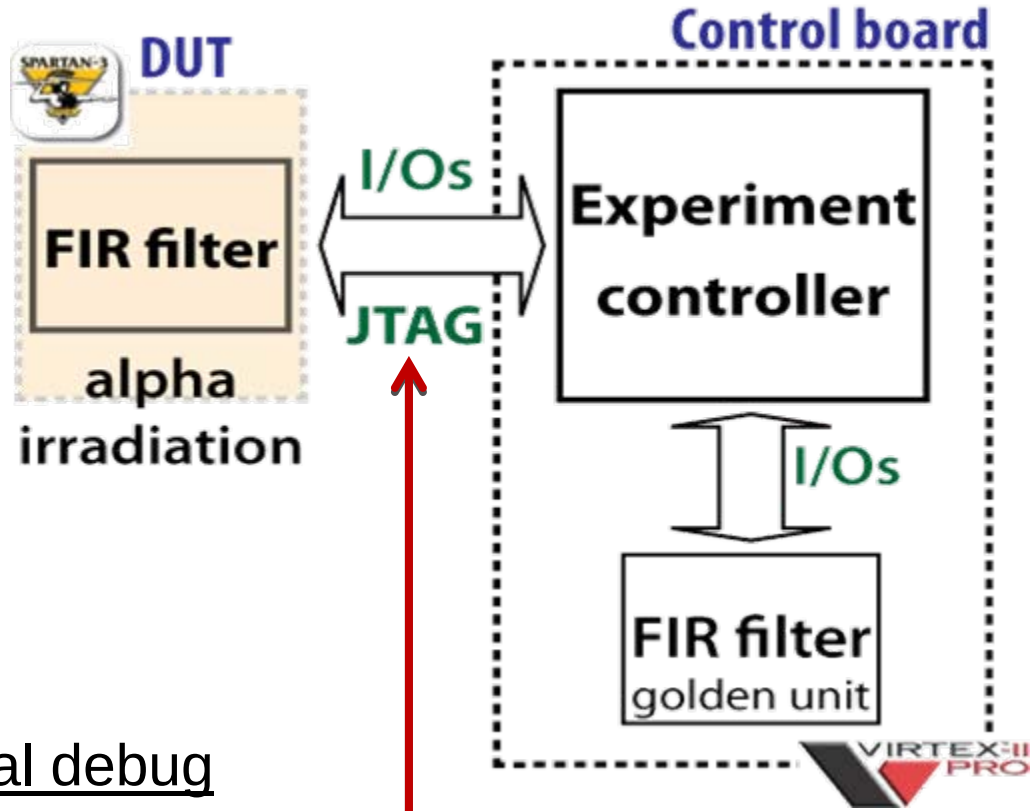
Area comparison:
TMR-RRNS vs. Our RRNS implementation

Filter	Number of taps	Dynamic range	TMR-RRNS Overhead [# of LUTs]	Our Implementation overhead [# of LUTs]	%
FIR1	16	20	7407	2931	40%
FIR2	64	22	9774	3763	39%
FIR3	256	24	17037	5780	34%
FIR4	16	28	17127	5927	35%
FIR5	64	30	17196	5951	35%
FIR6	256	32	19242	7044	37%

Irradiation Experiment Setup

DUT: hardened FIR filter implemented on a Xilinx Spartan-3 FPGA (XC3S200) irradiated with **alpha particles** (^{241}Am source)

Control board:
- provides the stimuli to the DUT filter
- compares the circuit behavior to the expected ones
- reads back/configures the DUT configuration memory (JTAG)



DUT filter I/Os plus additional debug signals to monitor the "Legitimate Voters" behavior and to localize the induced errors

Experimental Results & Discussion

- The DUT was irradiated until an illegitimate or two different legitimate values were detected at the “legitimate voter” inputs. After each event the DUT was fully reconfigured.

- We collected thousand of events and observed **no errors** at the filter outputs after the “legitimate voter”, which **worked properly in all the situations**

- We encountered some “strange events”: in some occasions the legitimate voter received erroneous input data, even if there were no errors in the configuration memory. The device recovered after minutes (after several full reconfigurations), possibly due to **half latch** problems [Graham et al. TNS Dec. 2003]

Experimental Results & Discussion

The added debug signals allow us to localize the induced faults

Error Locations	Events [%]
FIR module	27 %
CRT block	59 %
Legitimate voter	14 %

Faults classification:

- **FIR module**: error in the binary to RNS converter or in the i-th FIR module
- **CRT block**: error in the RNS to binary conversion block
- **Legitimate voter**: error in the “Legitimate voter” block

Conclusions

We presented a hardening-by-design technique based on the Redundant Residue Number System well suited for hardening FIR filters and DSPs, featuring an innovative “legitimate voter” with the following features:

- ❑ fault tolerance with respect to upsets in the configuration memory, as demonstrated by radiation tests

- ❑ lower FPGA resource usage as compared to a conventional hardening approach based on the triplication of the output conversion (CRT) and “choose legitimate” blocks