Fault Tolerant ICAP Controller for High-Reliable Internal Scrubbing

Jonathan Heiner, Nathan Collins, & Michael Wirthlin

This work was supported by Lockheed Martin under a grant from the University Projects program with collaboration by Tim Gallagher and Jon Wilson.

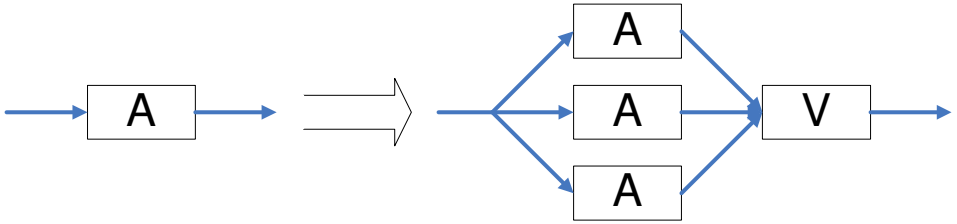
Outline

- FPGA Scrubbing Overview
- Internal Configuration Access Port (ICAP)
- Internal ICAP Architecture
- High Reliability Scrubber
- Radiation Test & Results
- Future Work & Summary

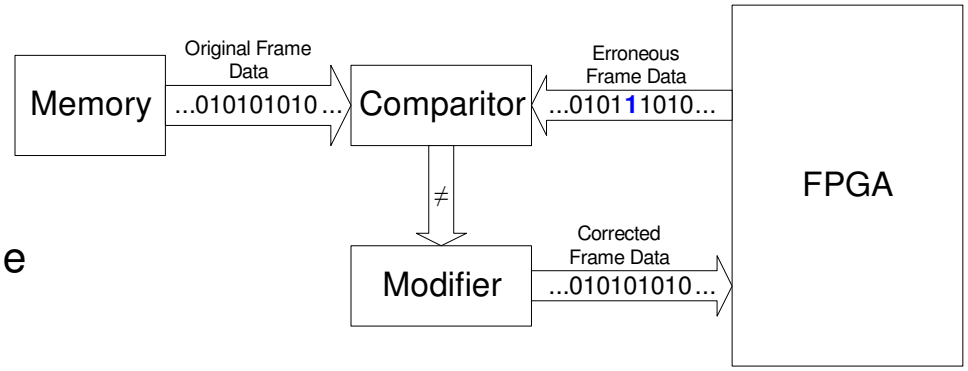
FPGA Fault Tolerant Strategy

- FPGAs provide SEU mitigation through redundancy and scrubbing

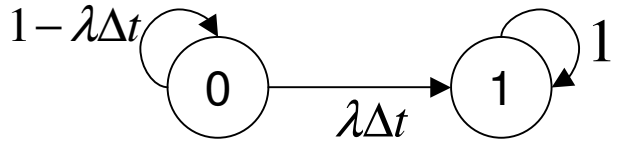
- Triple Modular Redundancy (TMR)
  - Triplicate module to introduce redundancy
  - Vote on outputs of triplicated module
  - Use greatest common result



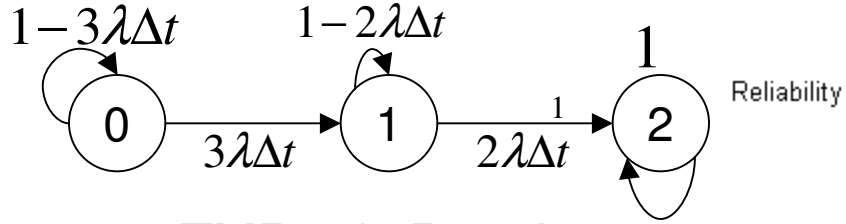
- Configuration Scrubbing
  - Readback frame data
  - Compare frame to original
  - Correct erroneous bits in frame
  - Writeback frame to FPGA



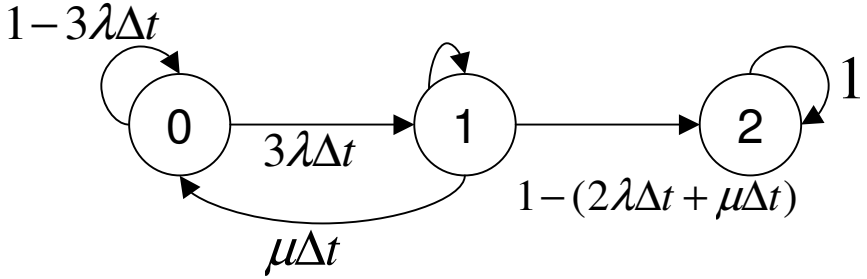
Continuous Time Reliability



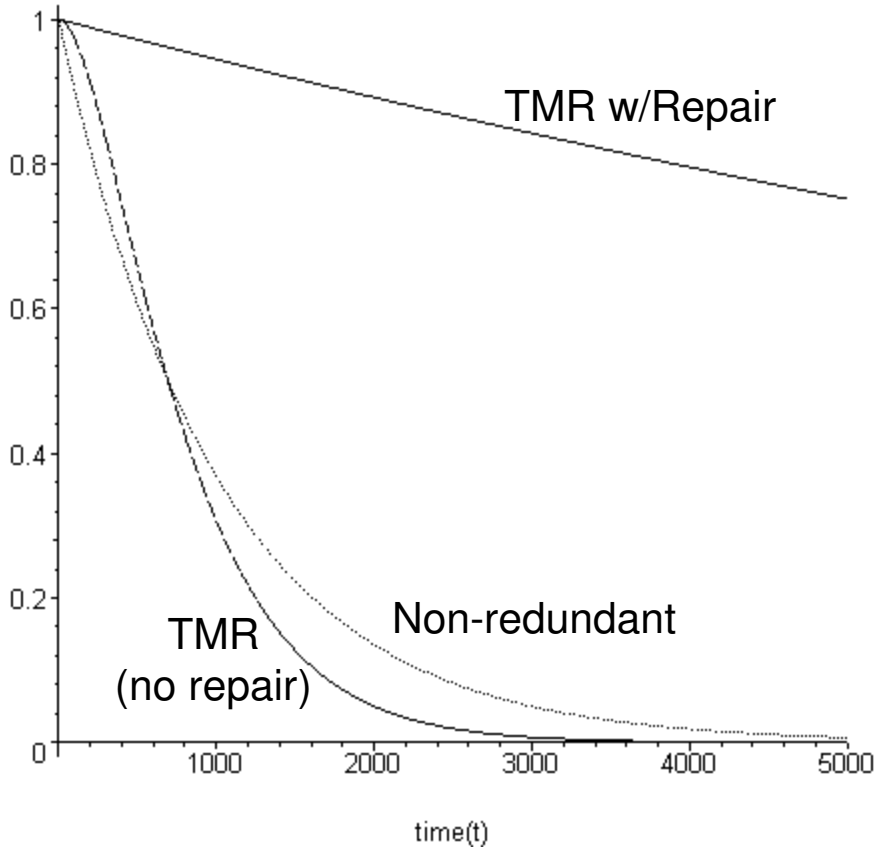
Non Redundant



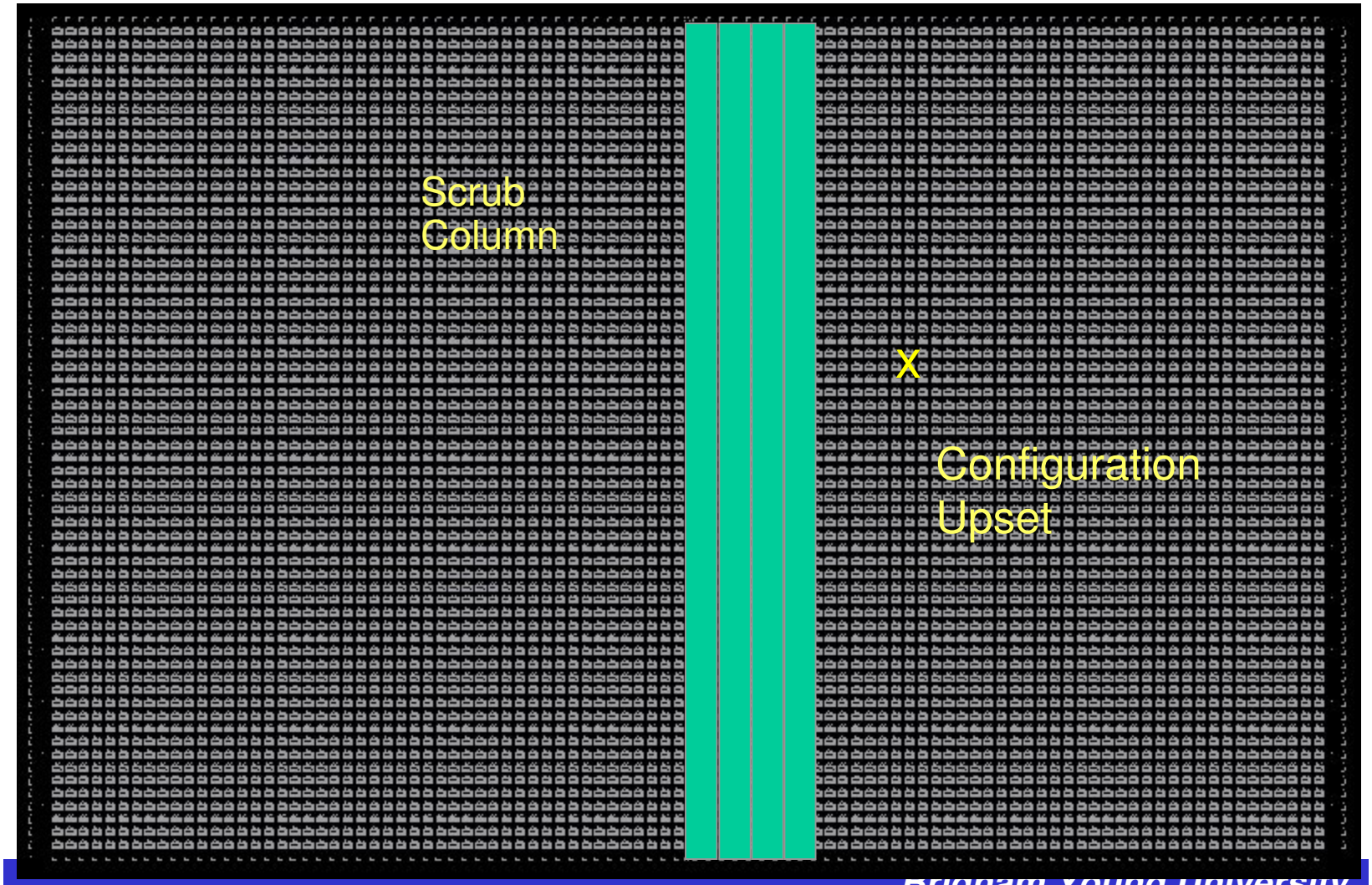
TMR w/o Repair



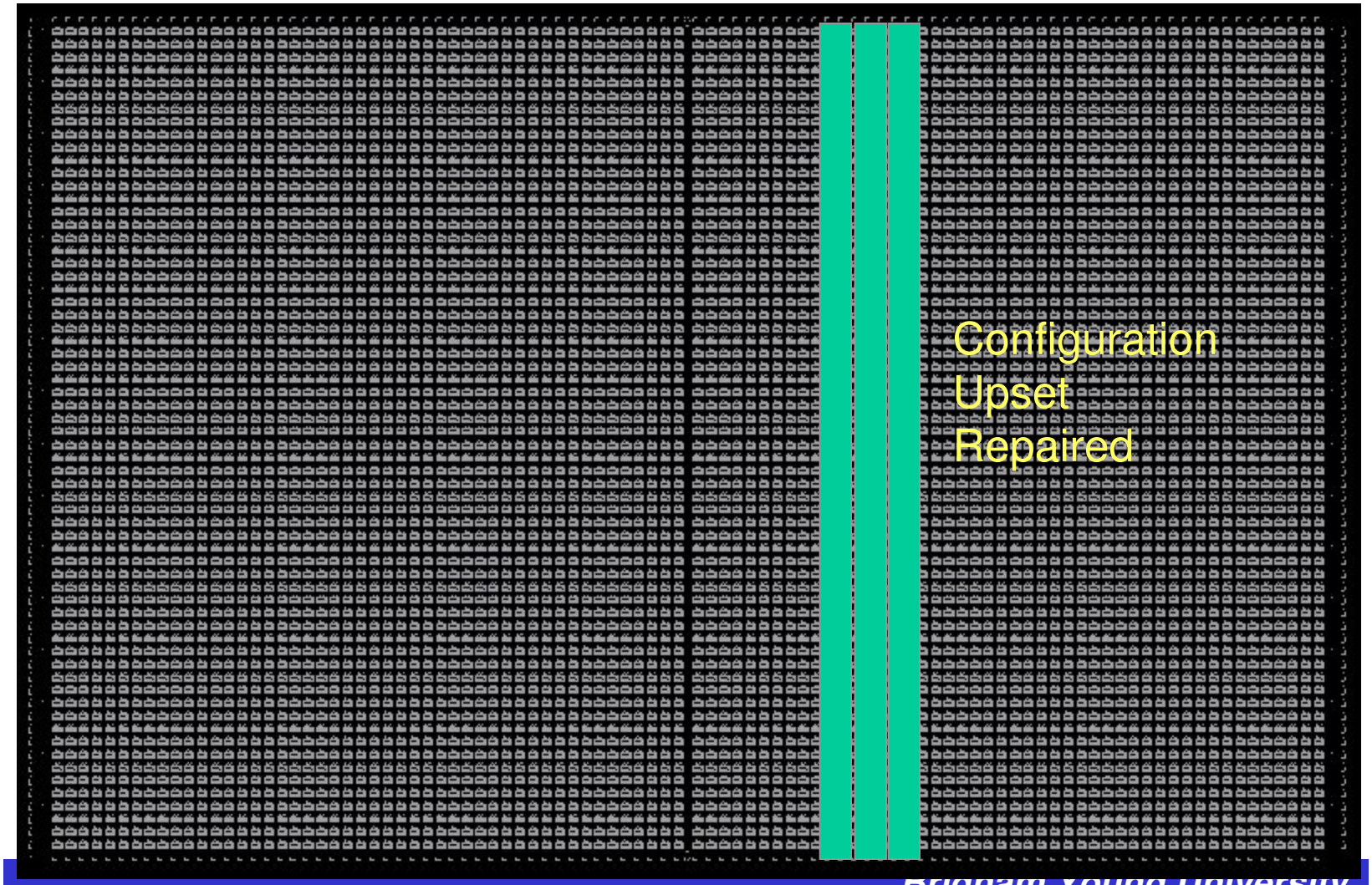
TMR w/Repair



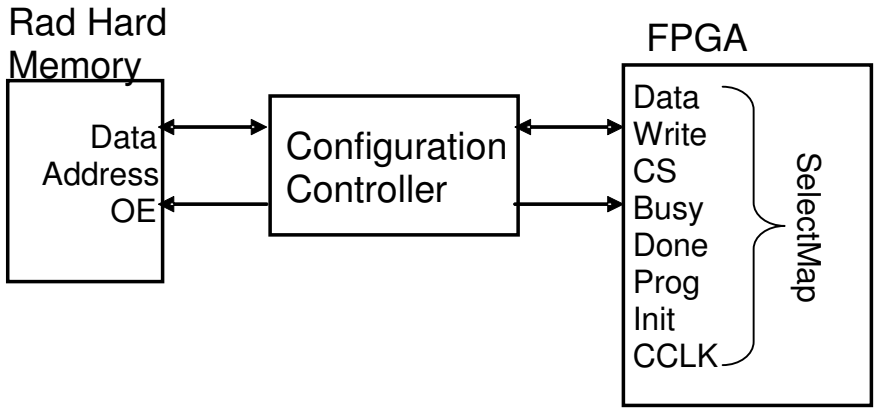
Configuration Scrubbing Example



Configuration Scrubbing Example



Traditional Scrubbing



- External Components
  - RadHard Memory
  - Configuration Controller
  - Dedicated IO
- External Scrubbers
  - Blind Scrubbing
  - Read-back Scrubbing

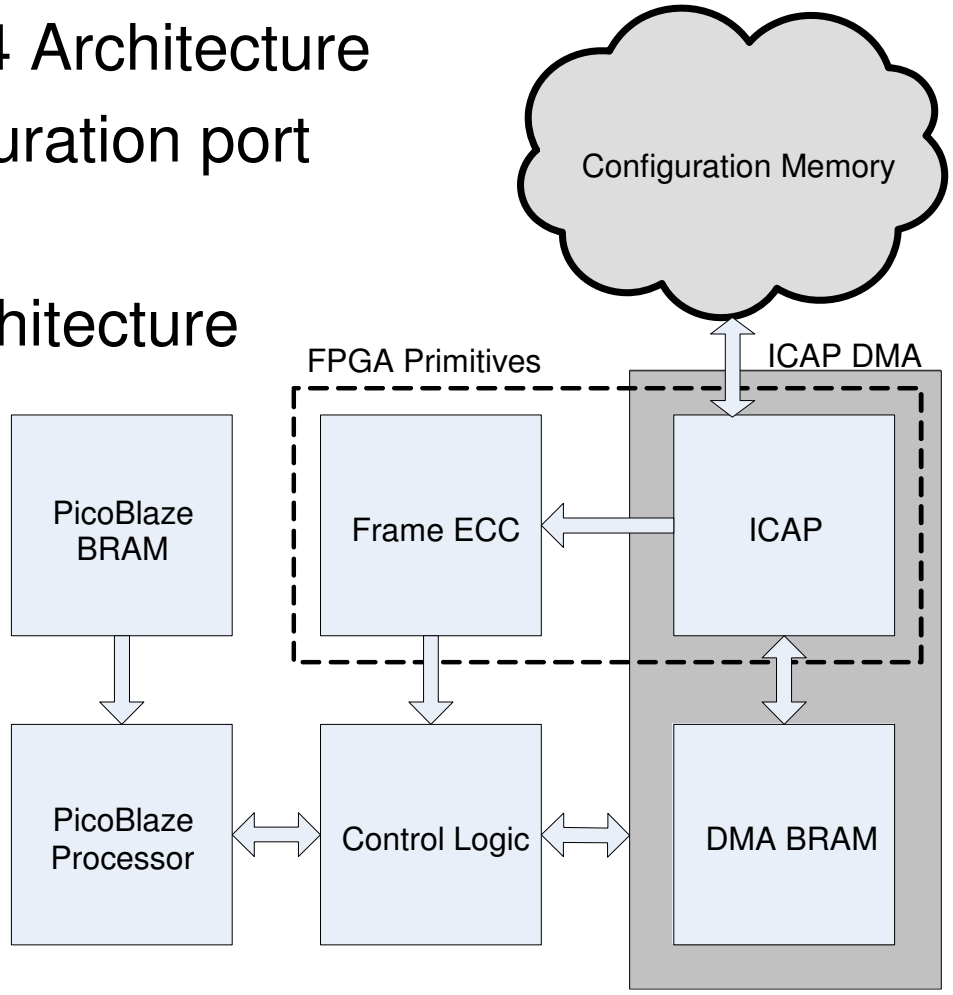
Traditional Scrubbing Process

- Read-back Scrubbing Process
  - Reads each frame sequentially
  - CRC or original frame comparison is performed on read frame for detection/correction
  - Corrected frame data is written back to configuration memory through SelectMap Interface
- Blind Scrubbing Process
  - Reads original frame data from memory
  - Writes frame to configuration memory through SelectMap Interface

Internal Scrubbing

- Described in XAPP714 Architecture
- Based on ICAP configuration port

- Internal Scrubbing Architecture



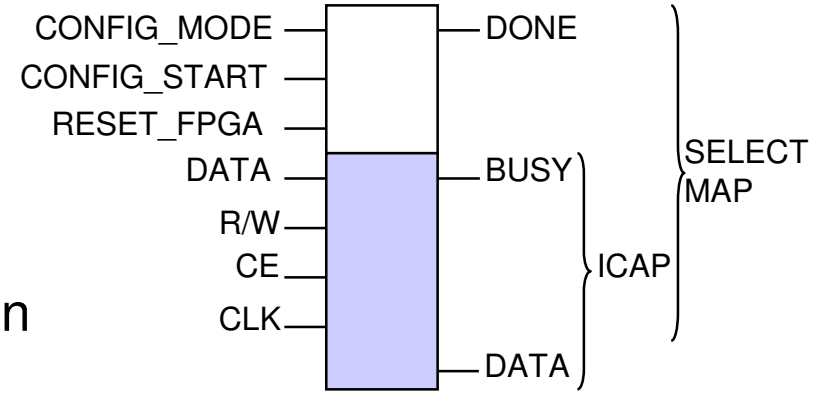
Internal Scrubbing Strategy

- Internal Scrubbing Process
  - Perform readback of each frame via ICAP interface
  - Use FrameECC to detect errors
  - Correct errors based on FrameECC syndrome value
  - Write corrected frame back via ICAP interface

- Advantages
  - No external memory, external controller, or external IO pins
- Disadvantages
  - Additional circuit area required for scrubbing circuit
  - Reliability of scrubber

Internal Configuration Access Port (ICAP)

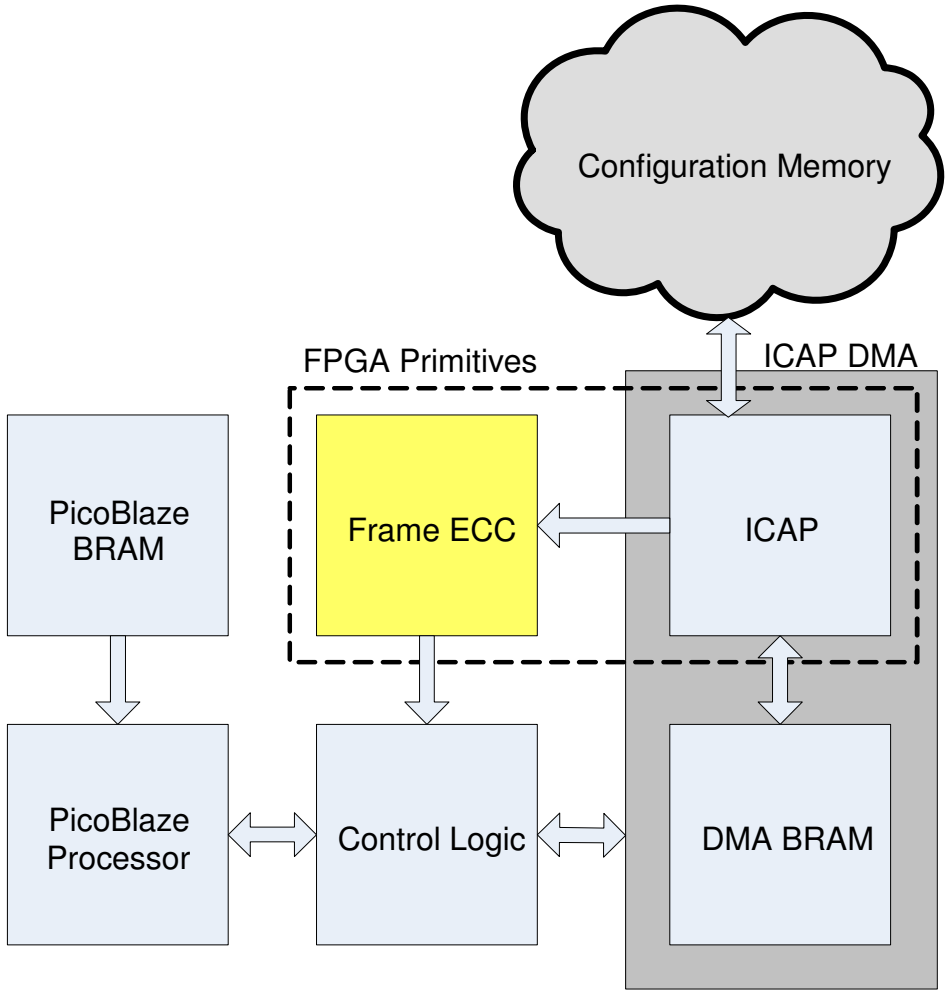
- Internal interface to configuration port
- Active readback and re-configuration
- Similar to SelectMap (separate I/Out data bus)
- Hard-wired Logic
- Current application usage
  - Dynamic Partial Reconfiguration
  - Encryption
  - Fault Tolerance/Injection



Detailed Architecture Overview

Frame ECC

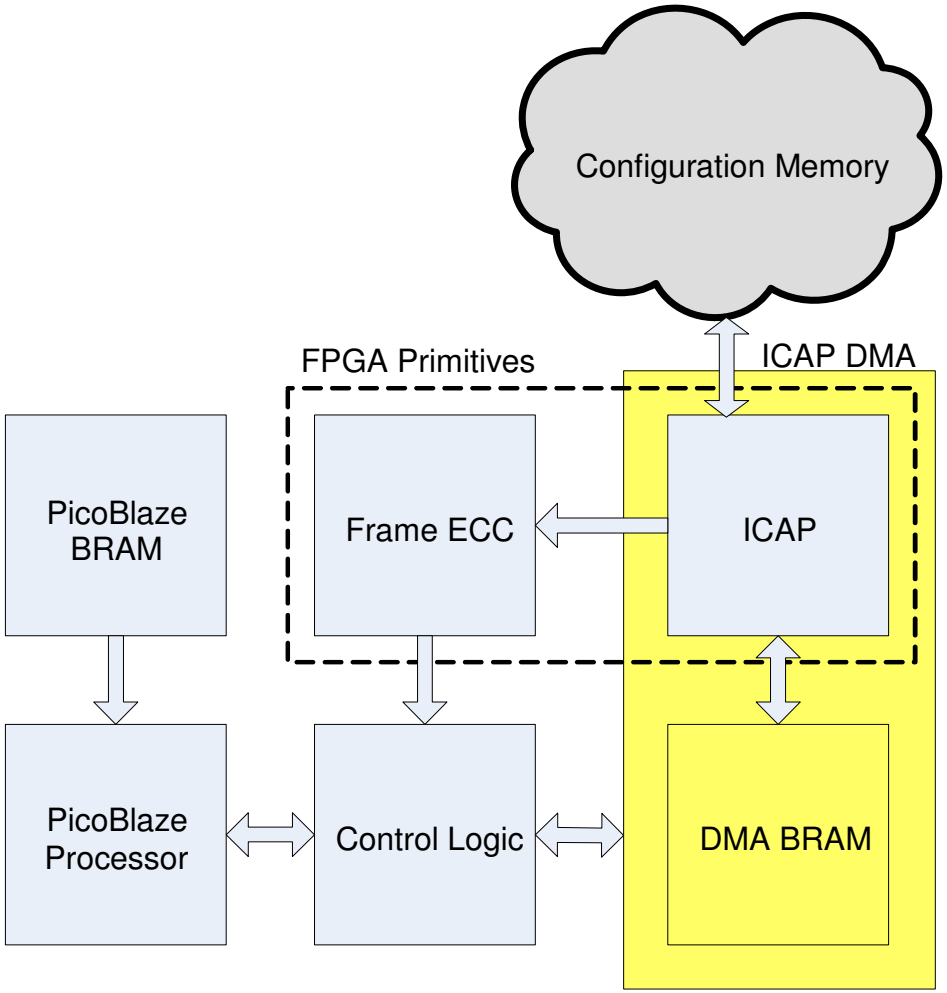
- Hard-wired internal component
- Performs SECDED algorithm on frame
- Provides syndrome word and error bit values
- Directly connected to read-port of ICAP



Detailed Architecture Overview

ICAP DMA

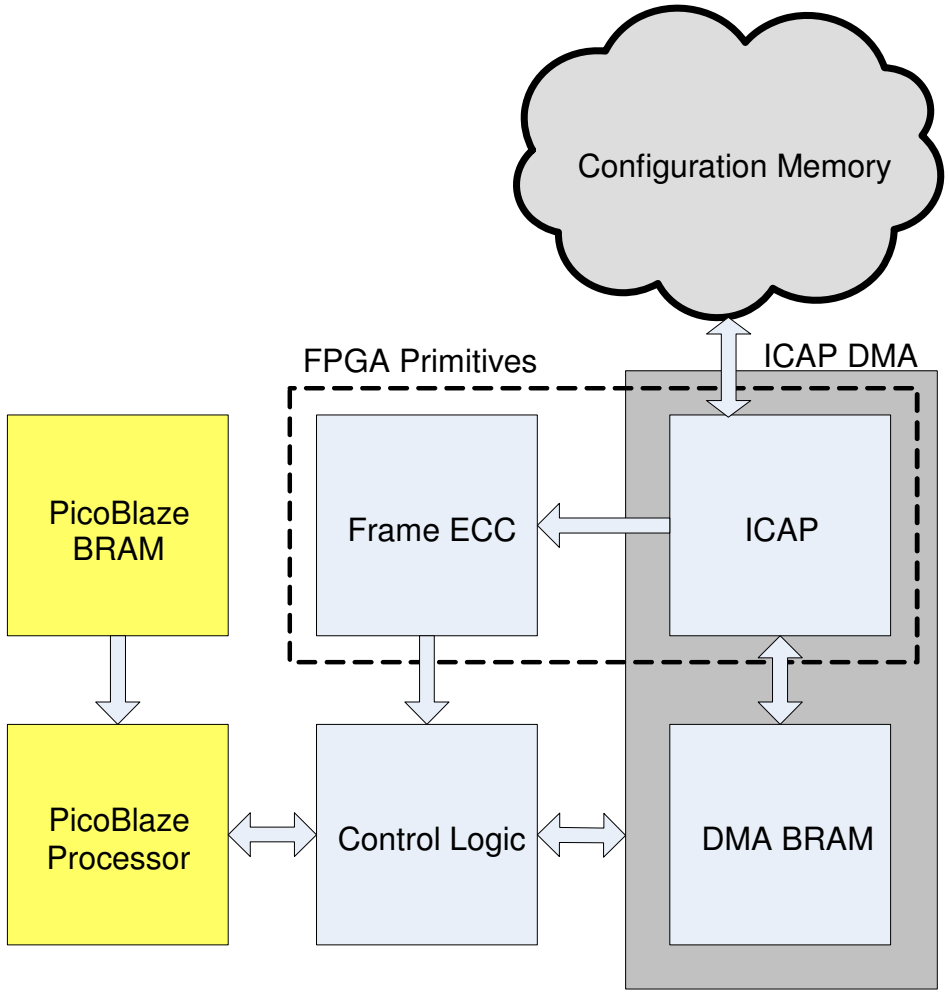
- Provides ICAP with data every clock cycle
- Stores ICAP output to DMA BRAM
- Transmits BRAM content to control logic



Detailed Architecture Overview

PicoBlaze Processor

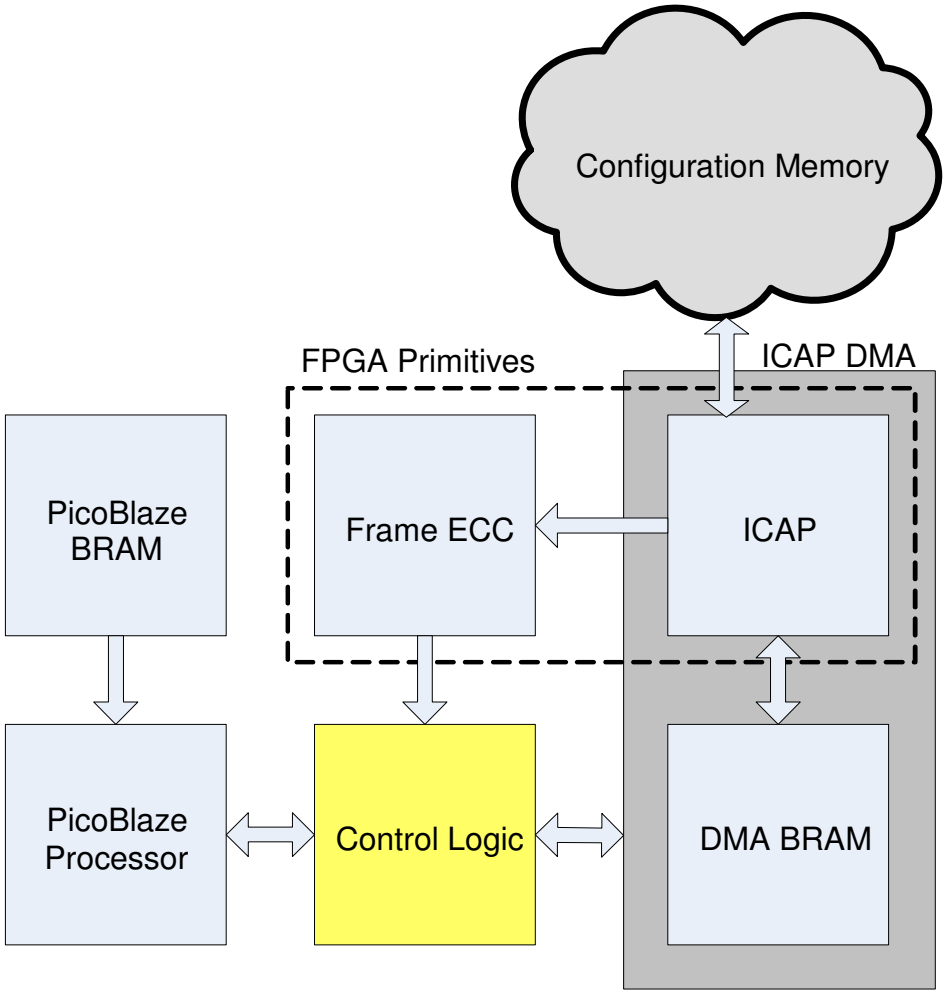
- 8-bit programmable µController
- Performs scrubbing logic
- BRAM contains pre-compiled scrubbing program
- Software used for ease of modifying logic



Detailed Architecture Overview

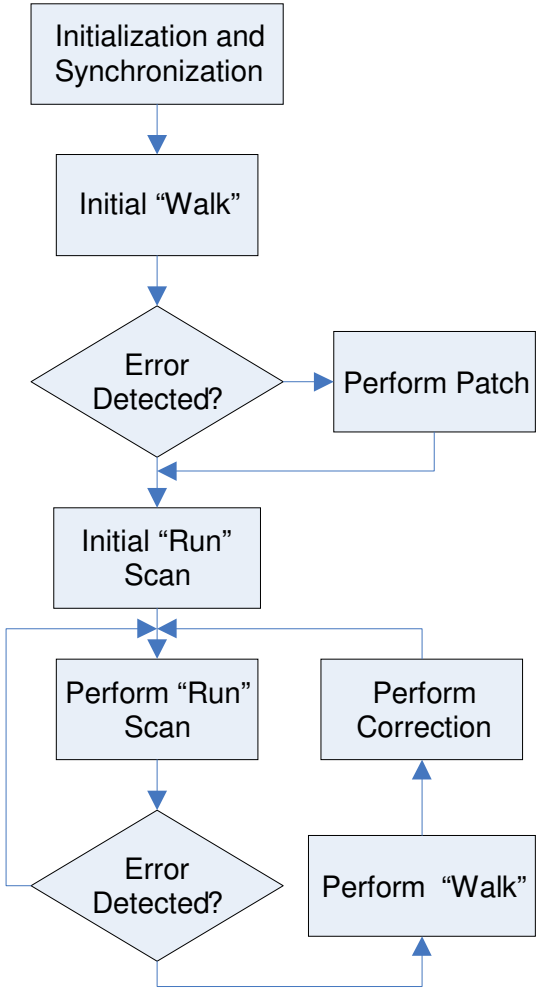
Control Logic

- Synchronize data transfer between picoblaze and ICAP DMA
- Maintains timing and data requirements



Scrubber Program

- Initializes devices
- “Walk” – slow scan
  - Approx. 24ms to 278ms @ 100Mhz
  - Actual Detection
- “Run” – fast scan
  - Approx. 1.2ms to 14.6ms @ 100Mhz
  - Quick Detection (is there an error somewhere)
- Patch – Ignore SEU by modifying parity bits
- Correction – Correct SEU



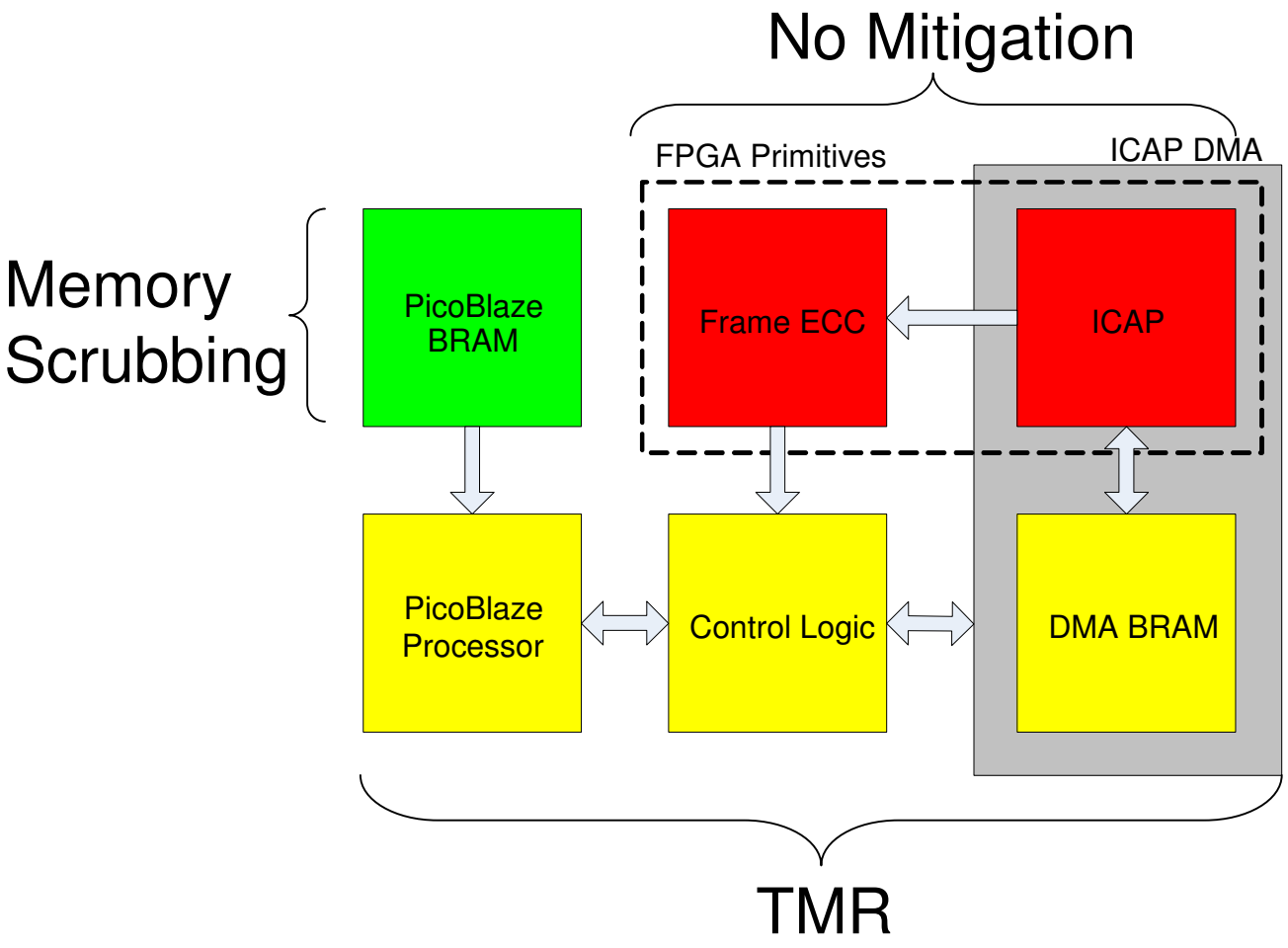
High Reliability Scrubber

Internal Scrubber is susceptible to configuration upsets

- Logic used to implement scrubber may be affected by SEUs
- Upsets within the scrubber logic may limit the ability of the scrubber to repair the fault

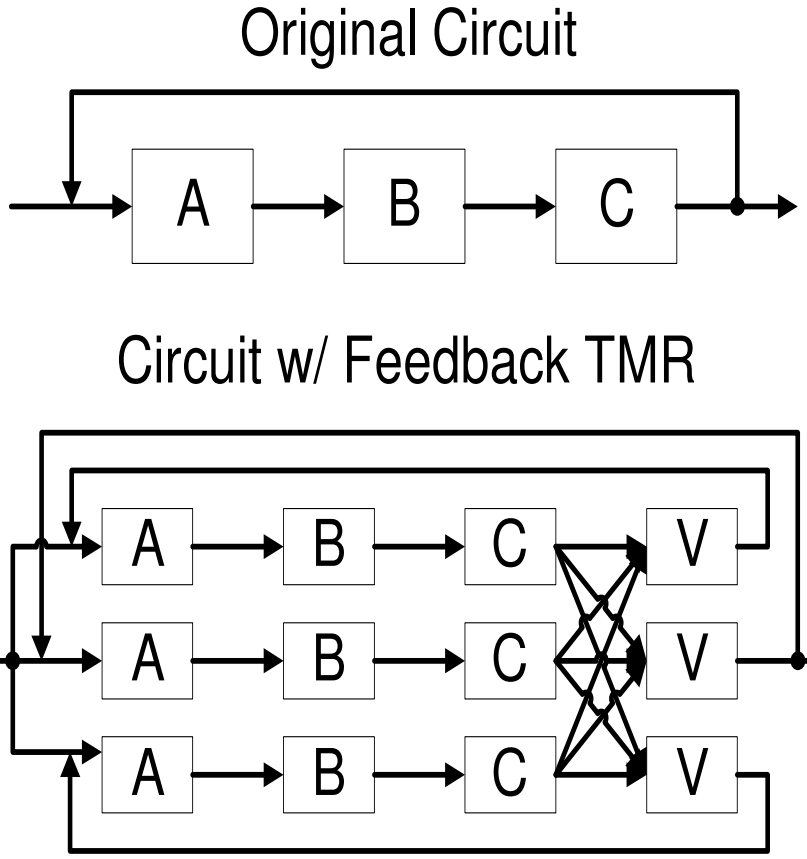
SEU mitigation technique needed to insure reliable scrubbing

High Reliable ICAP Scrubber



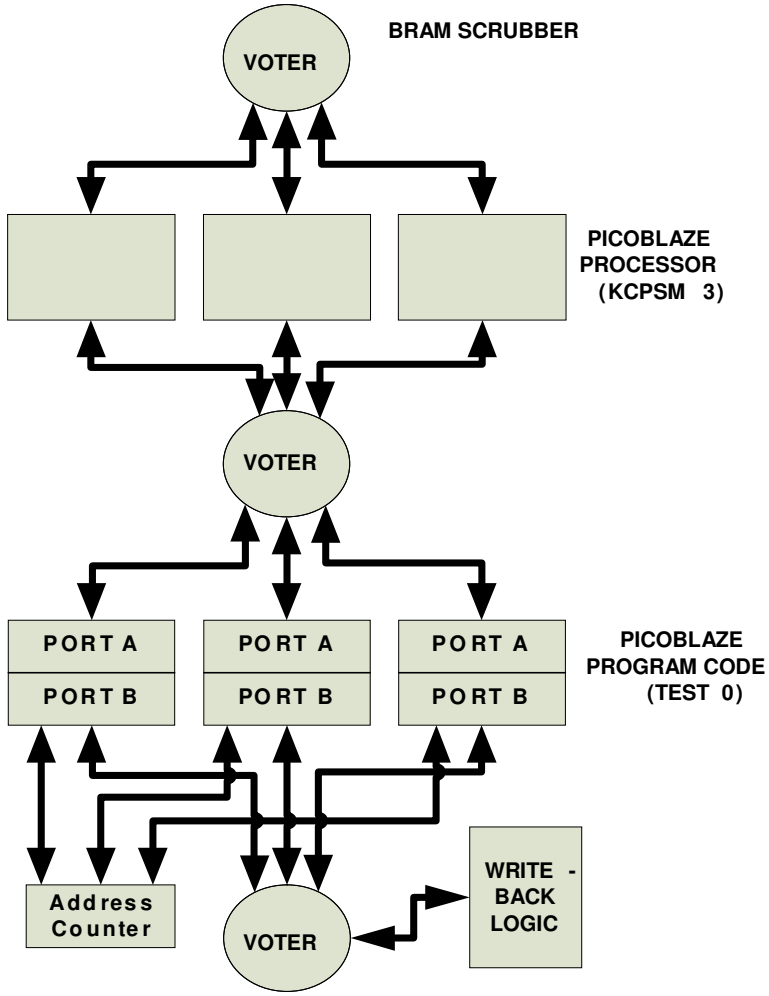
Triple Modular Redundancy (TMR)

- Mitigates all single bit upsets
- Allows scrubber to operate in presence of upsets
  - Scrubber will repair upset
- BL-TMR tool applied to circuit for selective mitigation



BRAM Scrubber

- Specialized BRAM scrubber for Picoblaze memory
- Continuously read and repair upsets within the memory



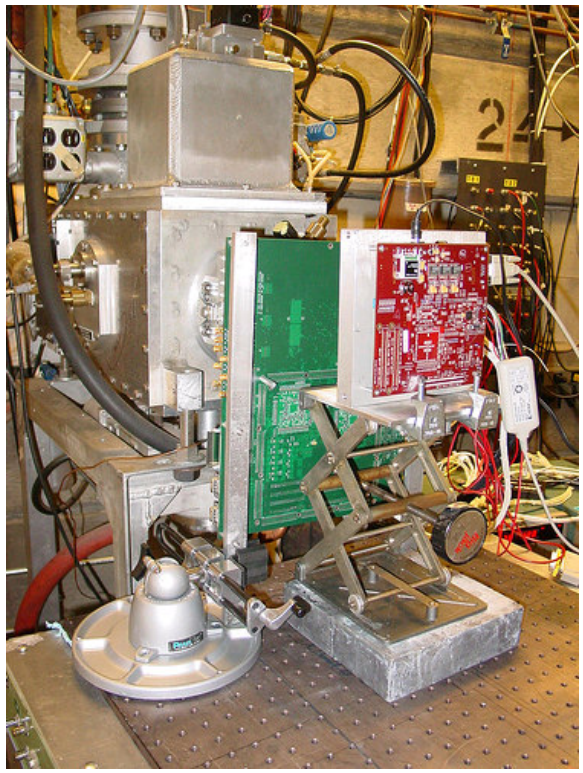
Scrubber Design Utilization

Resource	Non TMR	TMR
Flip Flops	680 (3%)	1082 (5%)
Slices	736 (6%)	1308 (12%)
BRAM	2	6

Virtex-4 LX-25

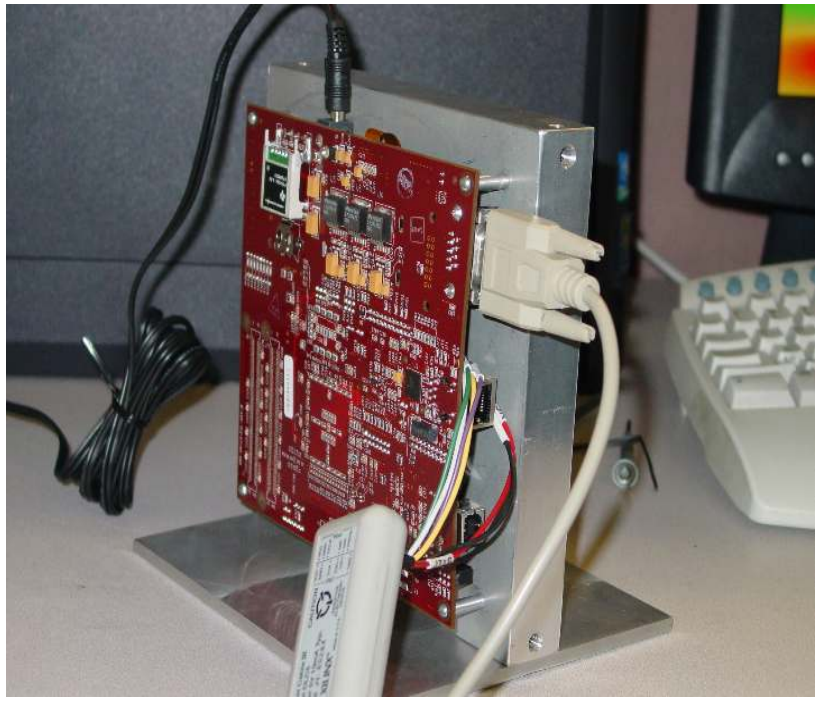
Radiation Test

- Determine the reliability of ICAP scrubber
  - Measure reliability of non-TMR scrubber
  - Measure reliability of TMR scrubber
- Test limitations
  - Operated behind another test
  - Did not have control over beam flux
  - Had to reconfigure with beam on



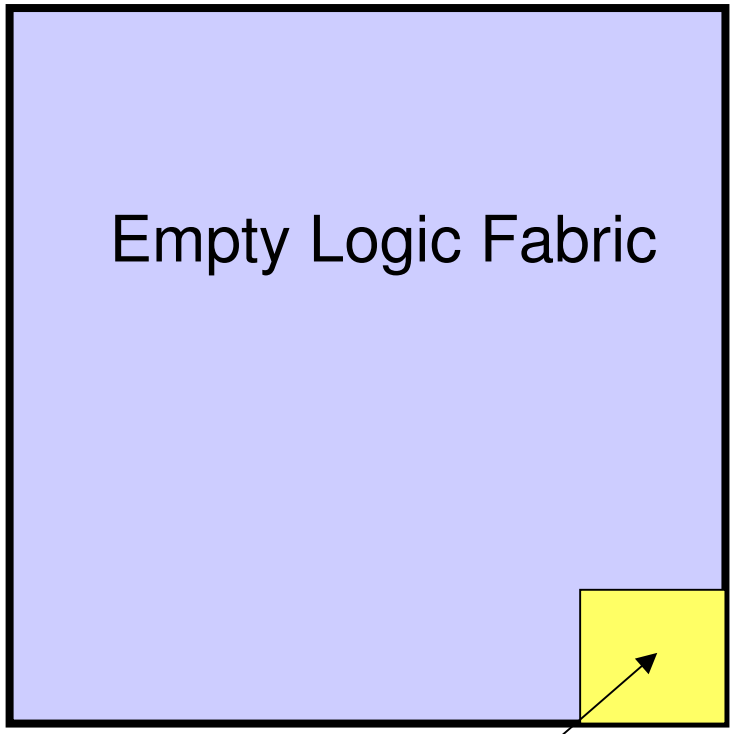
Radiation Test

- Board
  - Avnet Virtex-4 LX-25 Evaluation Board
  - 100Mhz Clock (50Mhz used)
  - RS232 port
- Shielding
  - 1” Aluminum Shield w/ 1”x1” Perforated Hole to expose FPGA
- Designs
  - Internal ICAP based Scrubber w/out TMR
  - Internal ICAP based Scrubber w/ TMR



Radiation Test Design

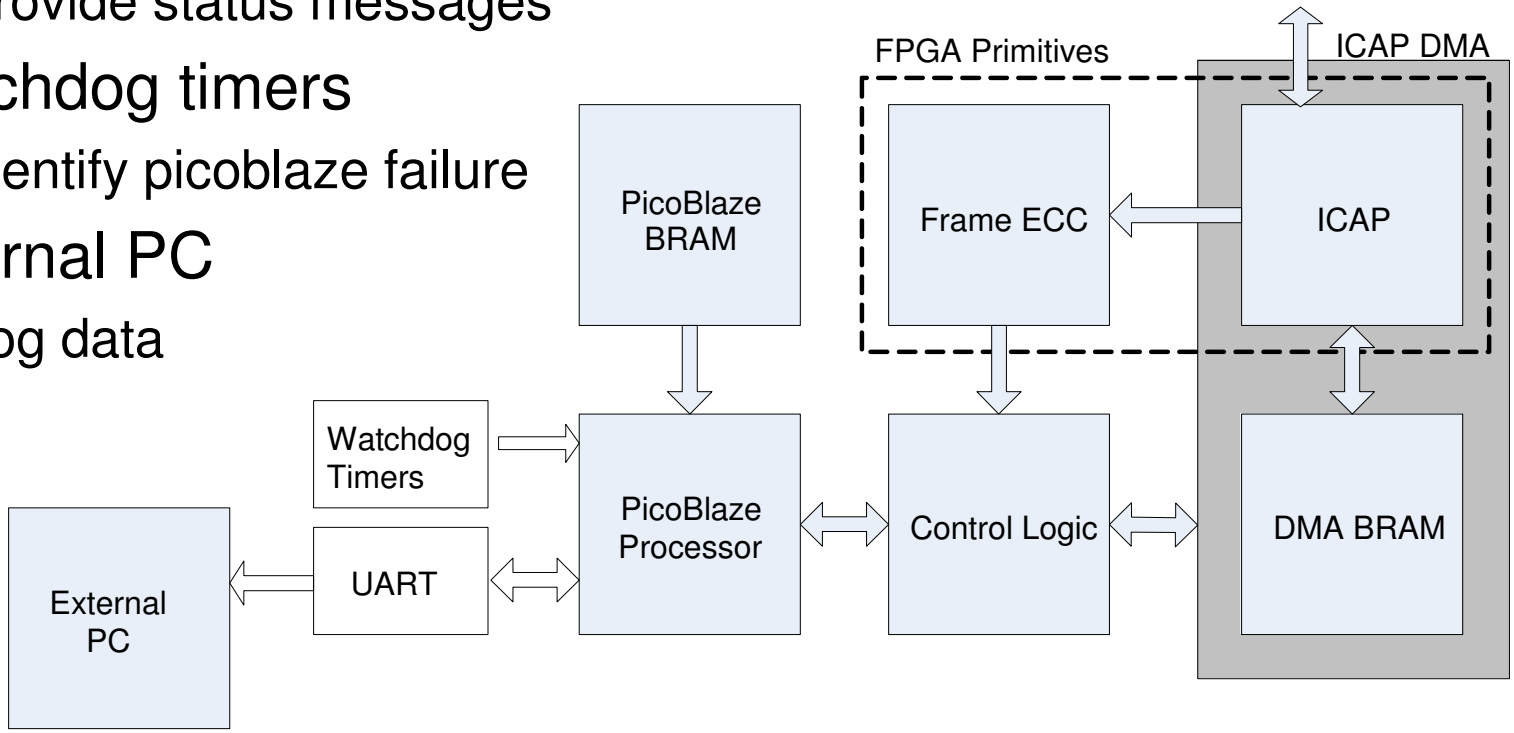
- ICAP controller
  - TMR design
  - Non TMR design
- No other FPGA circuitry
  - FPGA mostly empty
- Detect and repair upsets in all areas of FPGA
  - Unused logic
  - ICAP controller logic



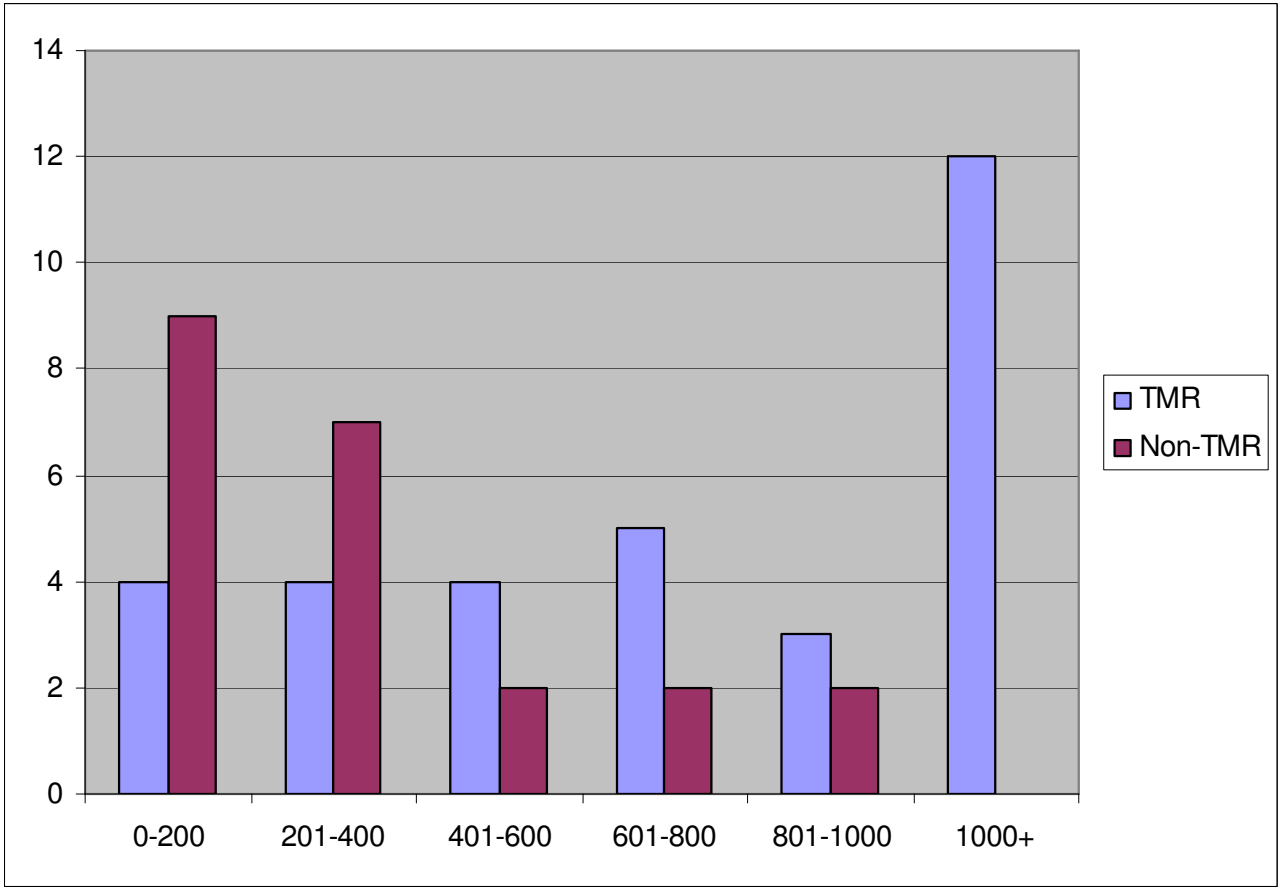
ICAP Controller
(both TMR and non TMR)

Data Collection and Monitoring

- UART
  - Transmit SEU data to PC
  - Provide status messages
- Watchdog timers
  - Identify picoblaze failure
- External PC
  - Log data



Configuration Upsets between Failure



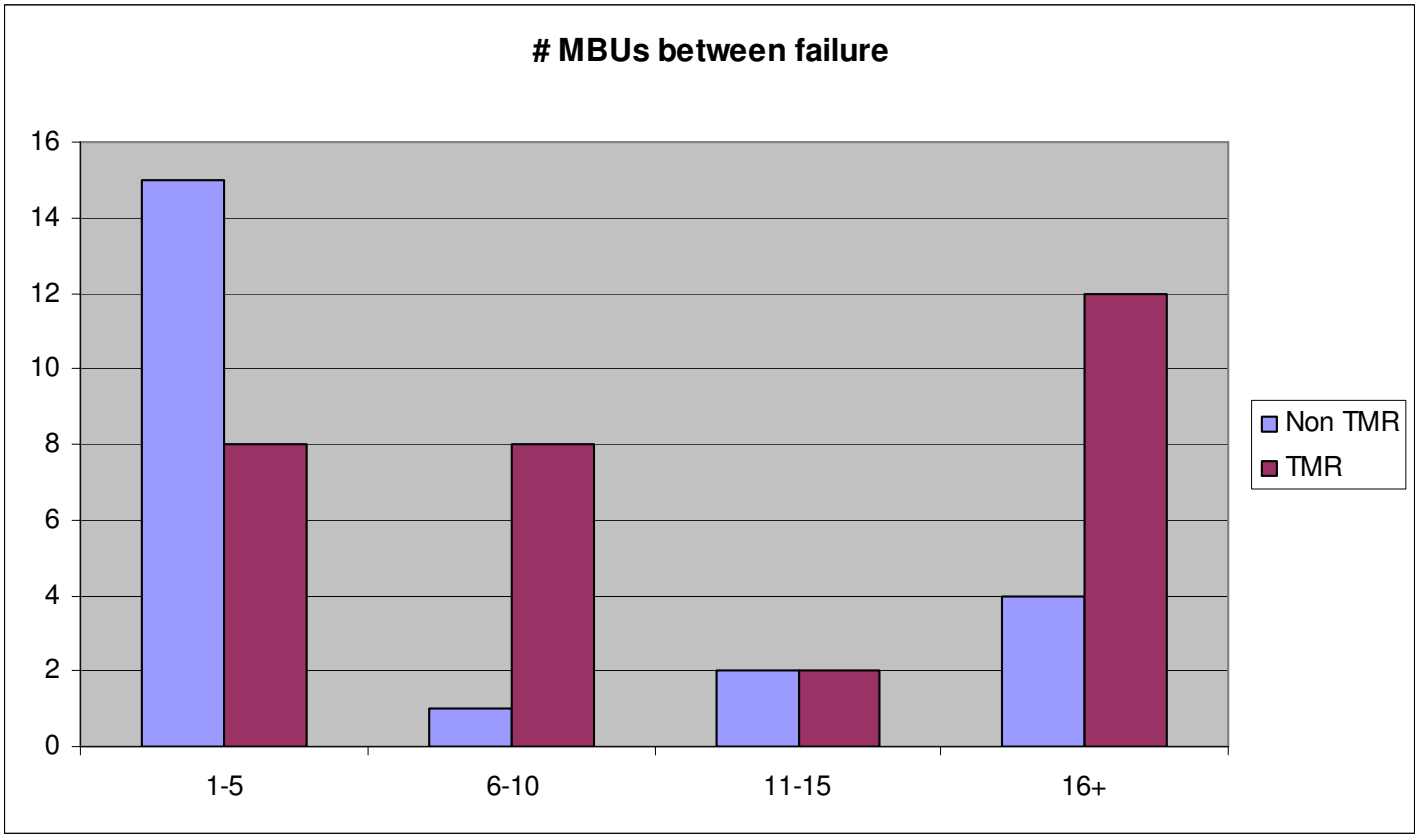
TMR: 1682 SEUs between failure

Non-TMR: 309 SEUs between failure

Multiple Bit Upsets

- FRAME ECC does not identify location of failure with multiple upsets within frame
  - Single Error Correction, Double Error Detection
  - Syndrome can not locate failures
- MBUs were detected but could not be corrected
  - MBUs accumulated during the test
  - Failures often occurred due to MBU accumulation
- Presence of MBU significantly slowed down scrubbing
  - Performed configuration “walk” with MBU
- 1.7% of upsets were intra-frame MBUs

Multiple Bit Upsets Between Failure



TMR: 10.4 MBUs between failure

Non-TMR: 7.6 MBUs between failure

Failure Modes

- Single Point Failures (were not isolated during test)
  - UART I/O
  - ICAP
  - Frame ECC
- Failure Modes (isolated during test)
  - Program crash
  - Invalid response from UART
  - Repeat FAR & syndrome values
  - Repeat FAR but different syndrome values
  - Repeat sets of FAR & syndrome values
  - FAR increments till end of FPGA row
  - Errors detected after test finished
  - Failed during reconfiguration

Conclusions

- ICAP scrubber worked correctly as expected
  - Detected upsets within FPGA fabric during operation
  - Repaired SEUs within the device
- Hi-Rel scrubber provided improved reliability
  - 5.4x higher SEU to failure than non-TMR
  - 1.4x higher MBU to failure than non-TMR
- ICAP hi-rel scrubber reliability limited by MBUs
  - Cannot remove MBUs
  - Failure due to accumulation of MBUs

Future Work

- MBU Detection & Correction
  - Investigate techniques for MBU correction
- VHDL Scrubber
  - Increased speed & possibly smaller circuit
- Dynamic Partial Reconfiguration

- Future uses of ICAP
  - Dynamic Partial Reconfiguration (bitstream compression)
  - Low cost Fault Injection