A Network-on-Chip for Radiation Tolerant, Multi-core FPGA Systems

Justin Hogan

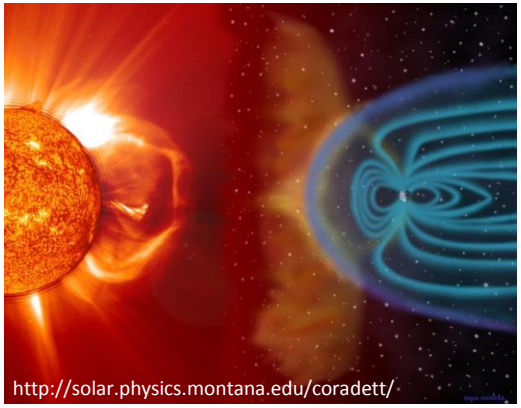
March 3, 2014

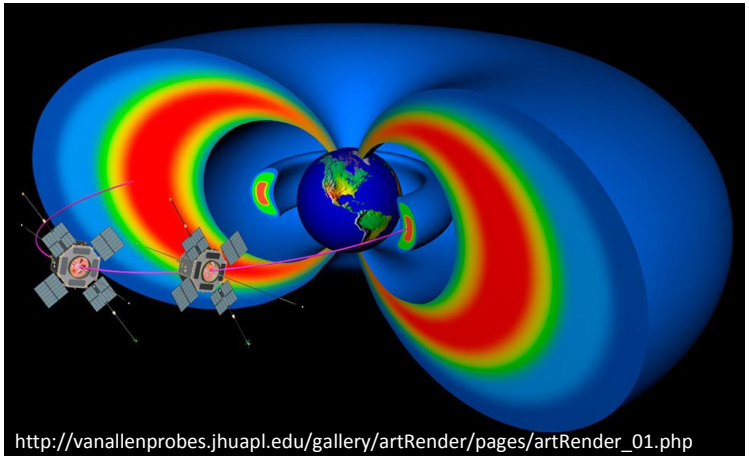
Presentation Overview

- Space radiation environment
- Radiation effects on electronics
- FPGAs in space
- Our research
- Advantages and Disadvantages
- Applications

Space Radiation Environment

- Ionizing radiation
  - Energetic electrons/protons
  - Heavy ions
- Radiation sources
  - Solar flares
  - Coronal mass ejection
  - Other cosmic sources
  - Magnetic field trapping

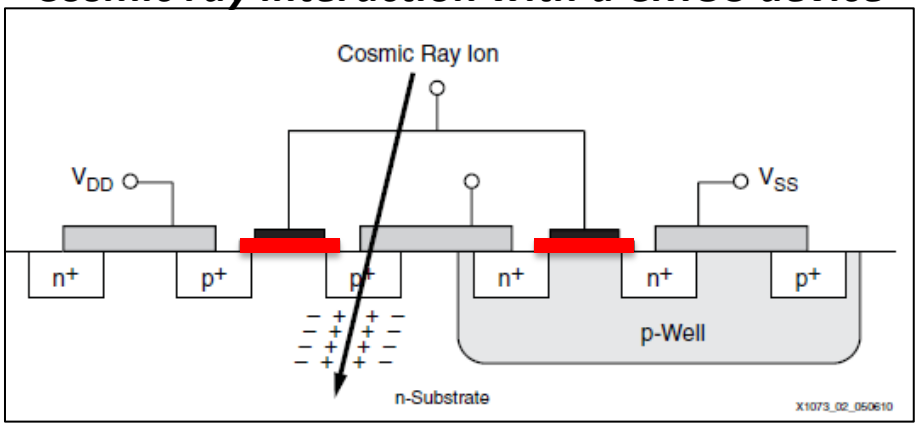




Radiation Effects on Electronics

- Cumulative
  - Total Ionizing Dose (TID)
    - Gradual degradation of transistor performance
    - Eventual device failure
- Transient
  - Single Event Effects (SEEs)
    - Single event transient (SET)
    - Single event upset (SEU)
    - Single event functional interrupt (SEFI)

Cosmic ray interaction with a CMOS device



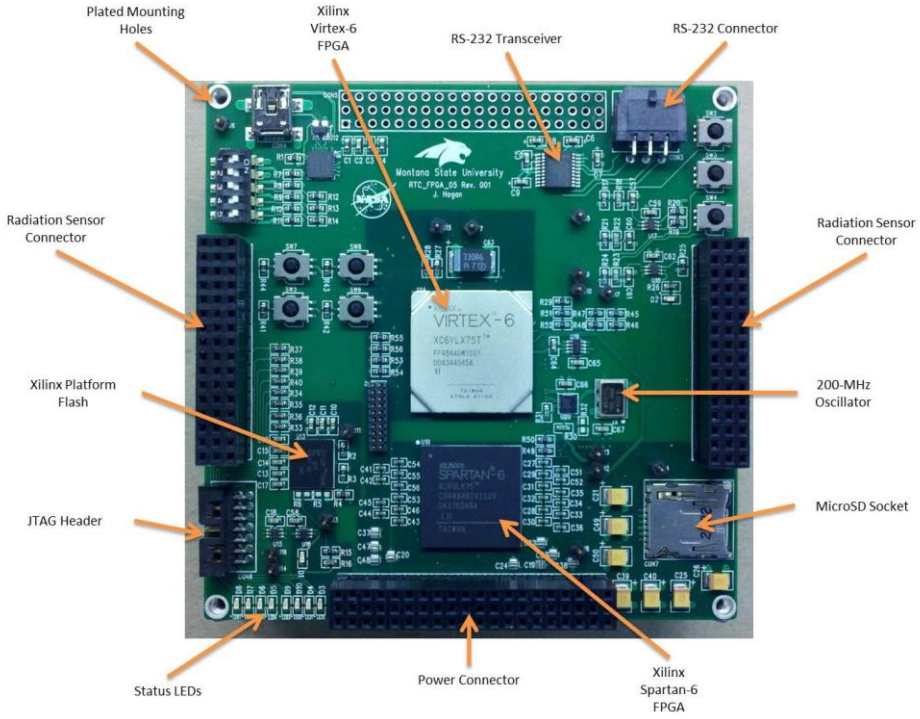
Xilinx Application Note XAPP1073 (v1.0)

Trends in Device Scaling

- With advancements in manufacturing processes…
  - TID tends to decrease
    - Thinner gate oxides trap less charge
  - SEEs tend to increase
    - Lower threshold voltages increase susceptibility

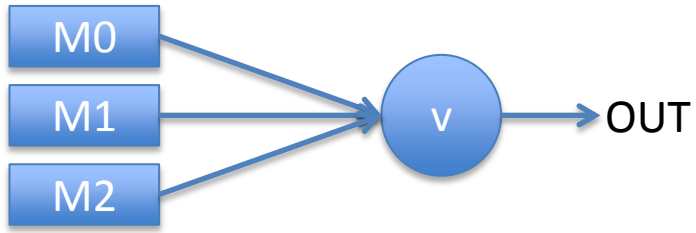
FPGAs in Space

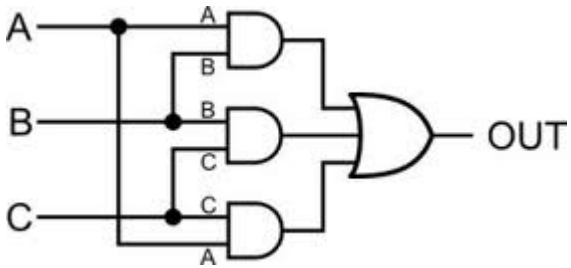
- SRAM FPGAs
  - Motivation for use
    - Reduced cost over rad-hard components
    - Increased computational efficiency
  - Main consideration is SEE mitigation



SEE Mitigation Techniques

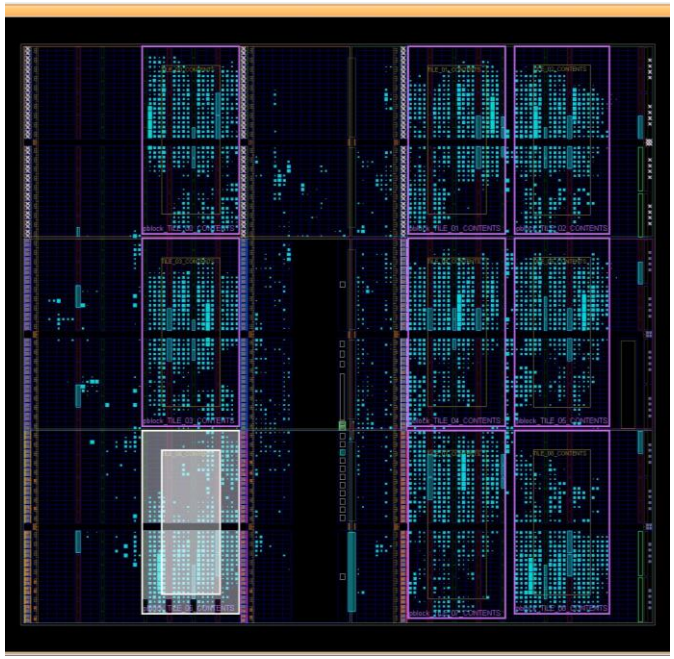
- Triple Modular Redundancy (TMR)
  - Mitigates single upsets
  - Increased resource utilization
- Scrubbing
  - Blind
  - Readback
- Process and Design
  - Rad-hard layout techniques
  - Silicon-on-Insulator





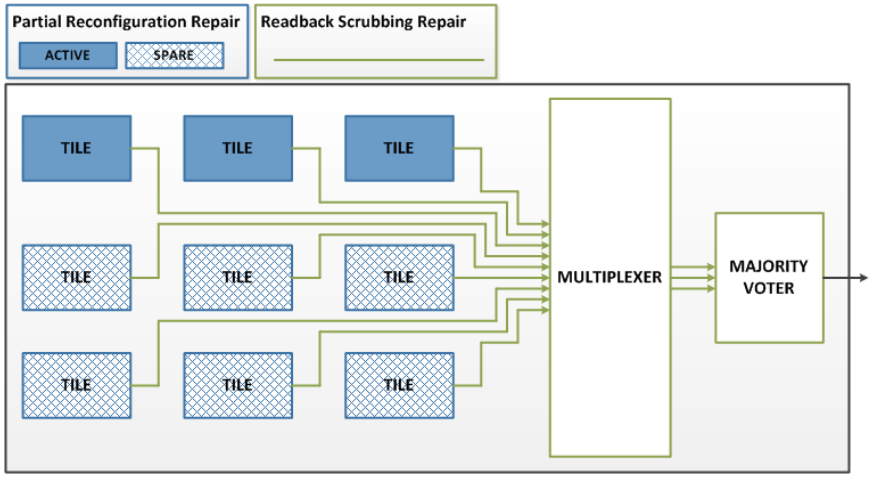
TMR+Spares

- Coarse-grain TMR
  - 9-tiles each containing a Microblaze
  - Partially reconfigurable regions
  - 3-active tiles, 6-spare tiles
- Fault detection
  - Feedback from voter
  - Developing readback
- Fault mitigation
  - TMR
- Fault recovery
  - Swap faulted tile for a spare
  - Repair faulted tile using active partial reconfiguration



Interconnect Faults

- Tile faults corrected through swap/reconfiguration
- Routing faults aliased as tile faults
  - Tile swapped while fault persists in routing
  - Tile swap is unnecessary



- **GOAL: Distinguish routing faults from tile faults to optimize the recovery response**

Solution

- Encode/Decode data at source and destination
- Error correction/detection codes (EDAC)
  - Tile swapped while fault persists in routing
  - Tile swap is unnecessary
- **GOAL: Distinguish routing faults from tile faults to optimize the recovery response**

Hamming Code Generation

Data = [d31:d0]

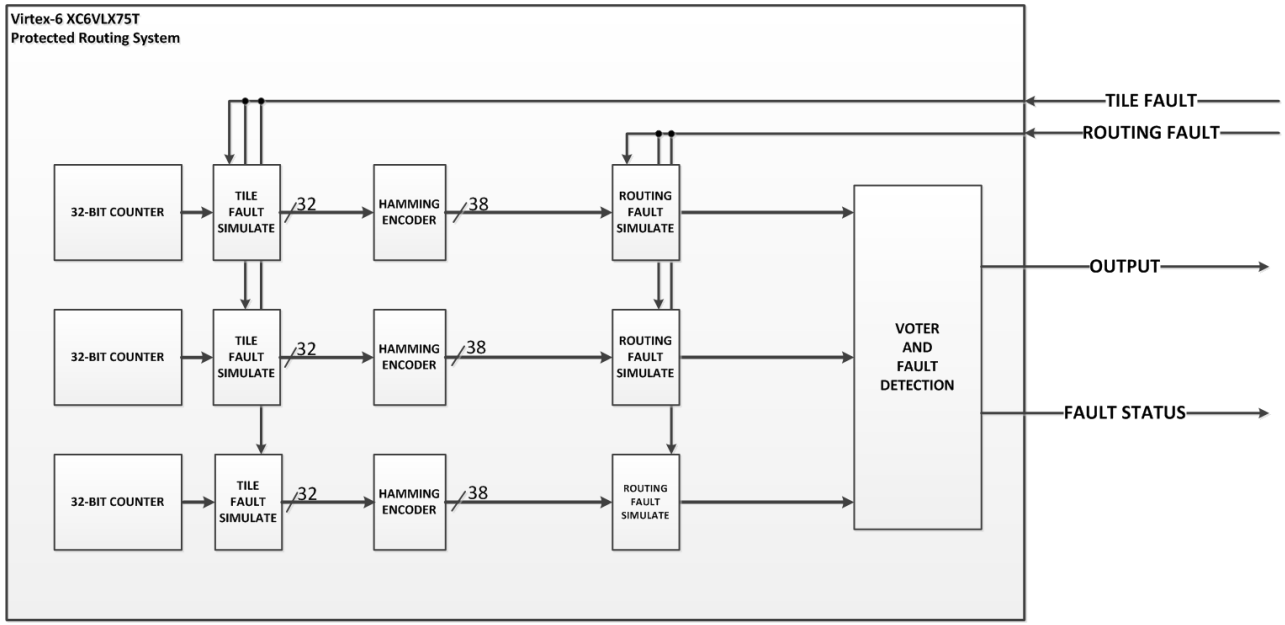
→ Hamming = [h5:h0], where:

h0 = XOR { data(0,1,3,4,6,8,10,11,13,15,17,19,21,23,25,26,28,30) }
h1 = XOR { data(0,2,3,5,6,9,10,12,13,16,17,20,21,24,25,27,28,30) }
h2 = XOR { data(1,2,3,7,8,9,10,14,15,16,17,22,23,24,25,29,30,31) }
h3 = XOR { data(4,5,6,7,8,9,10,18,19,20,21,22,23,24,25) }
h4 = XOR { data(11,12,13,14,15,16,17,18,19,20,21,22,23,24,25) }
h5 = XOR { data(26,27,28,29,30,31) }

Encoded Data = [e37:e0]

e31 = h5
e15 = h4
e7 = h3
e3 = h2
e1 = h1
e0 = h0
Data bits occupy remaining positions in descending order

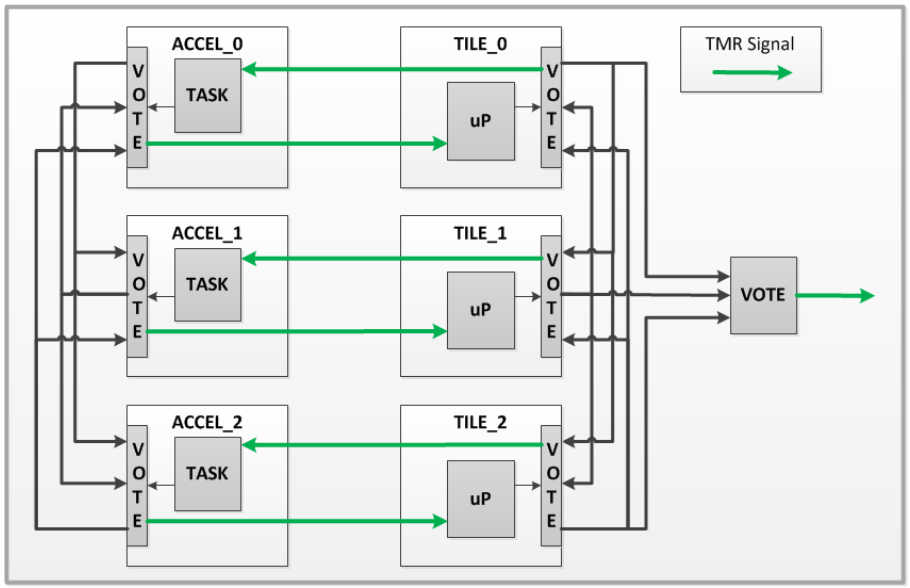
Simple Test System



- 32-bit counter generates output data
- Design-level fault injection simulates tile and/or routing faults
- Hamming encoder placed at the boundary of the tile
- Hamming decoder placed near the voter circuit
- Internal logic analyzer captures fault simulation, data recovery

Moving Forward

- This is an incremental step in increasing the performance of the system
- Transition from Microblazes to custom peripheral accelerator cores
- Minimize or eliminate single points of failure (e.g. voter)
- Implement a standardized interconnect allowing heterogeneous tiles



Questions