Improving FPGA Reliability in Harsh Environments using Triple Modular Redundancy with More Frequent Voting

Brian Pratt[†] , Michael Wirthlin[†]
Michael Caffrey[‡], Paul Graham[‡], Keith Morgan[‡], Heather Quinn[‡]
Severn Shelley [† ‡]

[†] Brigham Young University
[‡] Los Alamos National Laboratory

This work was supported by the Department of Energy at Los Alamos National Laboratory.
Approved for public release under LA-UR-07-7800; distribution is unlimited.

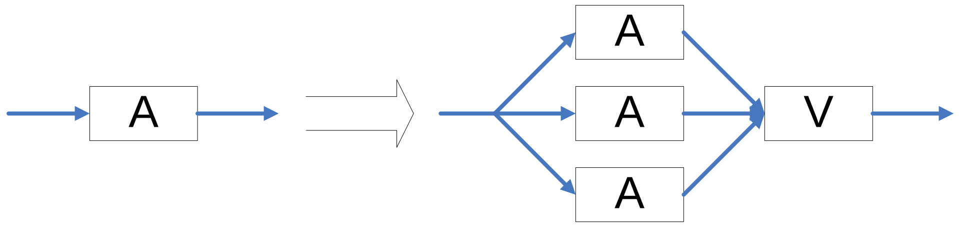
Outline

1. Triple Modular Redundancy
2. Reliability Models for TMR Systems
3. TMR with More Frequent Voting
4. Reliability Analysis of More Frequent Voting
5. Test Designs and Methodology
6. Simulation Results
7. Conclusions

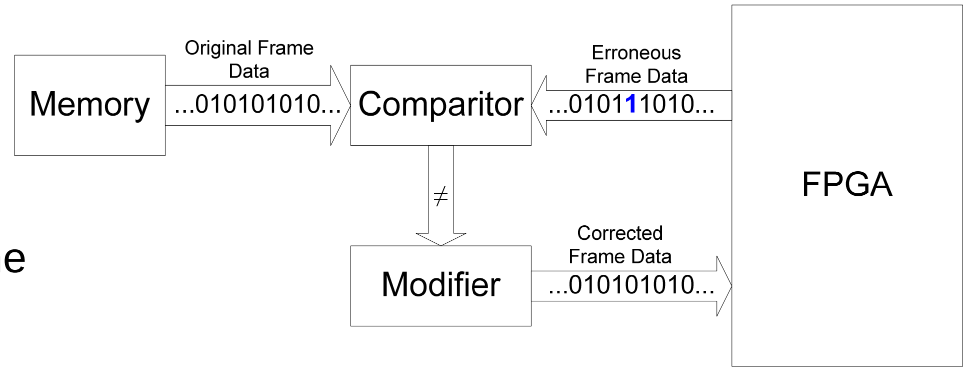
FPGA Fault Tolerant Strategy

- 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

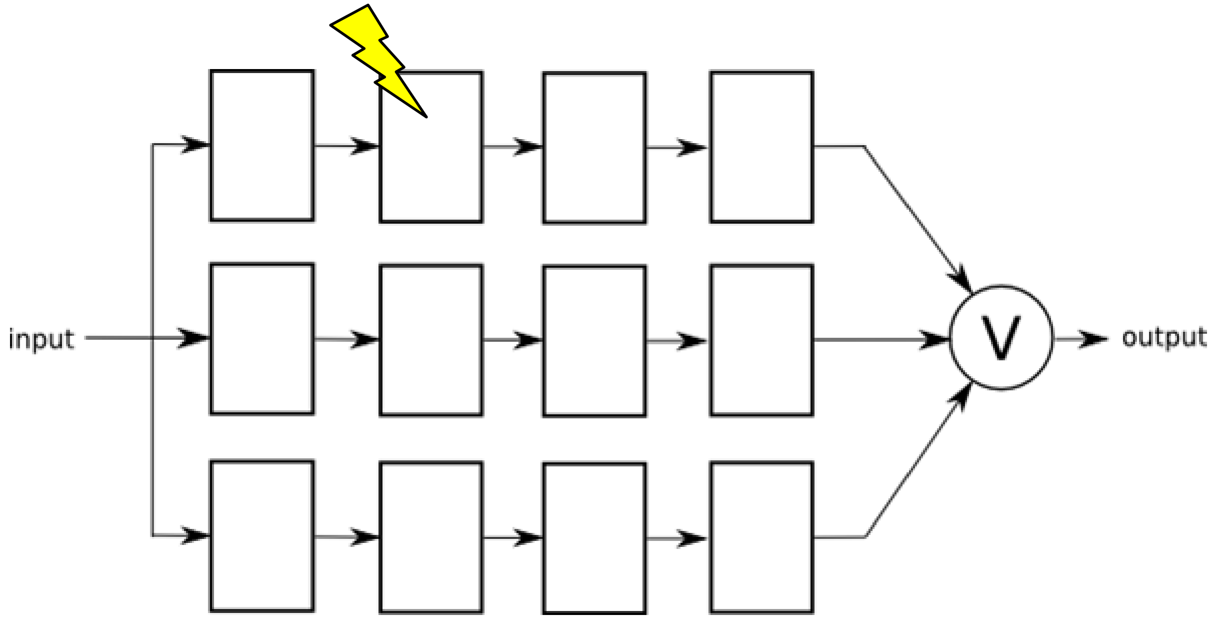


- Configuration Scrubbing
  - Readback frame data
  - Compare frame to original
  - Correct erroneous bits in frame
  - Writeback frame to FPGA



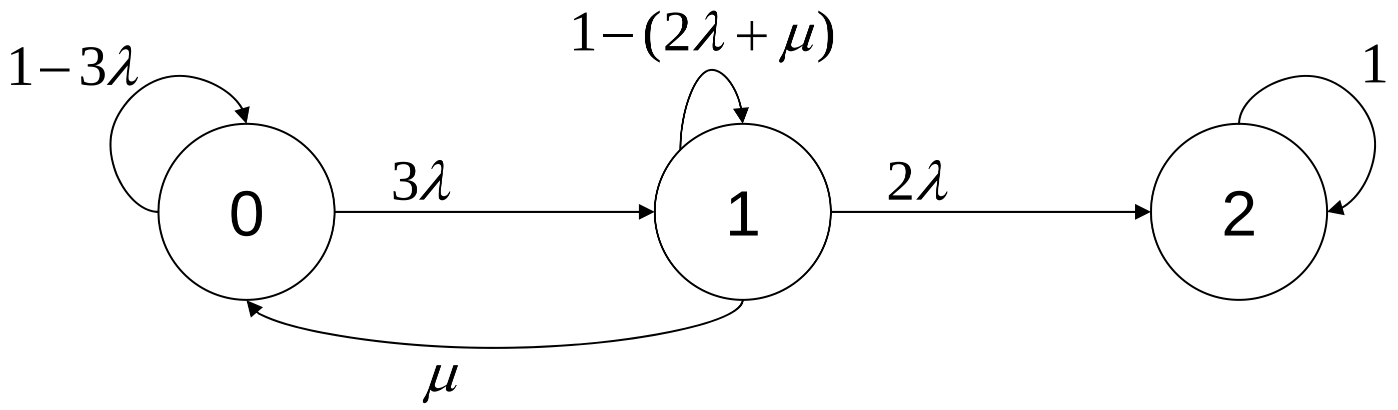
Triple Modular Redundancy (TMR)

- Three copies of each circuit module
- Majority voters select circuit output from redundant modules
- An upset affecting a single domain will not cause an error

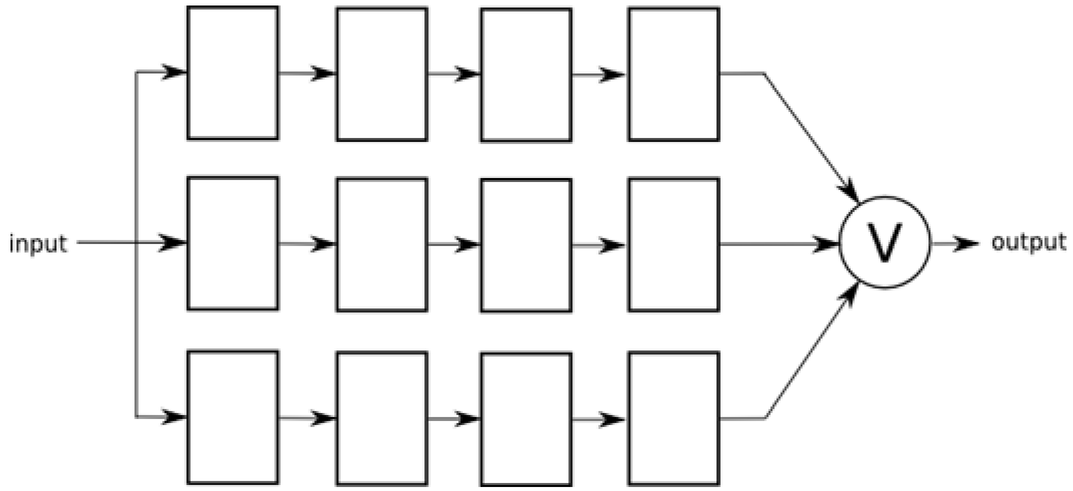


Reliability model for TMR with repair

- Markov model of TMR with repair
  - State 0: All three modules functioning
  - State 1: One of three modules has failed
  - State 2: Two or more modules have failed (TMR failure)
- A second upset before scrubbing can repair may cause TMR to fail

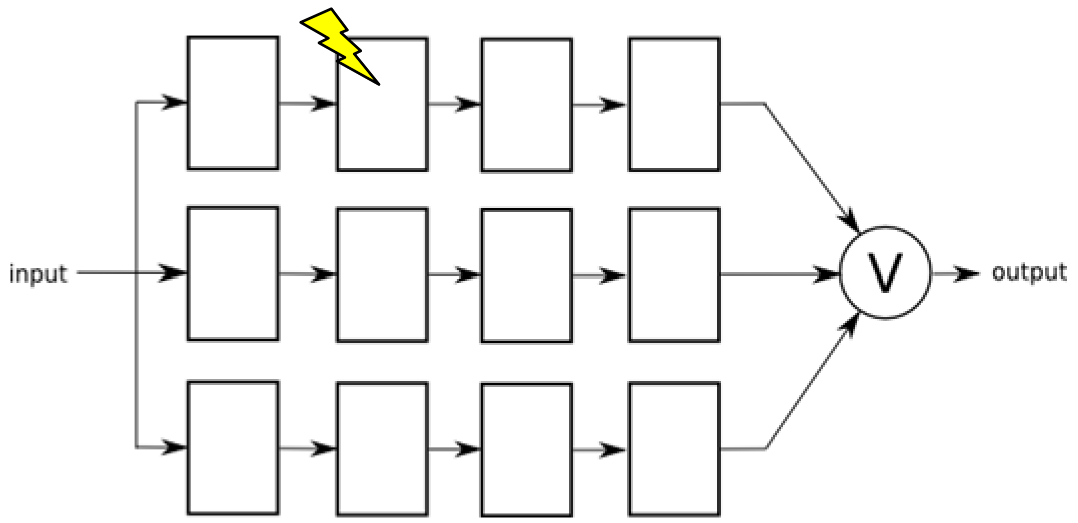

$$\begin{tikzcd} 0 \arrow[loop left, "1-3\lambda"] \arrow[r, "3\lambda"] & 1 \arrow[loop above, "1-(2\lambda+\mu)"] \arrow[l, bend left, "\mu"] \arrow[r, "2\lambda"] & 2 \arrow[loop right, "1"] \end{tikzcd}$$

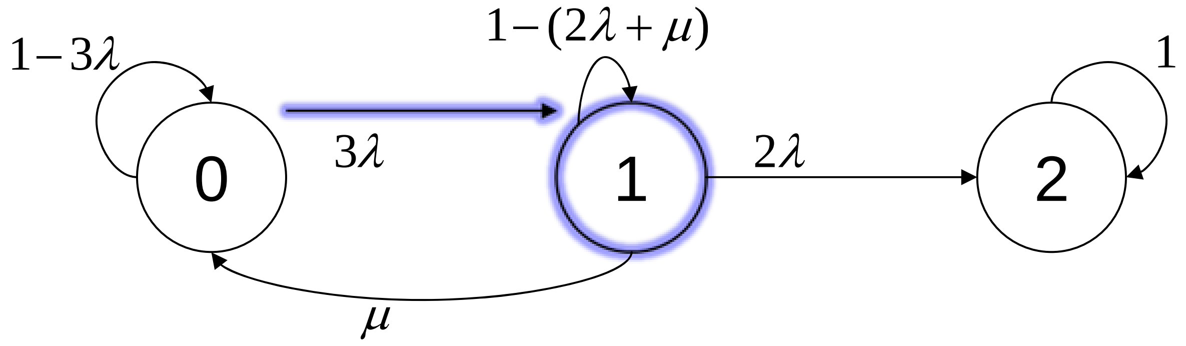
TMR Markov Model Example



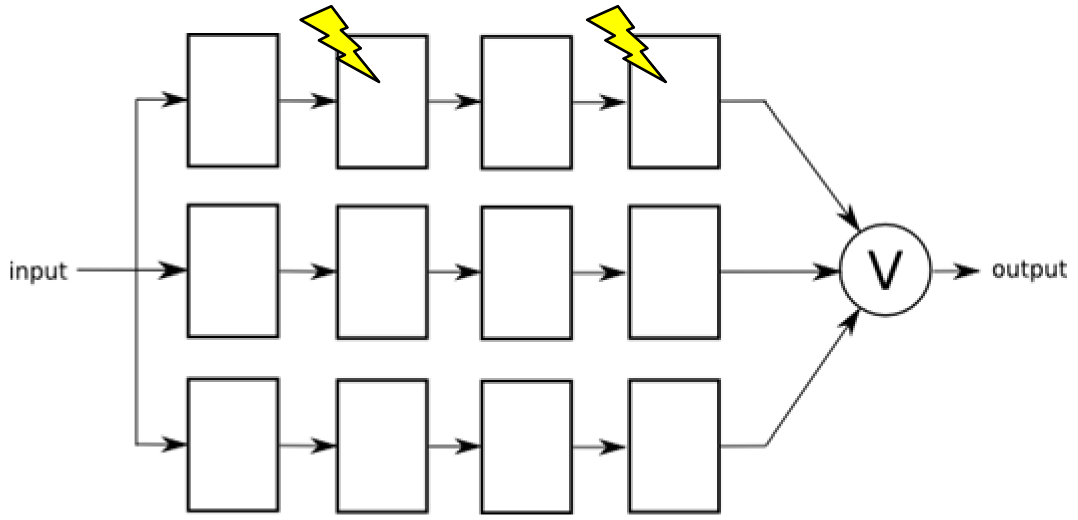
$$\begin{tikzcd}
0 \arrow[loop left, "1-3\lambda"] \arrow[r, "3\lambda"] & 1 \arrow[loop above, "1-(2\lambda+\mu)"] \arrow[r, "2\lambda"] \arrow[l, bend left, "\mu"] & 2 \arrow[loop right, "1"]
\end{tikzcd}$$

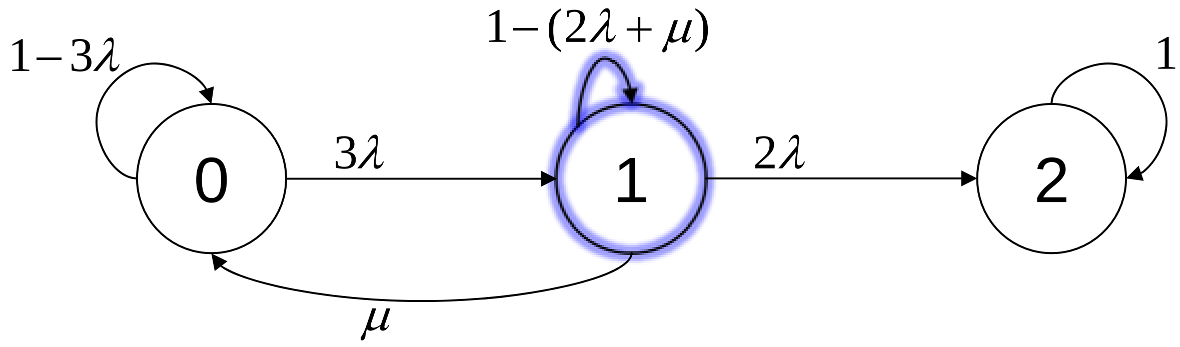
TMR Markov Model Example



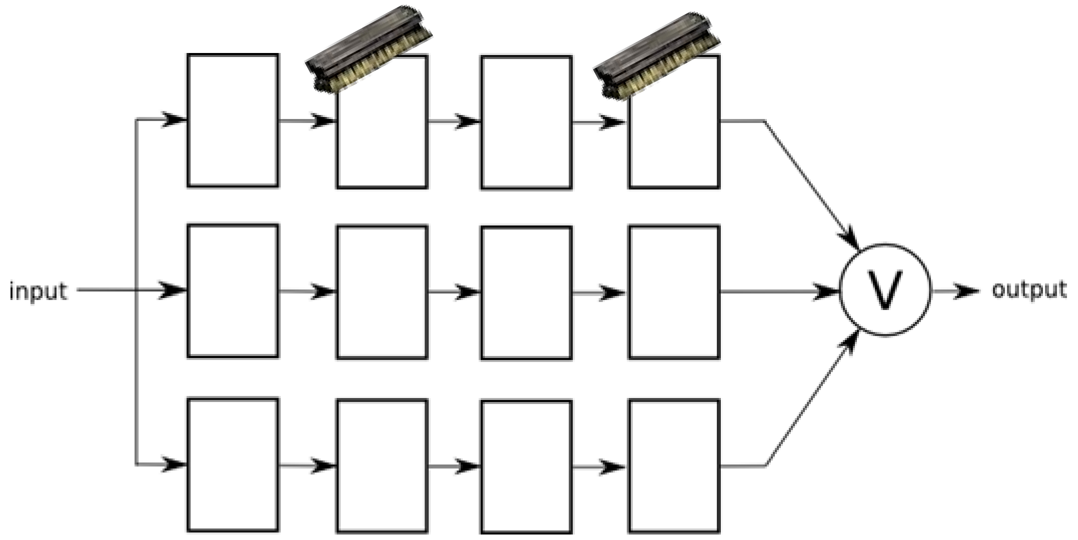


TMR Markov Model Example



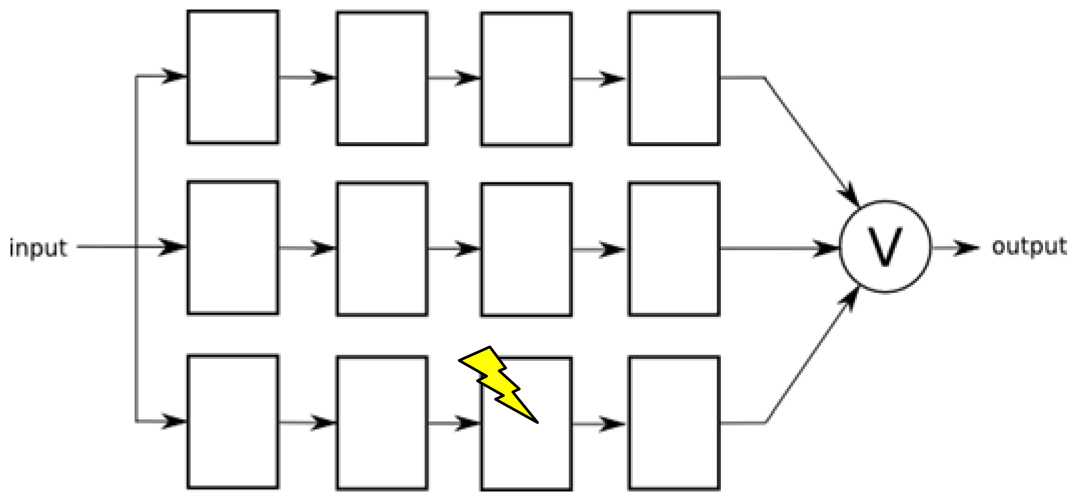


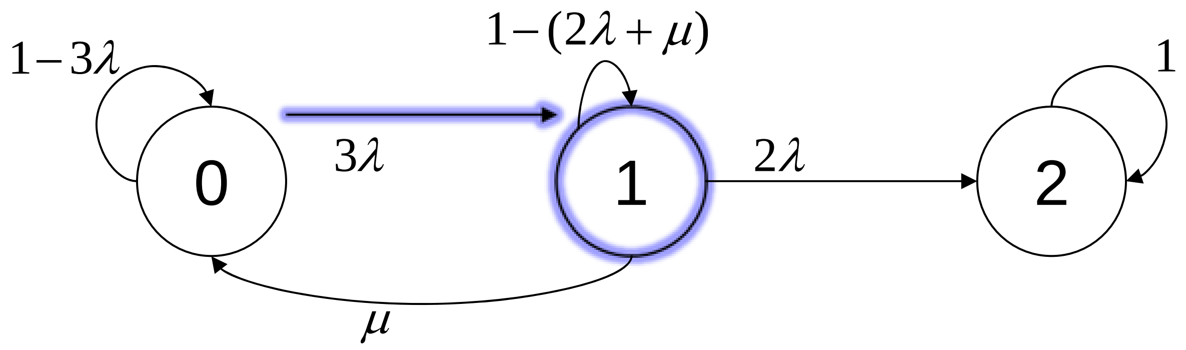
TMR Markov Model Example



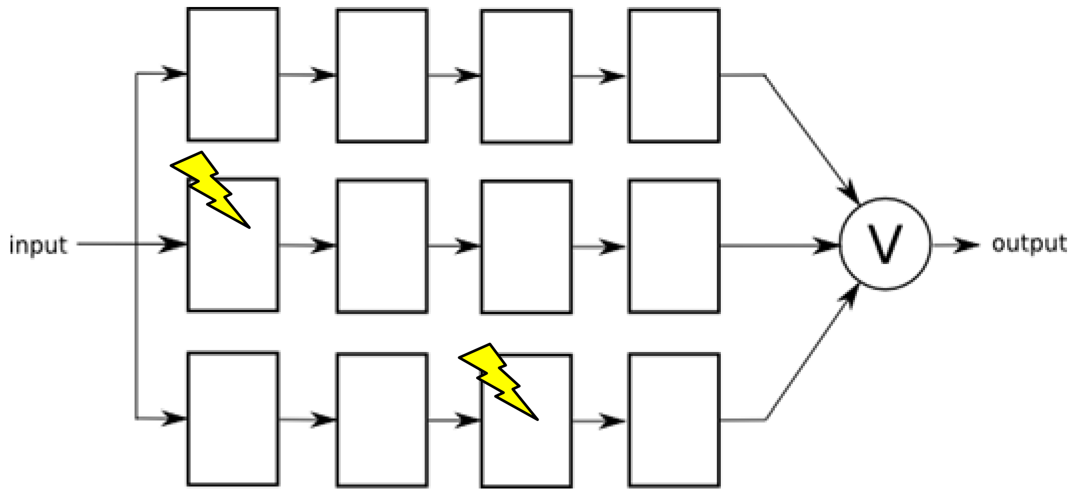
$$\begin{tikzcd}
0 \arrow[loop above, "1-3\lambda"] \arrow[r, "3\lambda"] & 1 \arrow[loop above, "1-(2\lambda+\mu)"] \arrow[l, bend left, "\mu"] \arrow[r, "2\lambda"] & 2 \arrow[loop above, "1"]
\end{tikzcd}$$

TMR Markov Model Example



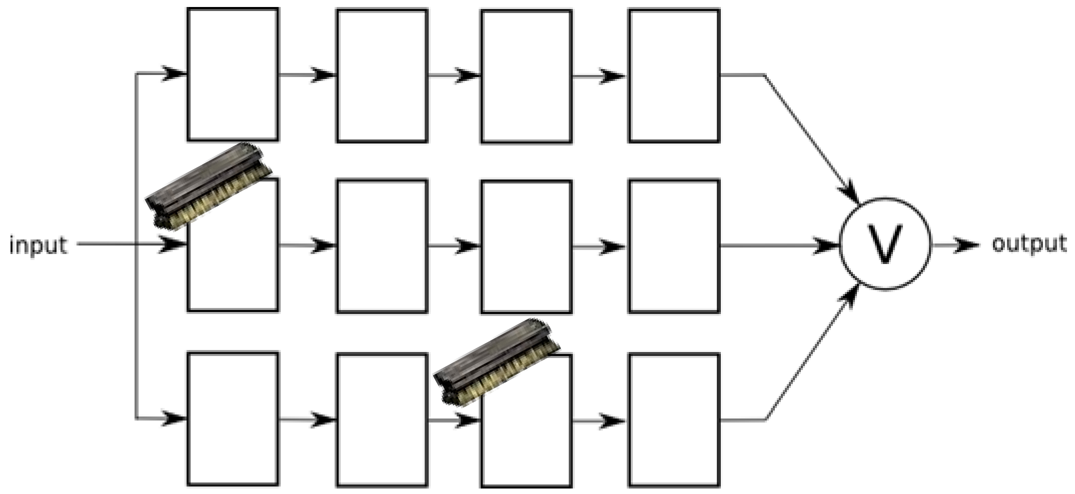


TMR Markov Model Example



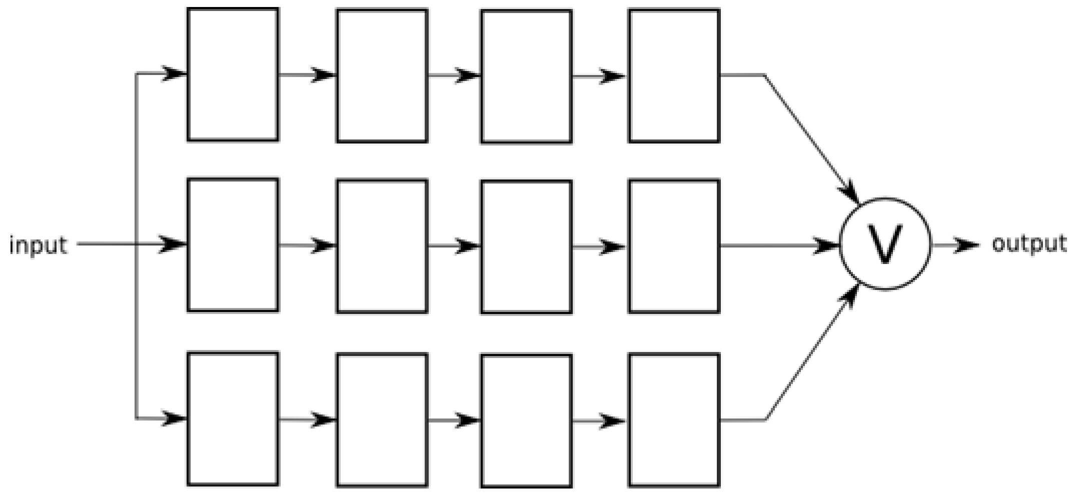
$$\begin{tikzcd}
0 \arrow[loop left, "1-3\lambda"] \arrow[r, "3\lambda"] & 1 \arrow[loop above, "1-(2\lambda+\mu)"] \arrow[l, bend left, "\mu"] \arrow[r, "2\lambda"] & 2 \arrow[loop right, "1"]
\end{tikzcd}$$

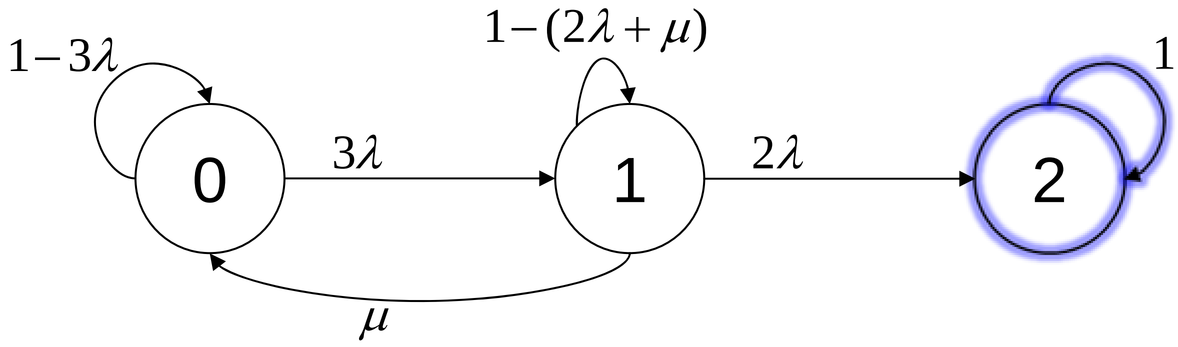
TMR Markov Model Example



$$\begin{tikzcd}
0 \arrow[loop above, "1-3\lambda"] \arrow[r, "3\lambda"] & 1 \arrow[loop above, "1-(2\lambda+\mu)"] \arrow[l, bend left, "\mu"] \arrow[r, "2\lambda"] & 2 \arrow[loop right, "1"]
\end{tikzcd}$$

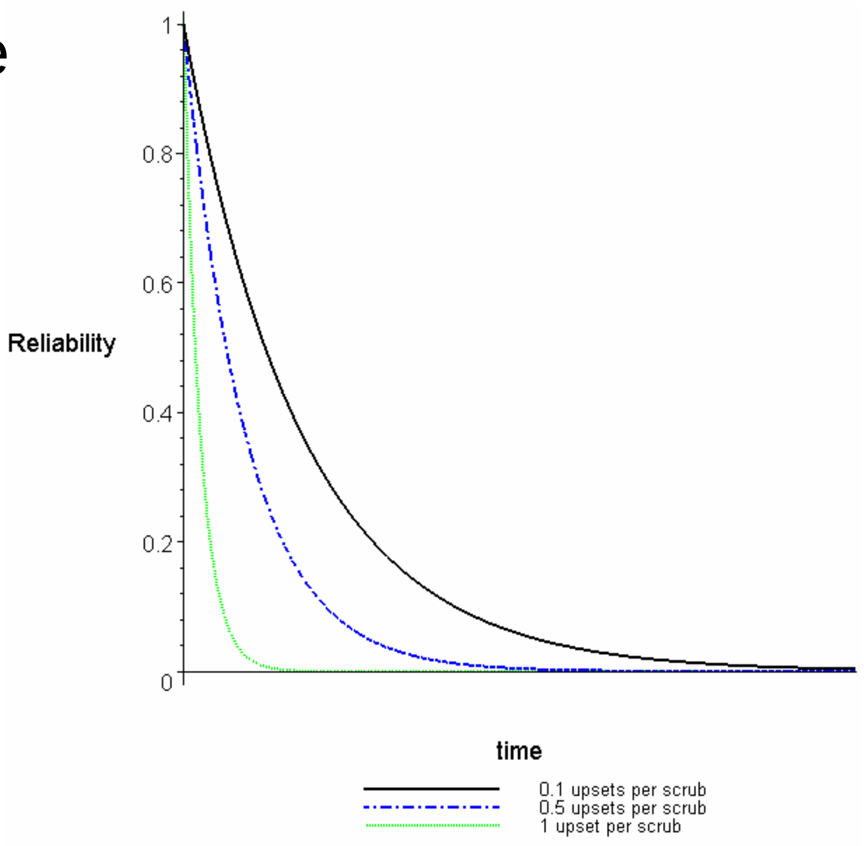
TMR Markov Model Example





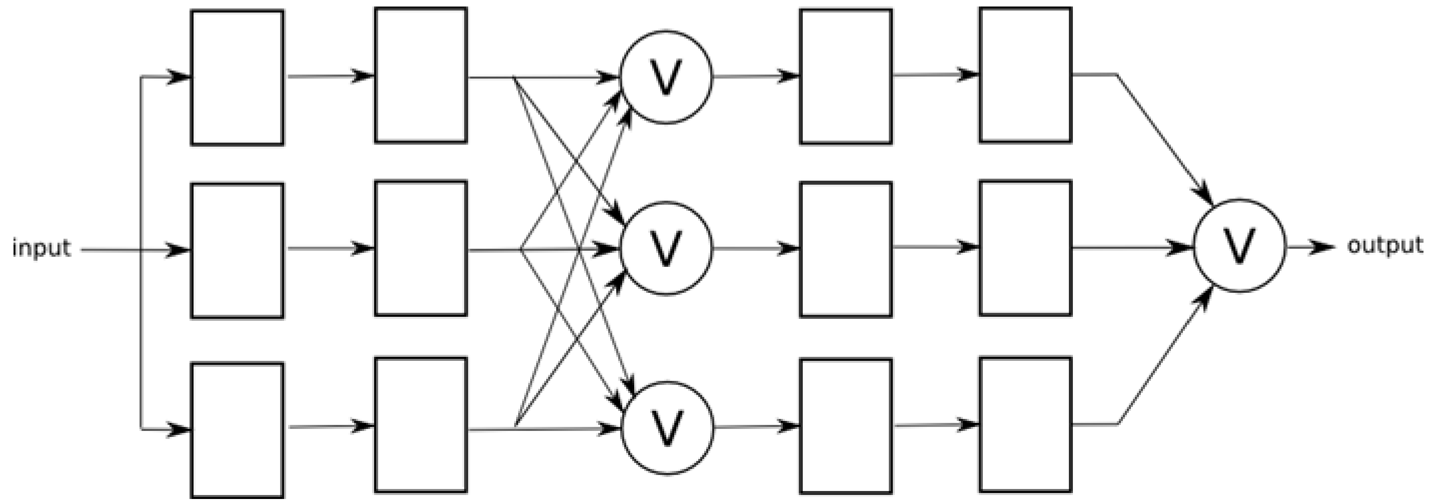
Reliability of TMR with repair

- Different combinations of upset rate and repair (scrubbing) rate result in distinct reliability curves
- The upset rate/repair rate ratio determines the reliability of the system
- Faster scrub rates needed for harsher radiation environments



TMR with More Frequent Voting (MFV)

- Partition design into smaller modules
- Separate partitions with voters
- Each partition is isolated from others
- Each partition has lower probability of failure
- Combined probability of failure is also lower



TMR with More Frequent Voting (MFV)

- Our analysis and experiments address the effects of multiple independent upsets (MIUs)
  - Due to harsh radiation environments
- Failures due to multiple-bit upsets (MBUs) are not addressed here
- Previous work has addressed single-bit domain crossing upsets in relation to TMR with partitions
  - F. Lima Kastensmidt, L. Sterpone, L. Carro, M. Sonza Reorda, "On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs," pp. 1290-1295, 2005.

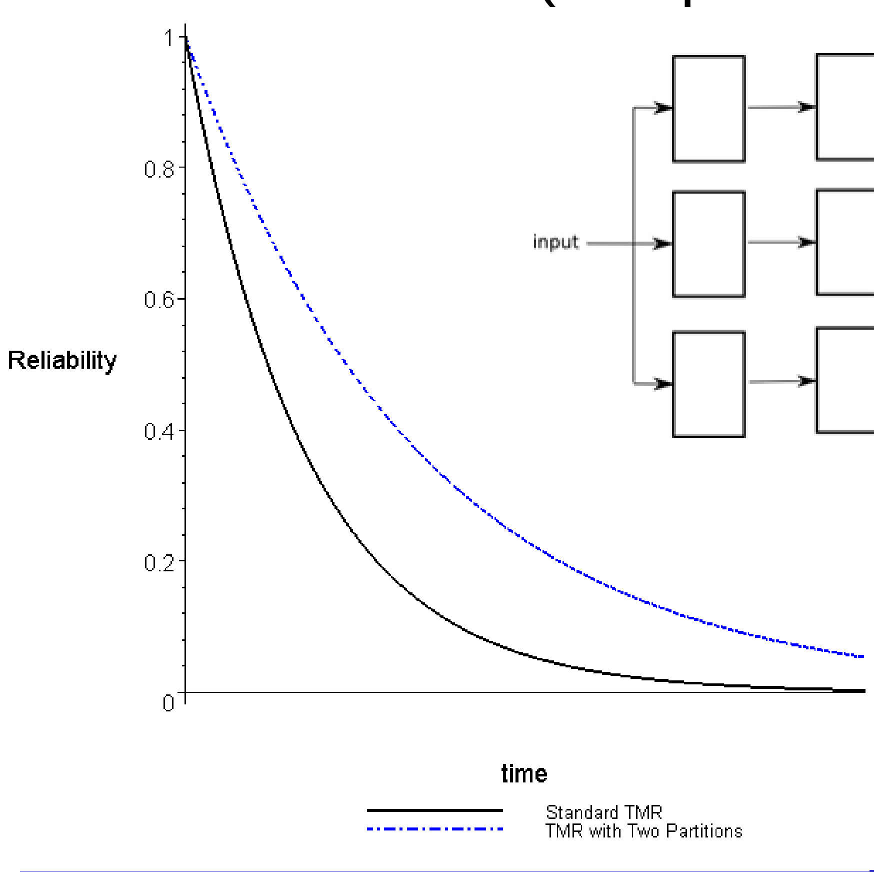
Reliability model for MFV TMR

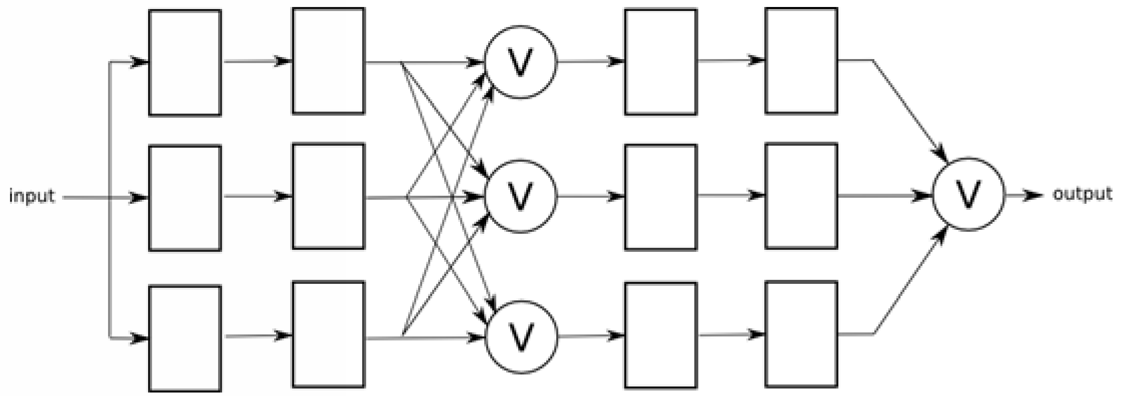
- Markov model of TMR with two partitions
  - State 0: All three modules functioning
  - State 1: One module in one partition has failed
  - State 2: One module in each partition has failed
  - State 3: Two or more modules in the same partition have failed (TMR failure)
- A second upset *in the same partition* may cause TMR to fail

$$\begin{tikzcd}
0 \arrow[r, "6\frac{\lambda}{2}"] & 1 \arrow[r, "3\frac{\lambda}{2}"] \arrow[l, bend left, "\mu"] \arrow[rr, bend left, "2\frac{\lambda}{2}"] & 2 \arrow[r, "4\frac{\lambda}{2}"] \arrow[ll, bend left, "\mu"] & 3 \arrow[loop right, "1"]
\end{tikzcd}$$

Reliability of MFV TMR – Two partitions

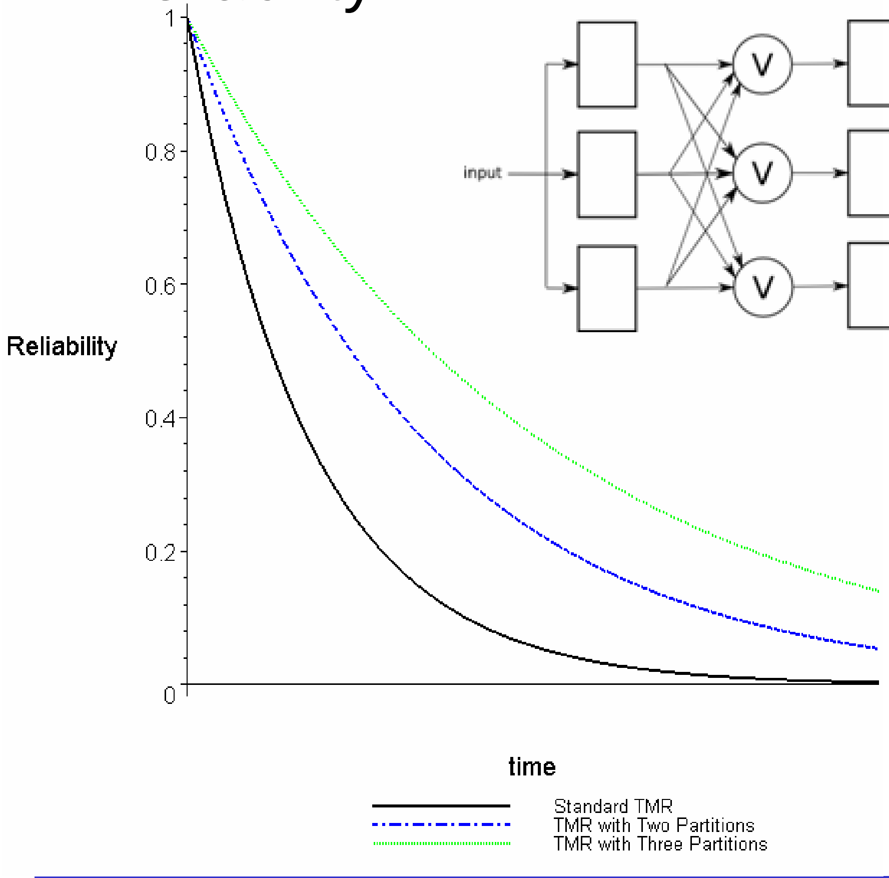
- TMR with two partitions is more reliable than standard TMR (one partition)

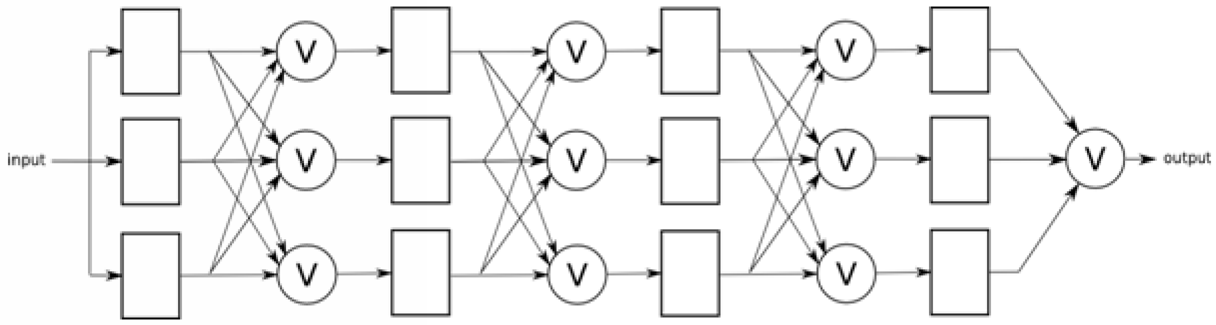




Multiple Partitions

- In general, more partitions will increase overall reliability



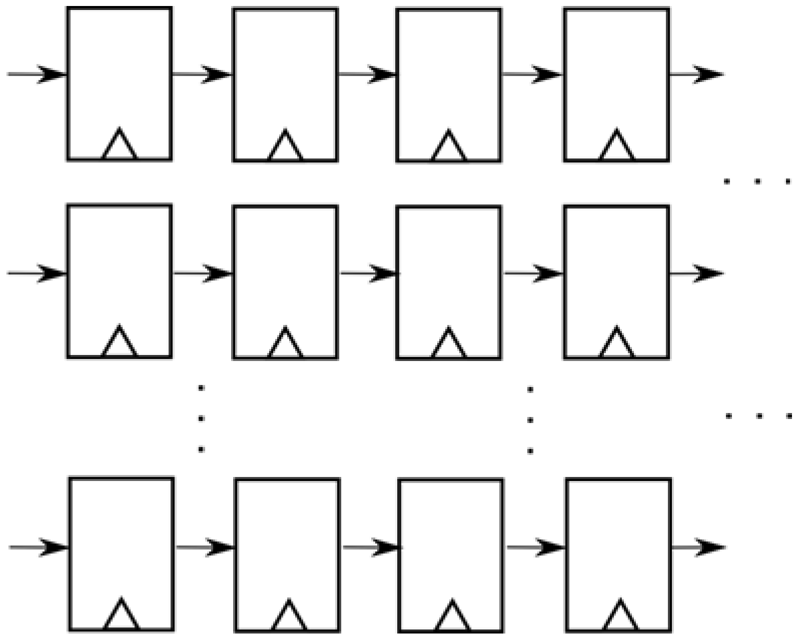


More frequent voting study

- Goals of this study
  - Evaluate effectiveness of TMR with more frequent voting on FPGA technology
- Use fault injection to evaluate reliability
  - Insert various numbers of faults before evaluating and repairing
    - To simulate different upset rate/repair rate ratios
- Question: How does reliability improve?

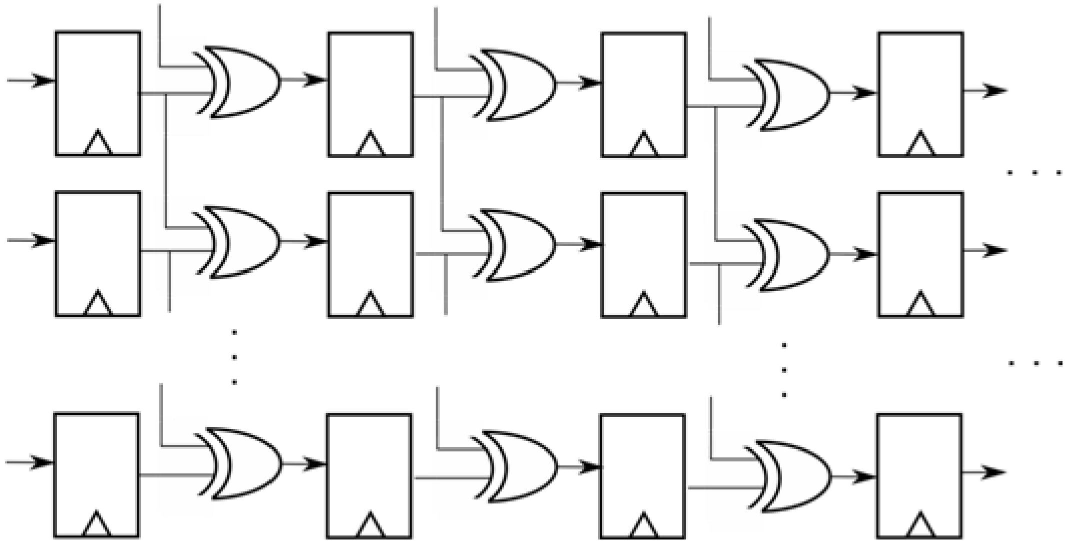
Test Design #1

- 32-bit shift register designed to fill target device
  - 100 registers deep
- Tested on Xilinx Virtex 1000 device



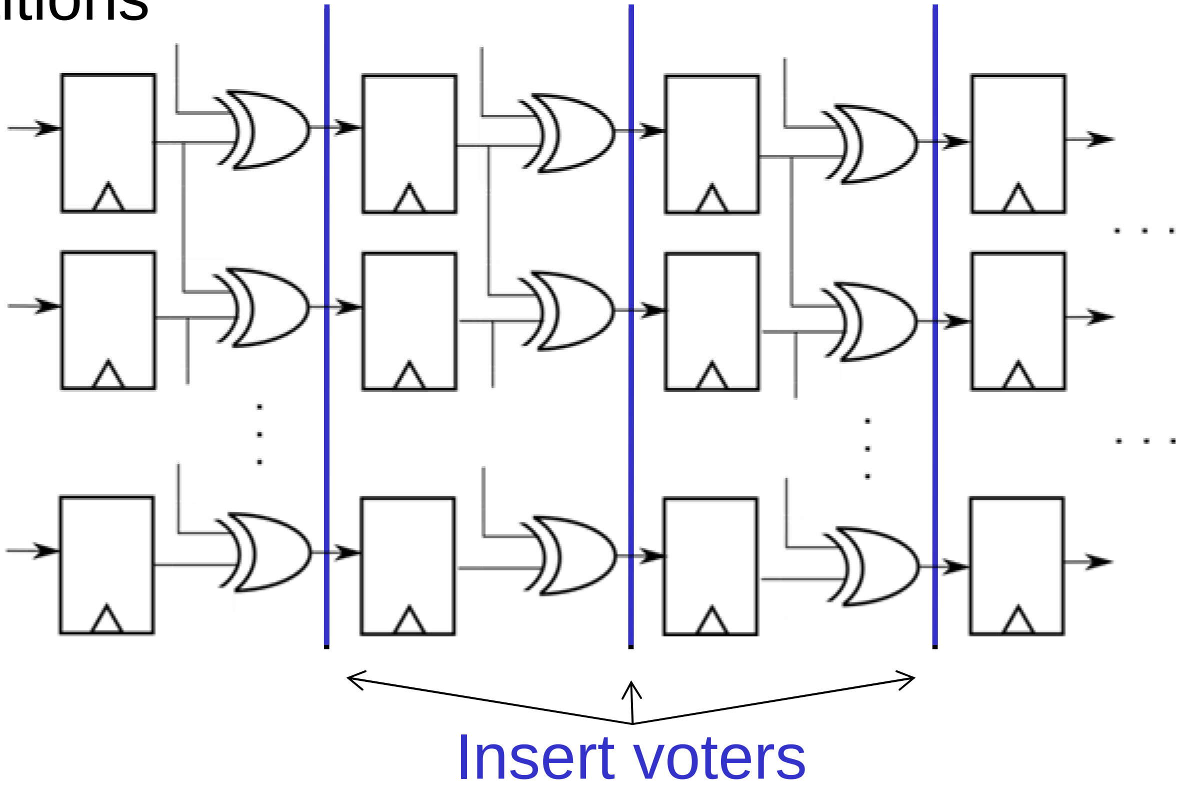
Test Design #2

- 32-bit shift register designed to fill target device
  - 250 registers deep
  - Shift register at each bit position affects all others due to xor operations
    - To prevent isolation between bit positions
- Tested on Xilinx Virtex 4 SX55 device



Test Design with more frequent voting

- Partitions separated with triplicated voters
- Multiple design points with different numbers of partitions

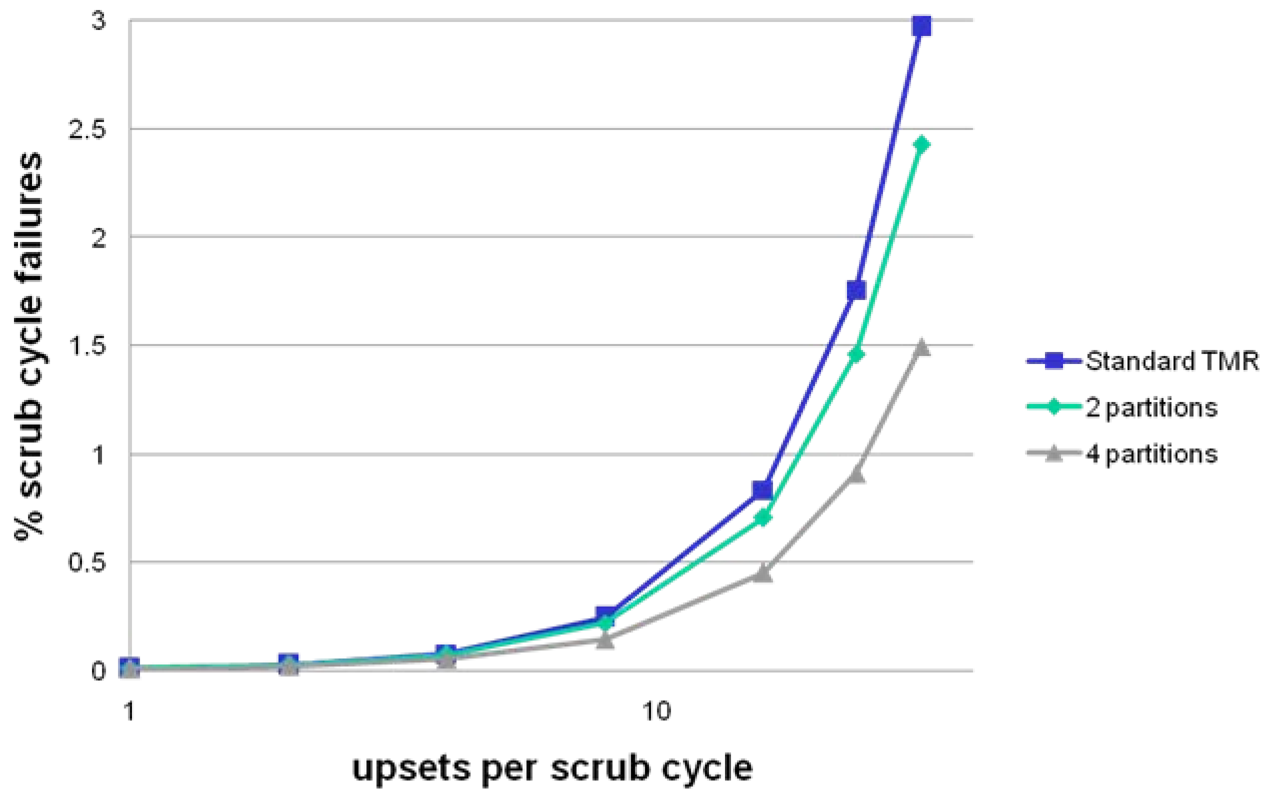


Test Methodology

- Two test platforms
  - SLAAC-1V with Xilinx Virtex 1000
  - SEAKR board with Xilinx Virtex 4 SX55
- Experiment with different numbers of partitions
  - Several design points created from each test design
- Evaluate each design point with multiple upset rates
  - Simulating different upset rate to repair rate ratios
- Measure results with fault injection

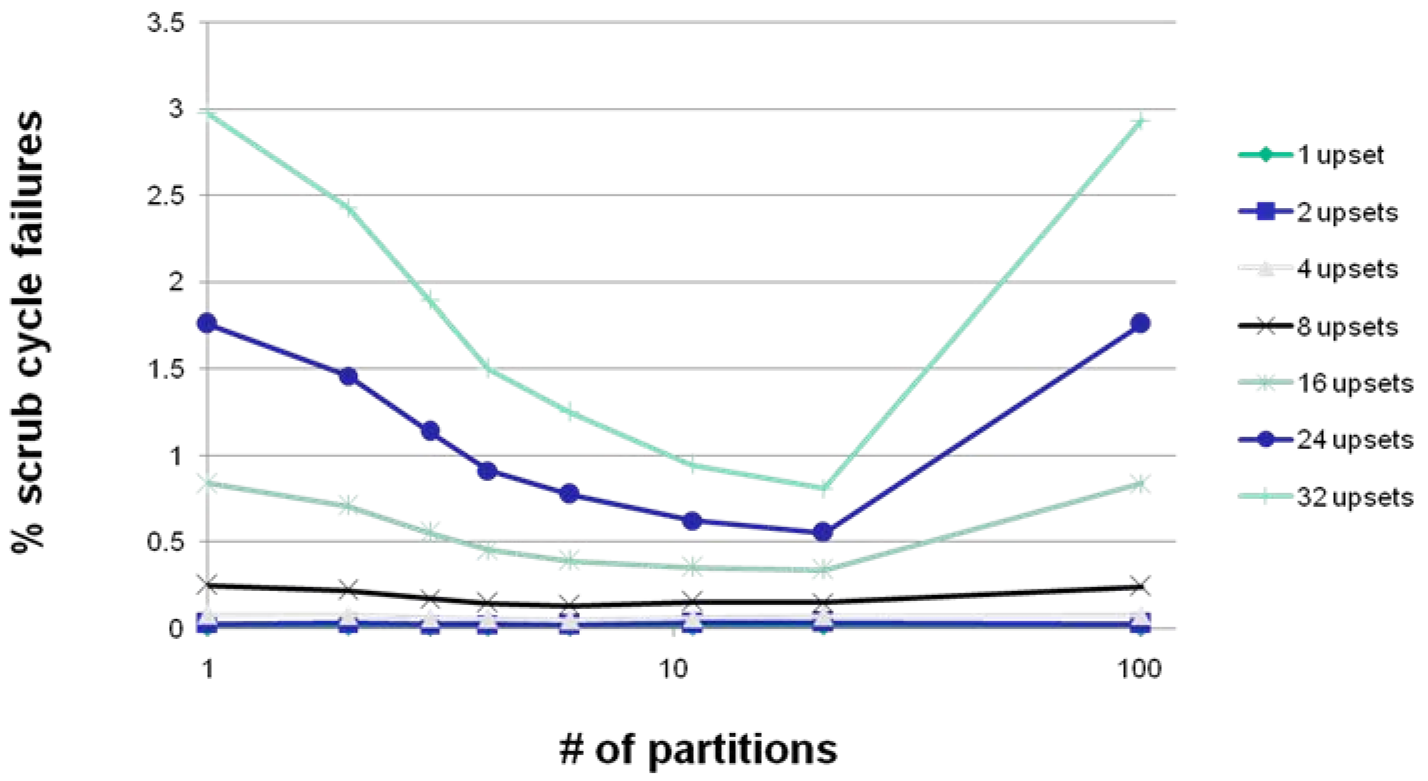
Fault Injection Results

- Test Design #1 on Virtex 1000
- Reliability increased with more partitions



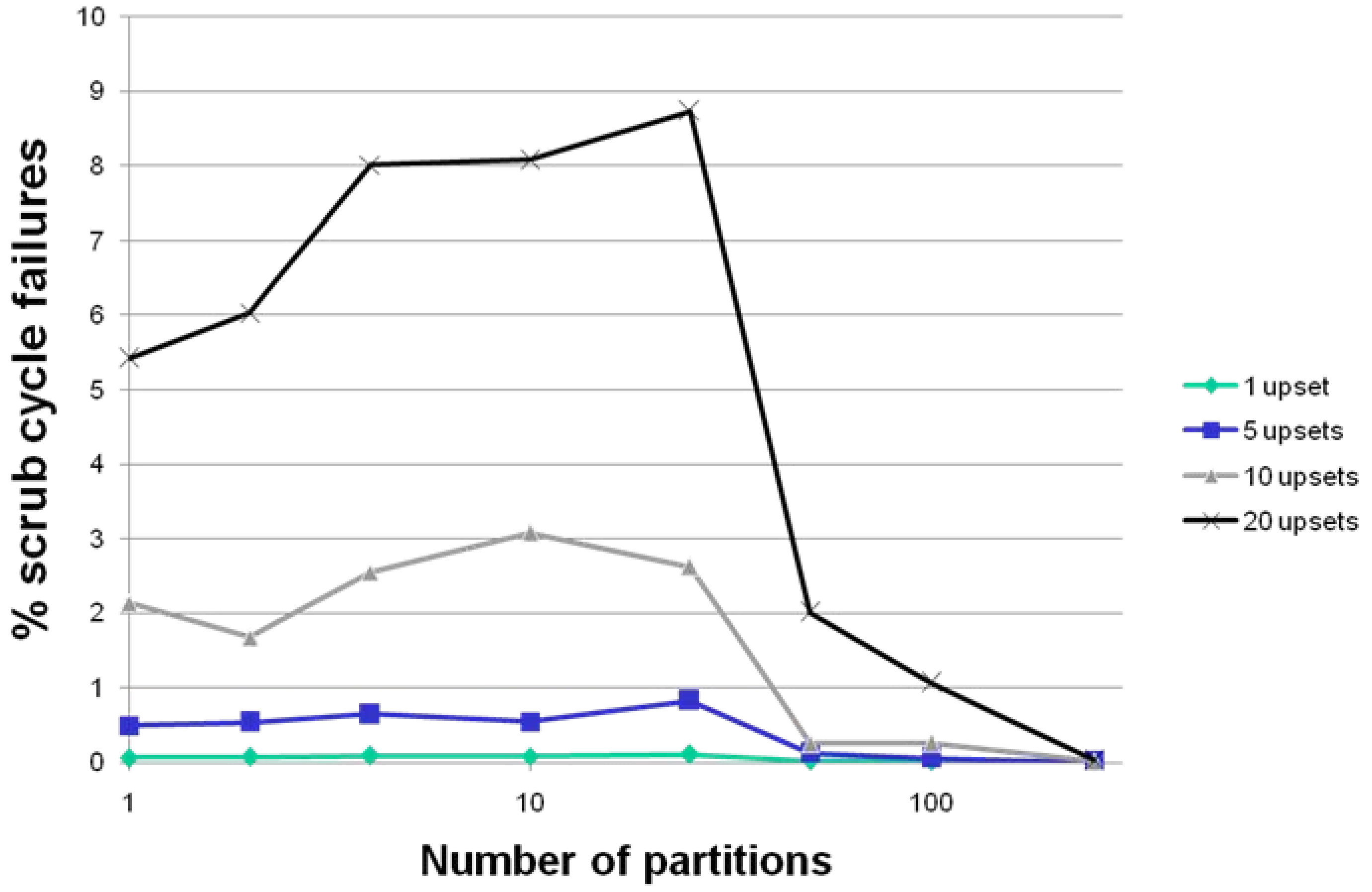
Fault Injection Results

- Test Design #1 on Virtex 1000



Fault Injection Results

- Test Design #2 on Virtex 4 SX55

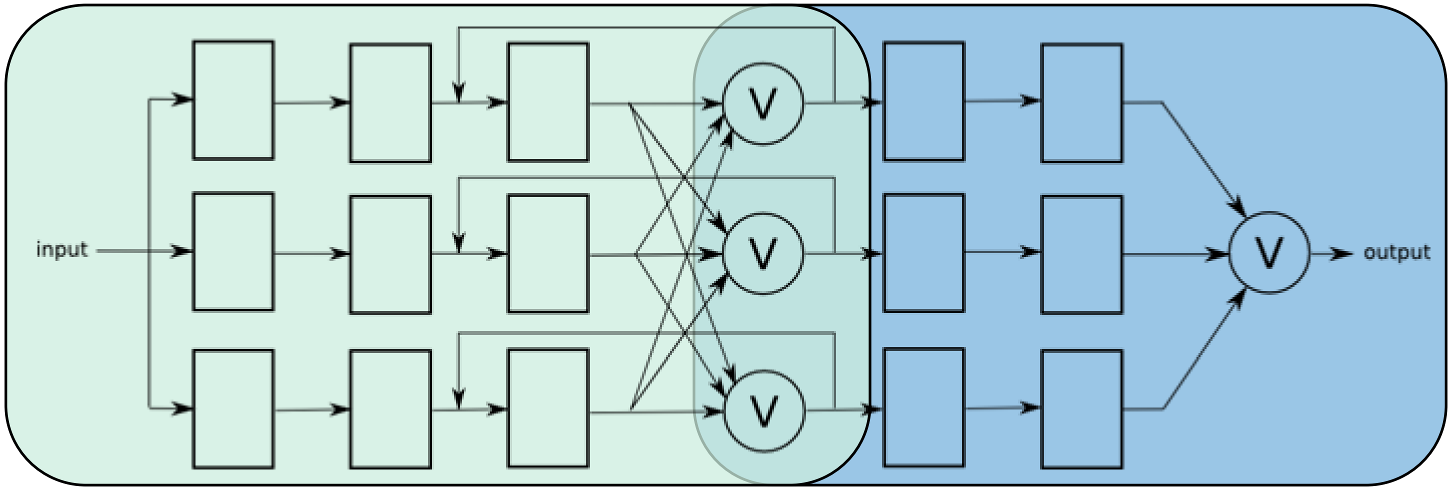


Summary of Results

- Results for V4 design showed that reliability *decreased* as partitions were added at first
  - Reliability increased after 25 partitions for this design
  - Routing may have had an impact
    - Routing behavior changes with density of device
  - More domain-crossing single-bit failures due to larger design size?
- Reliability improvements are modest
- Largest gains in reliability for larger number of upsets per scrub cycle

Natural Partitioning

- Some partitioning naturally occurs with standard TMR
- Feedback TMR
  - Voters inserted in feedback loops, creating partitions
- Parallel streams of logic
  - Portions of logic that do not affect each other
  - e.g. bits in a simple shift register



Conclusions

- TMR with more frequent voting increases reliability in the presence of multiple upsets
- Reliability gains are not large for low upset/scrub rates
- Future work:
  - Evaluate cost vs. benefit
  - Understand architectural differences
  - Radiation testing
  - Evaluate effects of natural partitioning