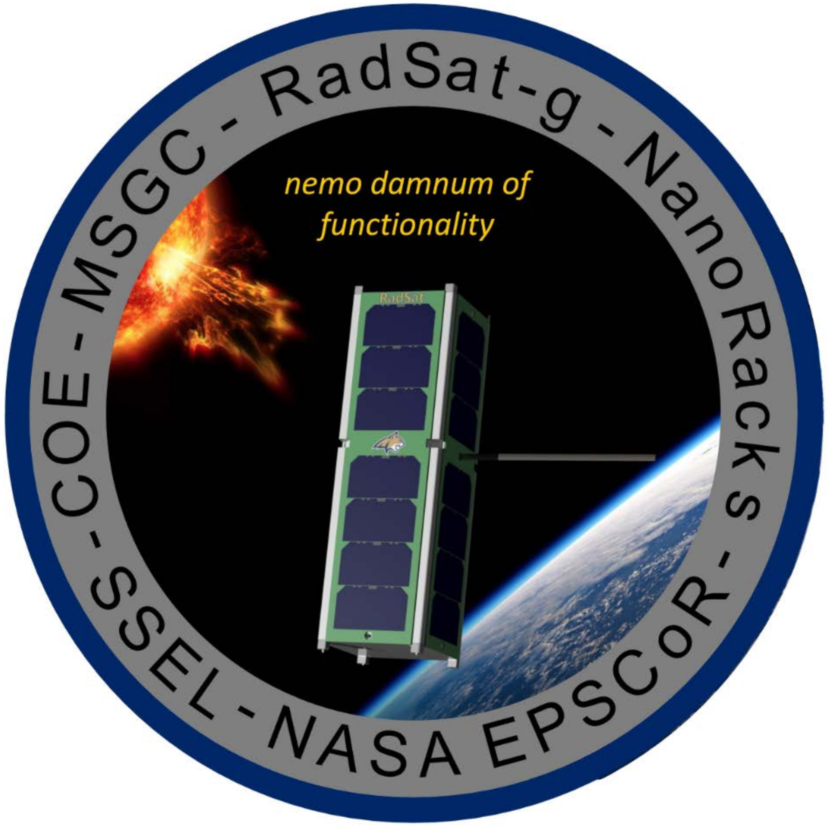
Next on the Pad – “RadSat”

A Radiation Tolerant Computer System

Authors

Montana State University

Brock J. LaMeres, Colin Delaney, Matt Johnson,
Connor Julien, Kevin Zack, Ben Cunningham
Todd Kaiser, Larry Springer, David Klumpar



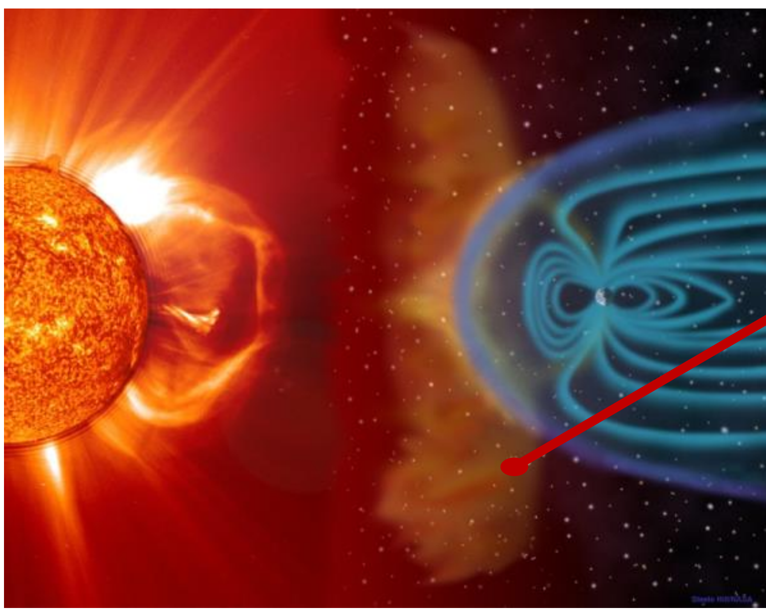
Montana State University College of Engineering

Montana State University SSEL Space Science & Engineering Lab

Objective of the RadSat Mission

Demonstrate a Single Event Effect (SEE) Mitigation Strategy

- The computer delivers radiation tolerance through a reconfigurable/redundant architecture.
- The computer delivers low cost using COTS parts.
- The computer delivers higher performance (computation & power efficiency) by exploiting modern process nodes (Artix-7).



Why Does the SmallSat Community Care?

Computation

- SmallSats are doing more and more on-board data processing (e.g., *images, sensor data, communications*).

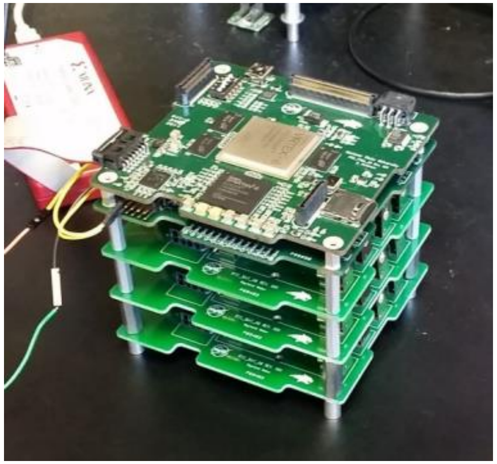
Radiation Tolerance

- Cutting edge process nodes (28nm) provide increased computation but are becoming more susceptible to radiation induced faults (SEEs).
- As SmallSat missions achieve longer duration and move into deep space, radiation becomes more and more of a concern (both TID & SEE).

Cost

- Any SmallSat computing solution must be cost effective to align with SmallSat theme.

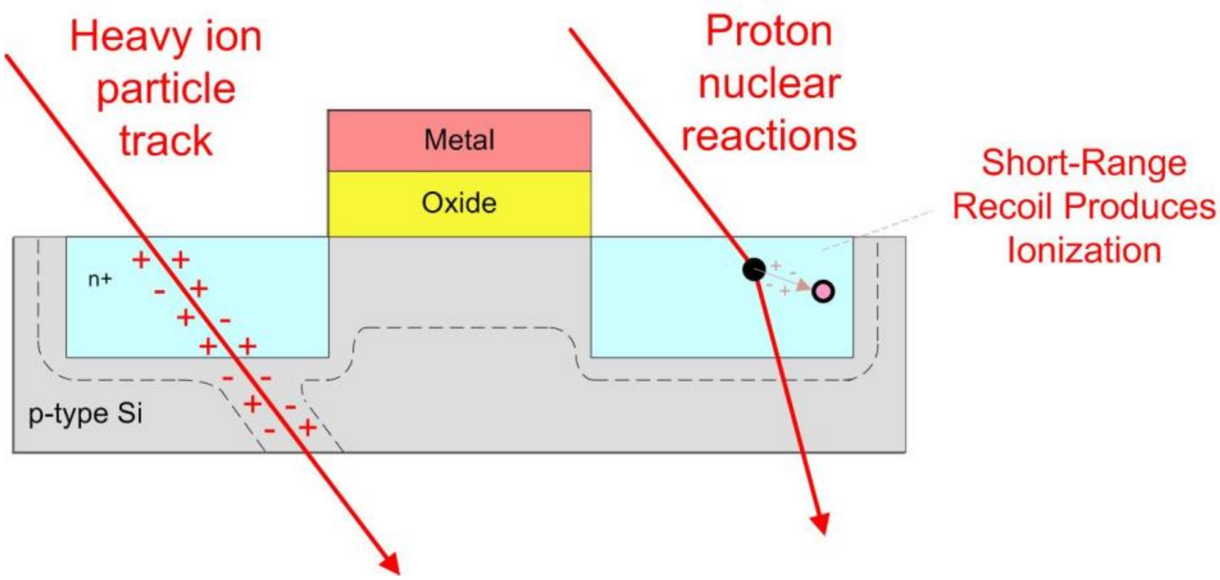
  (i.e., *"launch more, inexpensive, satellites"*)



Single Event Effects (SEE)

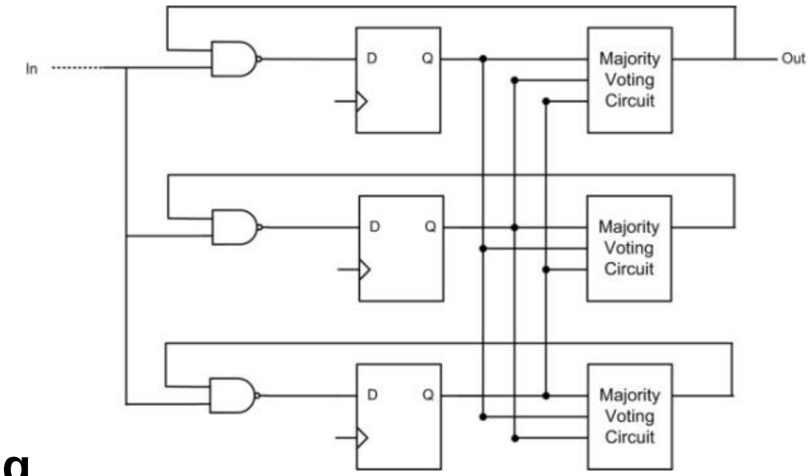
- Electron/hole pairs created by a single particle passing through semiconductor.
- Primarily due to heavy ions and high energy protons.
- Excess charge carriers cause current pulses.
- Creates a variety of destructive and non-destructive damage.

"Critical Charge" = the amount of charge deposited to change the state of a gate



How We Currently Deal with Radiation

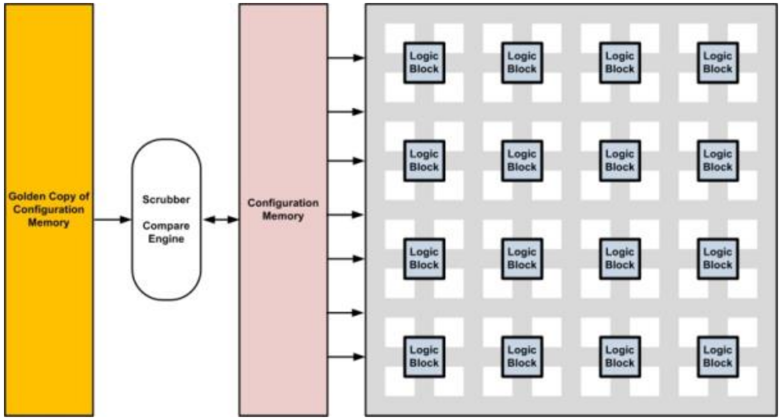
Dealing with Single Event Effects



- **Architecture: Triple Module Redundancy**
  - o Triplicate each circuit
  - o Use a majority voter to produces output

- **Background Memory Checking: Scrubbing**
  - o Compare contents of a memory device to a "Golden Copy"
  - o Golden Copy is contained in a radiation immune technology (fuse-based memory, MROM, etc…)

Note: TMR+Scrubbing is the recommended mitigation approach for FPGA-based aerospace computers

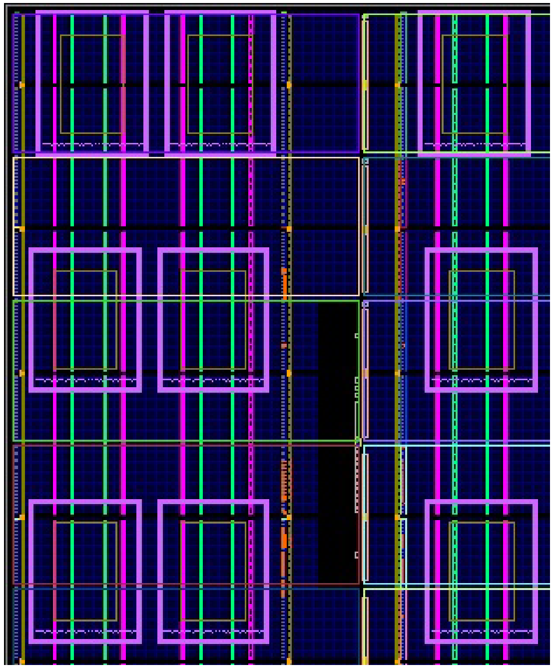


Fault Tolerance Through Abundant Spares

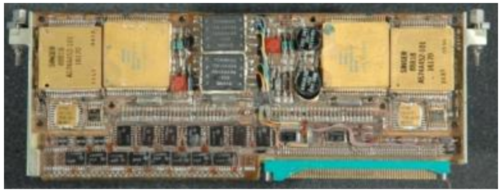
1. **TMR + Spares**
   - 3 Tiles run in TMR with the rest reserved as spares

2. **Spatial Avoidance and Background Repair**
   - If TMR detects a fault, the damaged tile is replaced with a spare and foreground operation continues
   - The tile is "repaired" in the background via **partial reconfiguration (PR).**

3. **Scrubbing**
   - Blind scrubbing continually runs through tiles (fast)
   - Readback scrubbing periodically runs through rest of fabric (slower)



9 MicroBlaze Processors on Artix-7



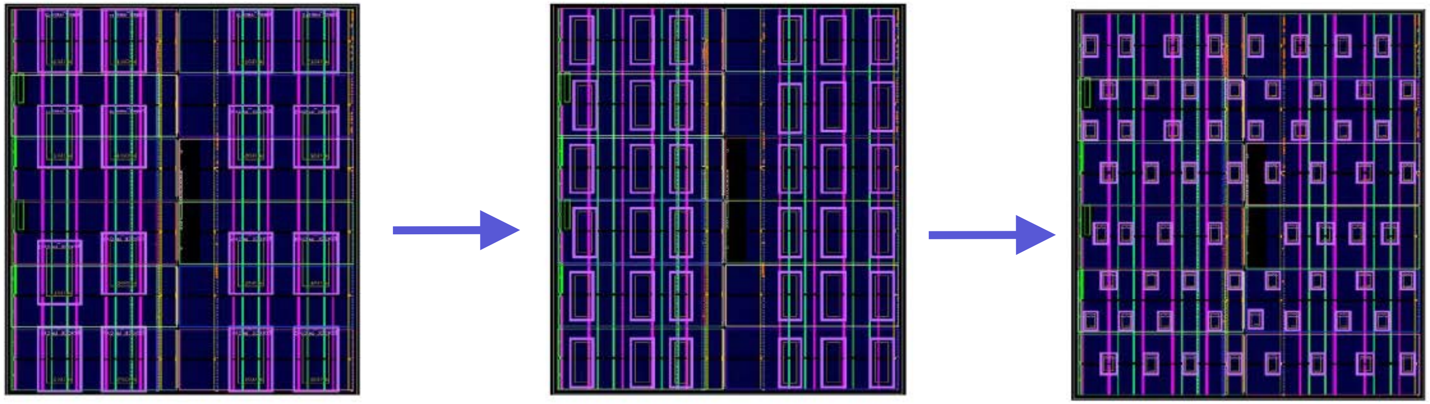
Precedent: Shuttle Flight Computer (TMR + Spare)

Our Approach

Why do it this way?

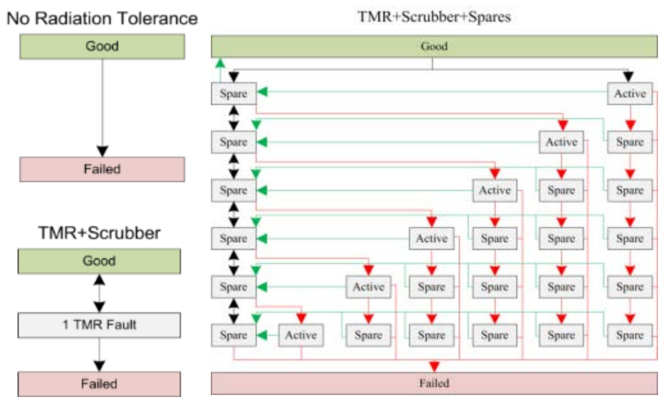
With Spares, it basically becomes a flow-problem:

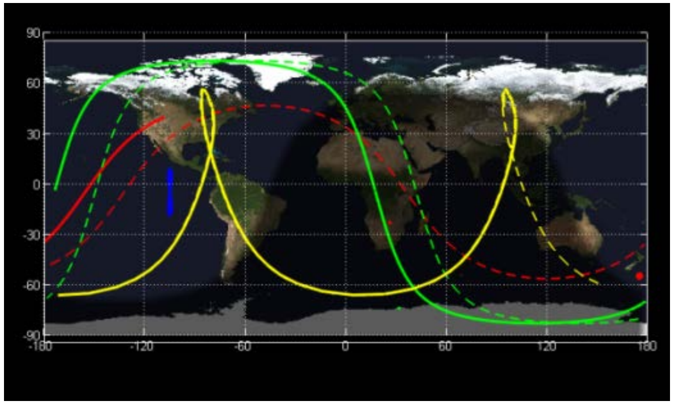
- TMR produces the right output, but repair is inevitable.
- Partial Reconfiguration is faster than Full Reconfiguration.
- Brining on a spare is faster than Partial Reconfiguration.
- If the repair rate is faster than the incoming fault rate, you’re safe.
- If the repair rate is slightly slower than the incoming fault rate, spares give you additional time.
- The additional time can accommodate varying flux rates.
- Abundant resources on an FPGA enable dynamic scaling of the number of spares.



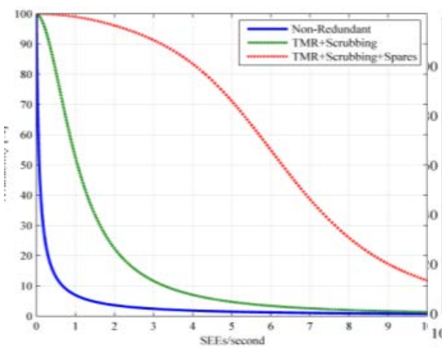
Modeling: Is this an improvement to TMR+Srubbing?

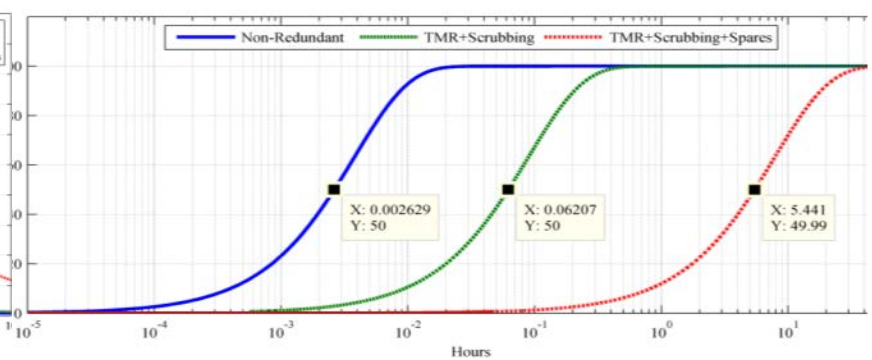
- We use a Markov Model to predict *Mean-Time-Before-Failure.*
- We want to see if it improves MTBF over non-redundant & TMR+scrubbing.
- The fault rate was extracted from CREME96 for 4 different orbits for Virtex-6 FPGA.
- The repair rate was found empirically.





ISS
HEO
HRBE
GEO



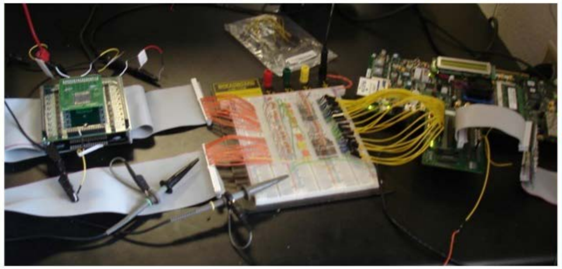


Ok, it looks promising…

History of Technology Maturation

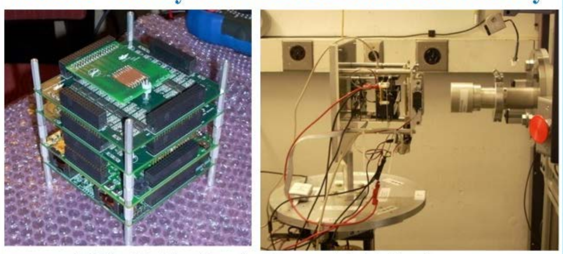
10 years…

TRL 3 – Proof of Concept



2008-2009: Prototype demonstration at MSFC.

TRL 4 – Subsystem Validation in Laboratory



2010-12: Testing in Texas A&M Cyclotron.

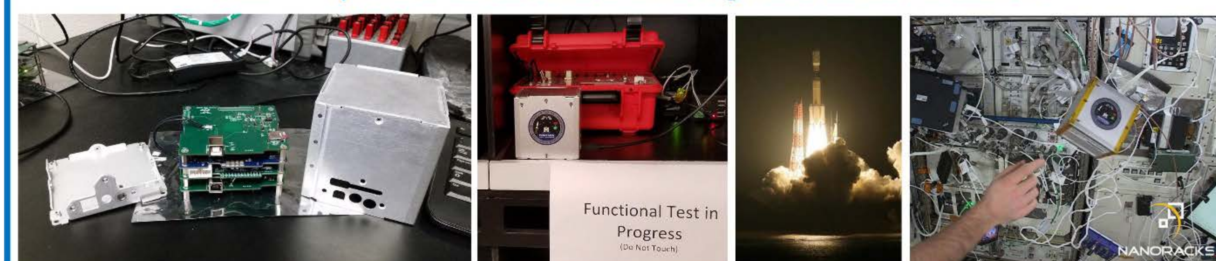
TRL 5 – Subsystem Validation in Relevant Environment

2011-13: High Altitude Balloon Testing (MSGC BOREALIS + LSU HASP).

TRL 6 – Subsystem Demonstration in Relevant End-to-End Environment

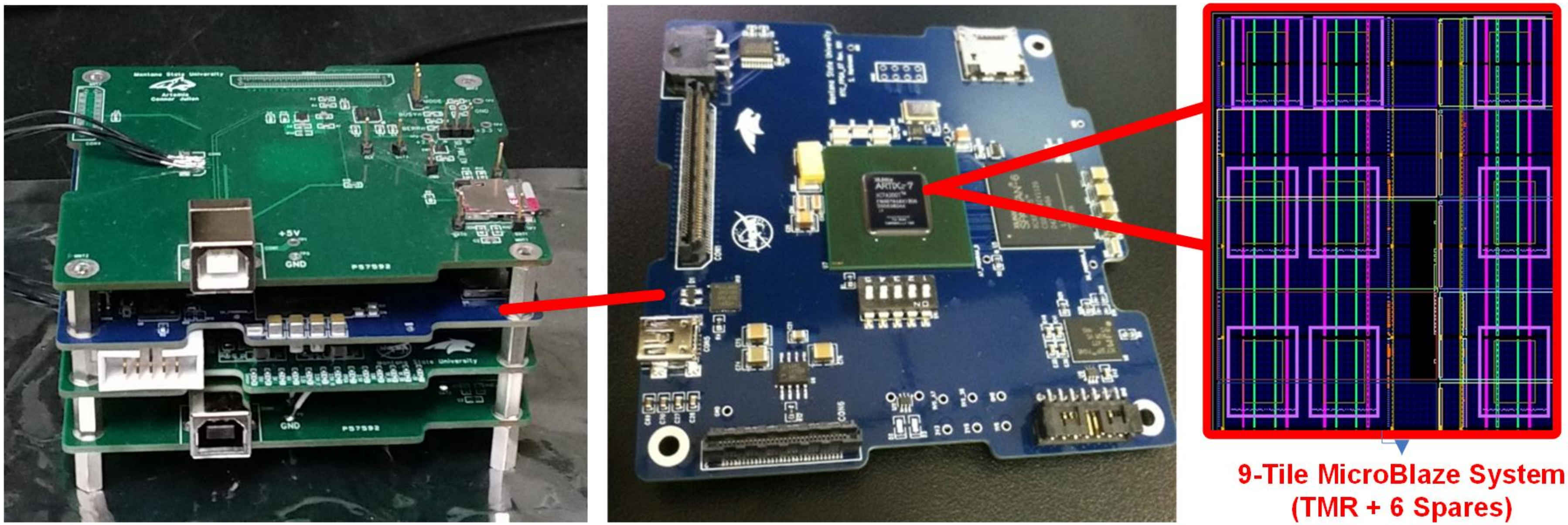
2014-16: Sounding Rocket Testing (UP Aerospace SpaceLoft-9 + Terrior-Improved Orion).

TRL 7 – System Demonstration in an Operational Environment



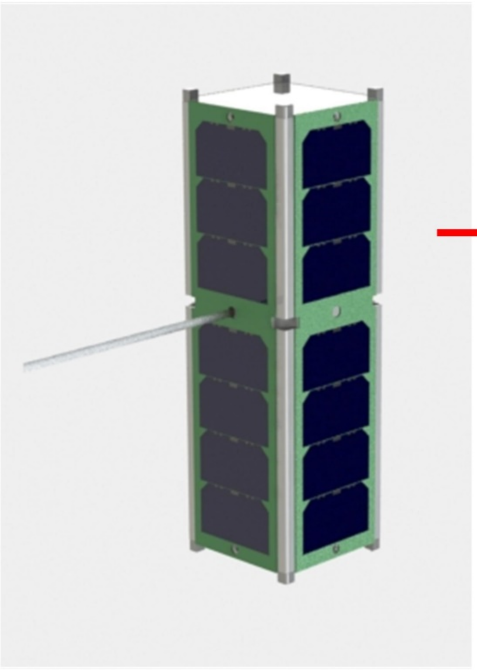
2014-16: Internal ISS Demonstration using NanoRacks CubeLab Experiment Locker (HTV6 Launch).

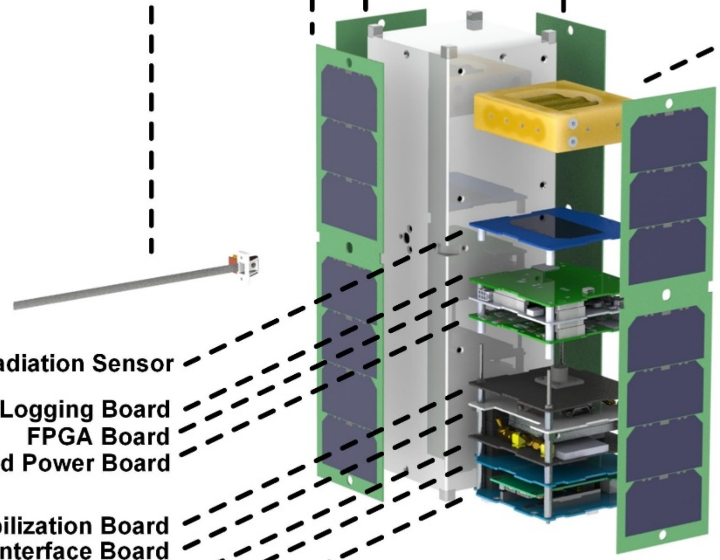
FPGA Experiment Stack



9-Tile MicroBlaze System
(TMR + 6 Spares)

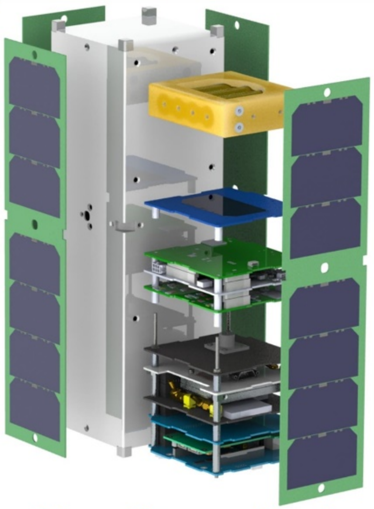
Integrated with Avionics into 3U Satellite



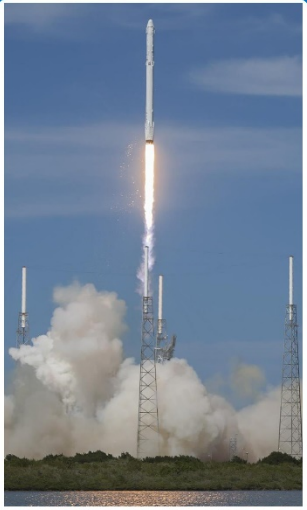


Mission Concept

Use ISS-based, NanoRacks CubeSat Deployer



3U satellite carrying the computer technology (RadSat)



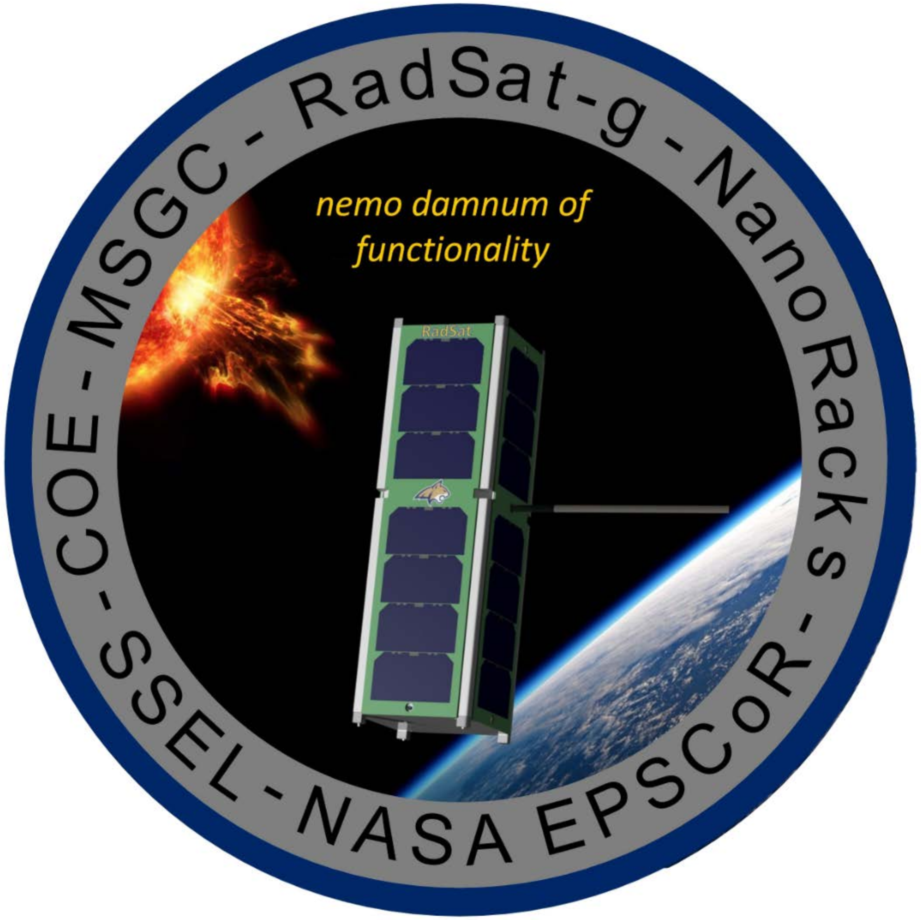
RadSat travels to ISS on resupply mission

RadSat put into LEO using NanoRacks CubeSat Deployer

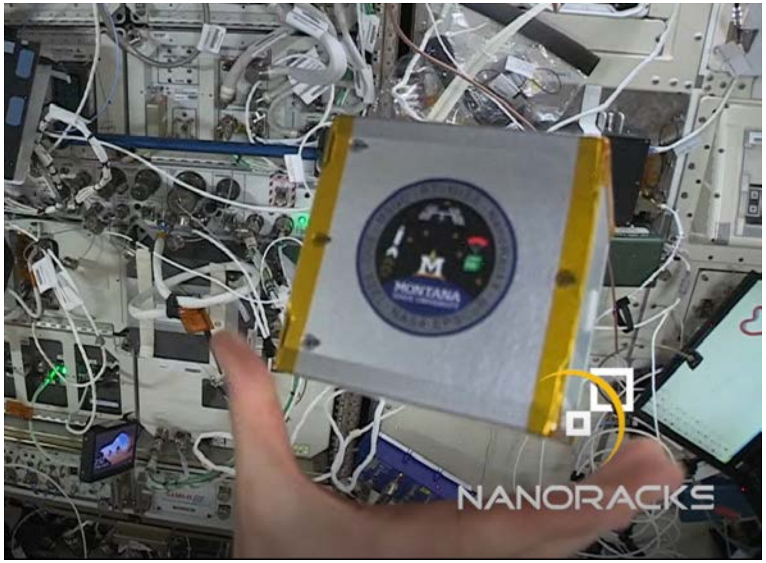
RadSat operated for ~12 months using existing MSU ground station

- **Manifested on ELaNa-23, OA-9 CRS Mission.**
- **Cygnus/Antares II flight out of Wallops Flight Facility.**
- **March 14, 2017.**
- **Operated from SSEL Ground Station in Bozeman, MT.**

Questions


COE - MSGC - RadSat-g - NanoRacks - SSEL - NASA EPSCoR -
nemo damnum of
functionality
RadSat


ANY TIME
ANY PLACE
MONTE!


NANORACKS