Center for Reliable Computing

COTS Technology & Issues - Space Environments

Philip P. Shirvani

Center for Reliable Computing
Stanford University

44th Meeting of IFIP Working Group 10.4
June 28, 2003

Acknowledgments

- Prof. Edward J. McCluskey
- Dr. Nahmsuk Oh
- Naval Research Laboratory (NRL)
- Funding: BMDO/IST, NASA

The Challenge – Stanford CRC ARGOS Project

- **Determine to what extent commercial electronics, e.g., microprocessors and RAMs, can be used in space**

The Approach

1. Design an Experiment to Collect Data
   - Actual satellite
   - Compare rad-hard and COTS boards
   - Evaluate fault tolerance (FT) techniques
2. Develop Techniques for Fault Tolerance
   - Software based – no special hardware
     - EDDI, CFCSS
     - Software-implemented EDAC
3. Develop a Method
   - Estimate distribution of errors
     - In various functional units

Outline

- Motivation and Background
- ARGOS Space Experiment Setup
- Software-Implemented Hardware Fault Tolerance
- Experiment Results
- Conclusion

Motivation

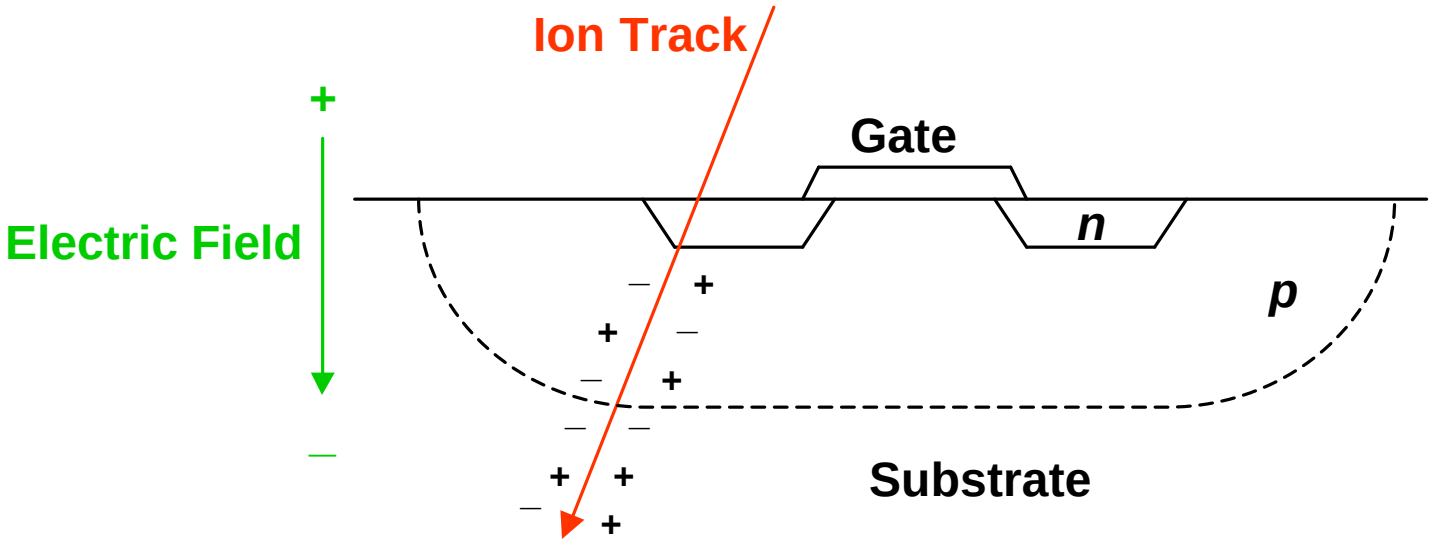
- Reliable Computing in Space
  - Failures Caused by Radiation
    - e.g., Single-Event Upsets (SEUs)
- Problems
  - Costly classical solutions
    - Hardware duplication
    - Radiation-hardening
  - Increasing sensitivity to radiation
    - Deep submicron technologies

Radiation Sources

- Sources in Space
  - Radiation belts
    - Particles trapped in Earth's magnetic field
  - Solar winds
  - Galactic cosmic rays
- Sources on Earth
  - α-particles from radioactive material
  - Secondary particles from cosmic rays
  - Thermal neutrons

Radiation-Matter Interaction

- **Electronic Charge Displacement (Ionization)**
  - **Electron-hole pair production**
    - **Short current pulse (causing an SEU)**
    - **Trapped holes in dielectrics**



Radiation Effects

- Cumulative Long-Term Degradation
  - Total Ionizing Dose (TID)
  - Displacement Damage Dose (DDD)
- Single-Event Effects (SEEs)
  - Single incident ionizing particle
  - Permanent or transient effects
  - Global or local effects
  - e.g., Single-Event Upsets (SEUs)
    - Soft errors, soft fails, transients

Mitigating Radiation Effects

- Fault Avoidance
  - Shielding
    - Heavy, large volume
  - Radiation hardening
    - Expensive, limited availability, old designs
- Fault Tolerance
  - Redundancy (hardware or software)
    - Overhead: price, performance, power, …

Problem Definition

- Computation in Radiation Environments
  - Without customized or rad-hard components

Solutions

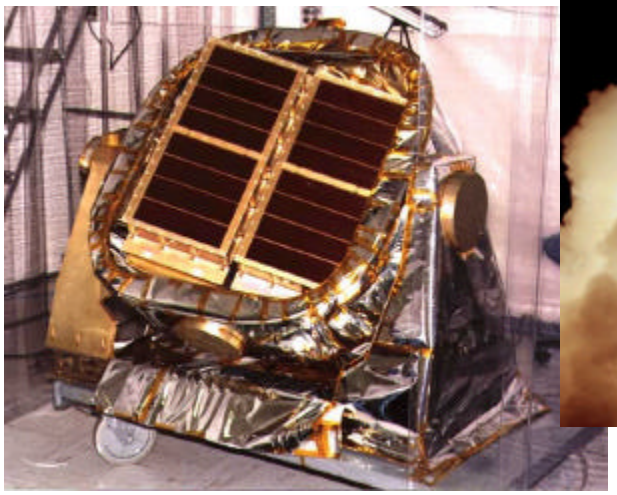
- Commercial Off-The-Shelf (COTS) Components
  - Relatively cheaper and higher performance
- Software based FT Techniques
  - Reliability improvement for COTS

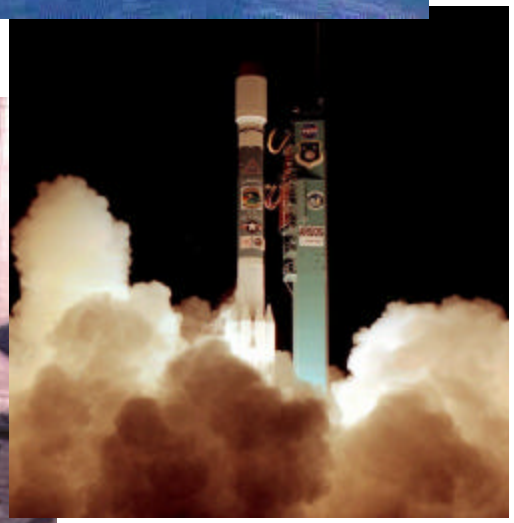
Outline

- Motivation and Background
- ARGOS Space Experiment Setup
- Software-Implemented Hardware Fault Tolerance
- Experiment Results
- Conclusion

Advanced Research and Global Observation Satellite

- Launch: Feb. 23, 1999
- Polar LEO orbit
  - 800 km Altitude, Sun Synchronous, 98° Inclination
- 9 Experiments
  - Including USA (Unconventional Stellar Aspect) experiment of NRL
    - Computing testbed





The ARGOS Project – Computing Testbed

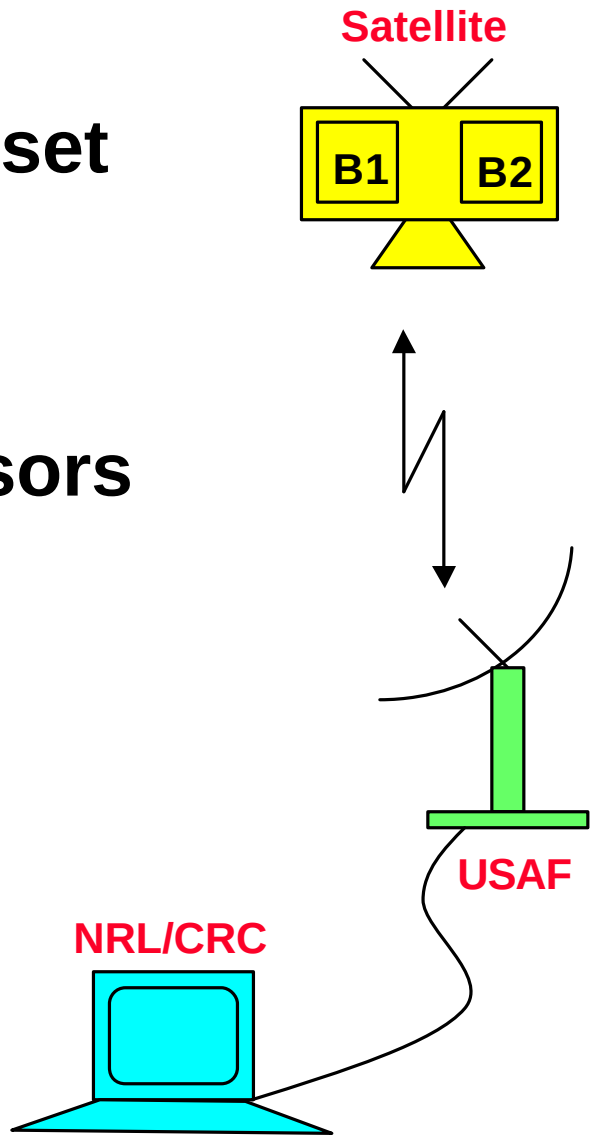
- Reliable Computing in Space
  - Autonomous navigation and data processing
- Goals
  - Comparison of rad-hard & COTS components
  - Evaluation of software-based FT techniques
  - Collection of error data
    - From a real space experiment
    - No simulation or fault injection

Previous Work

- Ground Testing
  - Artificial fault injection
- Space Testing
  - University of Surrey [Underwood 98]
    - COTS SRAMs for micro-satellites
  - MPTB [Dale 95]
    - RAMs, microprocessor, photonic devices
  - Hiten Satellite [Takano 96]
    - Hardware FT technique

Computing Testbed

- Hard Board
  - Harris RH3000 rad-hard chip set
  - SOI SRAMs
  - Hardware FT techniques
    - Self-checking pair processors
    - EDAC for memory
- COTS Board
  - IDT R3081
  - No error detection hardware
    - No EDAC
  - Only software FT techniques



Outline

- Motivation and Background
- ARGOS Space Experiment Setup
- Software-Implemented Hardware Fault Tolerance
- Experiment Results
- Conclusion

Software-Implemented Hardware Fault Tolerance (SIHFT)

- Software-Implemented EDAC [Shirvani 00]
  - Error Detection And Correction (EDAC)
  - SEU protection for main memory
- Error Detection by Duplicated Instructions (EDDI) [Oh 02-1]
- Control Flow Checking by Software Signatures (CFCSS) [Oh 02-2]
- Software-Implemented Error Recovery

Software-Implemented EDAC

- Intercepting all Reads and Writes
  - Infeasible in software
- Periodic Scrubbing
  - Scrubbing:
    - Reading out memory and correcting errors
  - Periodic:
    - e.g., every 30 seconds

☞ Limited to memory blocks with fixed contents:

- Code segments
- Read-only data segments

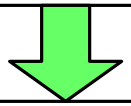
Self-Repair for EDAC Software

- Issue
  - SEU in code segment of EDAC software
    - Cannot repair itself
- Solution
  - Cross-Checking Pair
    - Each copy scrubs the other one
    - Assuming single error

Error Detection by Duplicated Instructions (EDDI)

- Duplicate Instructions
  - Master and shadow instructions
- Compare Master and Shadow Results
  - Detect transient errors in computations

```
ADD R3, R1, R2                ; R3 <- R1 + R2
MUL R4, R3, R5                ; R4 <- R3 * R5
ST  0(SP), R4                 ; store R4 in location pointed by SP
```

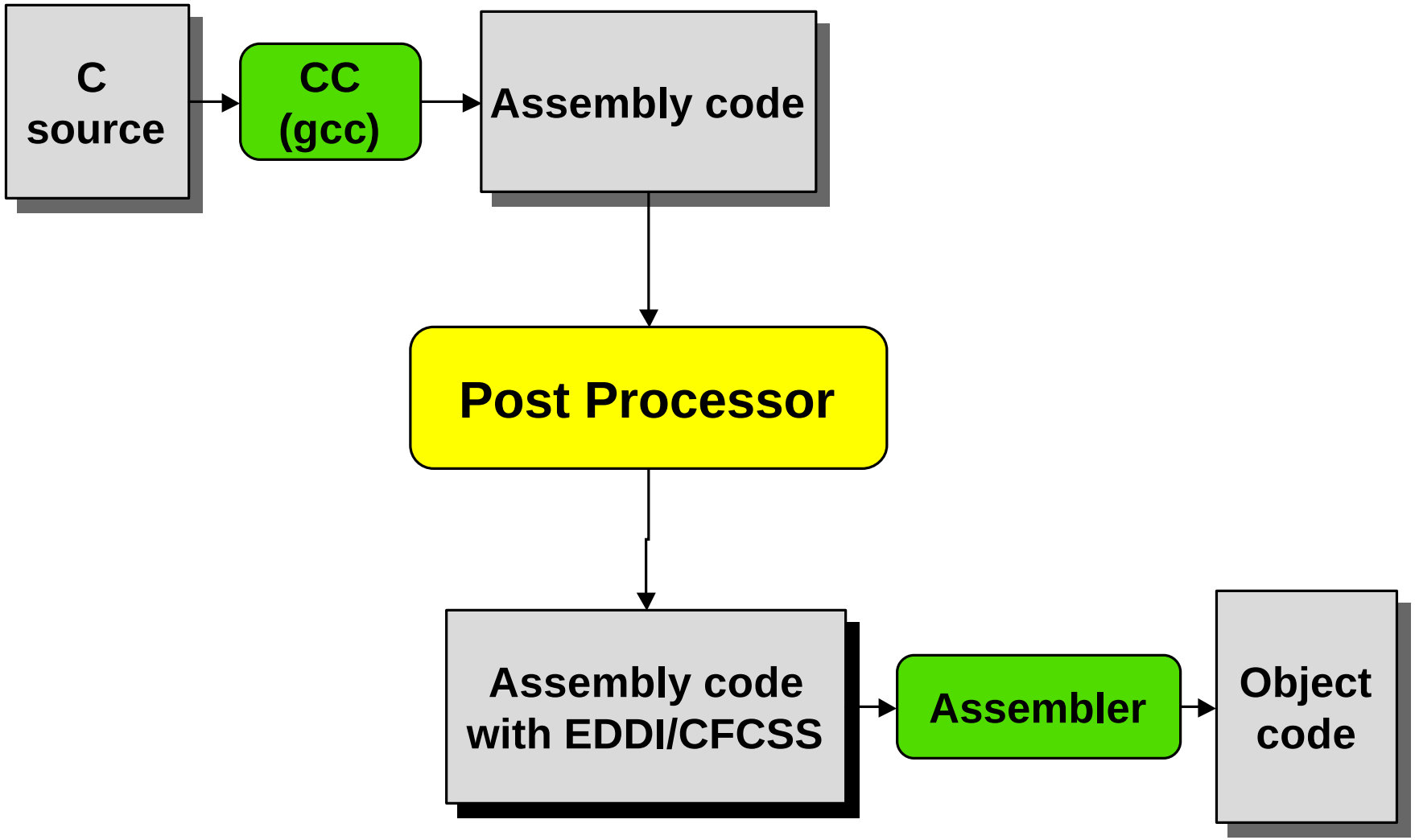


```
ADD R3, R1, R2                ; R3 <- R1 + R2        master
ADD R23, R21, R22             ; R23 <- R21 + R22     shadow
MUL R4, R3, R5                ; R4 <- R3 * R5        master
MUL R24, R23, R25             ; R24 <- R23 * R25     shadow
BNE R4, R24, ErrorHandler     ; compare master and shadow results
ST  0(SP), R4                 ; store master result
ST  offset(SP), R24           ; store shadow result
```

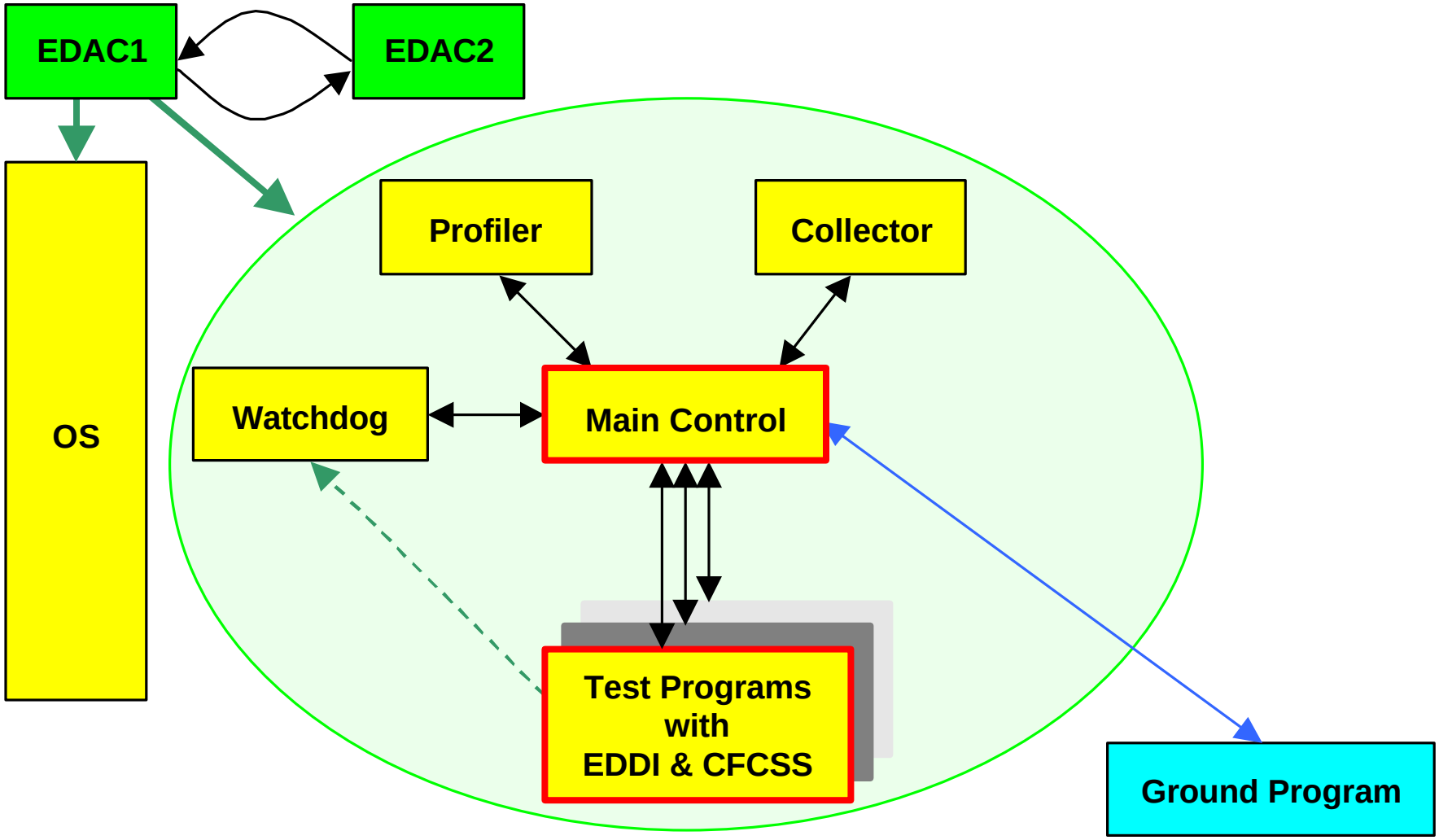
Control Flow Checking by Software Signatures (CFCSS)

- Assigned Signature Analysis Method
  - Unique signature for each basic block
- Interblock Control Flow Checking
  - Correct sequence of blocks followed
- Signature Comparison
  - Pure software
  - No extra hardware

Flow for Adding EDDI and CFCSS



COTS Board – Error Collection & Recovery Software



Software Modules

- Main Control
  - Overall coordination
- Watchdog Timer
  - Detect hang-ups
- Profiler
  - Measure each task’s CPU time
- Collector
  - Store information about errors
- Cross-Checking EDAC Pair
  - Detect and correct memory errors

Error Recovery

- Goal
  - Automatic recovery
    - Without assistance from ground station
- Mechanism
  - Separate task for each module (multitasking)
    - Independent contexts
- Steps
  1. Error is detected
  2. The erroneous task is terminated
  3. Code segment of task is checked by EDAC
  4. The task is restarted

Outline

- Motivation and Background
- ARGOS Space Experiment Setup
- Software-Implemented Hardware Fault Tolerance
- Experiment Results
- Conclusion

Hard Board – Test Programs

- Memory Test
  - Write a pattern in a block of memory
  - Loop
    - Read back and check for correct pattern
- Sine Table Generation
  - Load table
  - Loop
    - Calculate a sine table entry
    - Compare with entry in table

Hard Board – Experiment Results

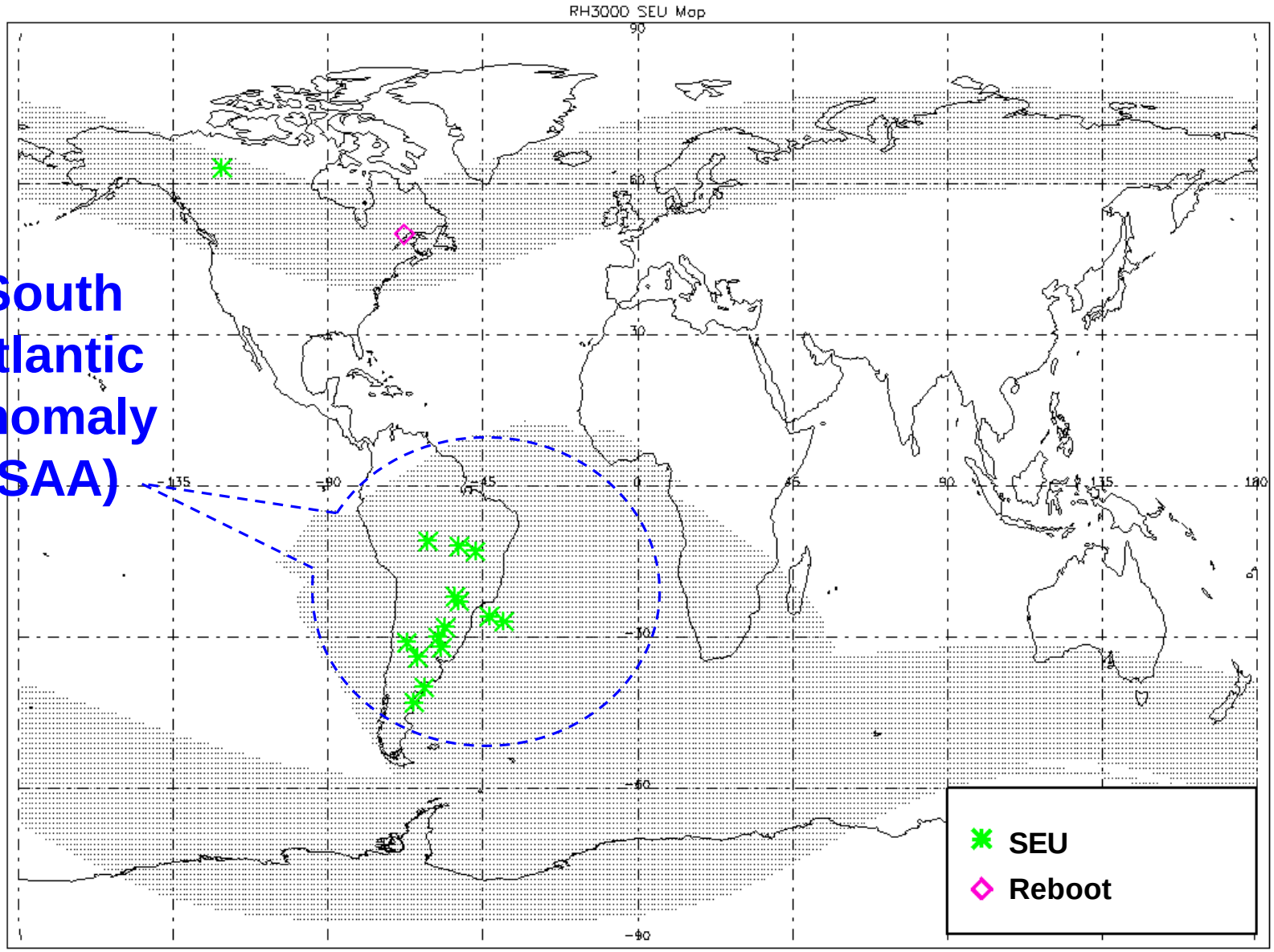
Program	*Data Size (KB)*	*Running Period (days)*	*Num. of Errors*
Memory Test	256	140	4
	512	349	4
Sine Table	128	191	3
	512	250	9

- **All Errors Detected by Software**
- **No Parity or EDAC Errors**
- **Agreement in Self-Checking Pairs**

⇒ **Suspected Source of Errors**

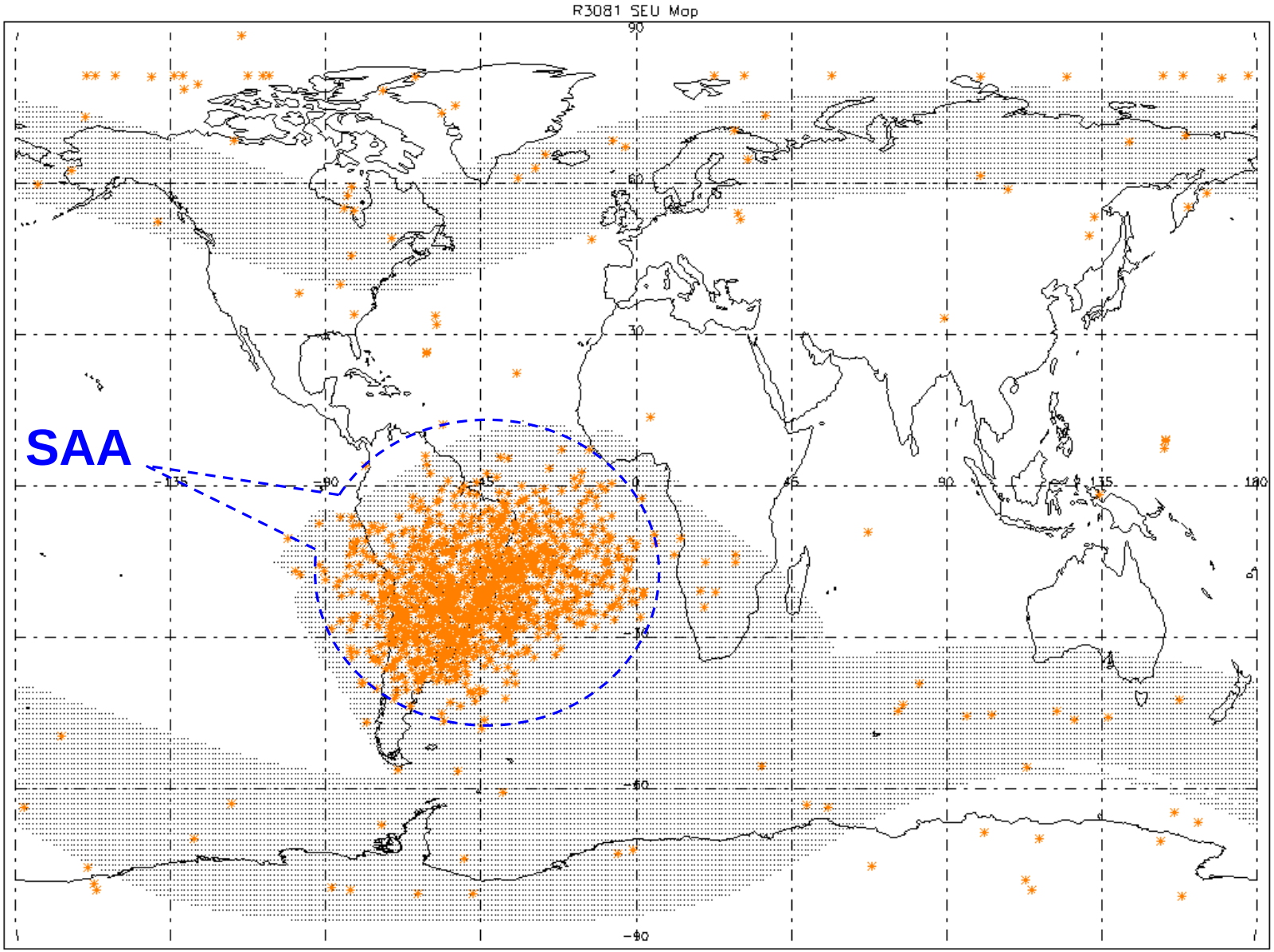
  - **Shared components, e.g., data buffers**

Hard Board SEU Map



South Atlantic Anomaly (SAA)

COTS Board SEU Map



COTS Board – Experiment Results

- Mostly Memory SEUs
  - 5.55 SEUs/MByte per day
- Software-Implemented EDAC
  - Protected memory size: 450KByte
  - Running time: 329 days
  - Errors Detected and Corrected: 831
- Reliability Improvement
  - Time to crash:
    - 2 days without software EDAC
    - 20 days with software EDAC

Memory SEUs

- **Fixed Pattern Test: 55 hex = 01010101 binary**

Data Size	*Running Period*	*Num. of Errors*	*SEU / MB per Day*
512KB cached	18 days	41	4.56
256KB cached	36 days	48	5.33
128KB cached	84 days	54	5.14
128KB non-cached	84 days	59	5.62

- **Overall Results**
  - **Average SER = 5.55 SEUs/MB per day**
  - **MBUs: 1.44%**

Pattern Sensitivity of SEUs

Pattern (hex)	Errors		0 to 1 bit-flips		1 to 0 bit-flips	
	Num	%	Num	%	Num	%
00,00,00,00,…	45	15.0	45	100.0	0	0.0
FF,FF,FF,FF,…	45	15.0	0	0.0	45	100.0
00,FF,00,FF,…	34	11.3	12	35.3	22	64.7
FF,00,FF,00,…	36	12.0	15	41.7	21	58.3
AA,AA,AA,AA,…	43	14.3	23	53.5	20	46.5
AA,55,AA,55,…	52	17.3	23	44.2	29	55.8
55,55,55,55,…	46	15.3	17	37.0	29	63.0
Total	301	100.0	135	44.9	166	55.1

COTS Board – Error Detection Coverage

- Test Programs
  - Insert sort, Quick sort, and FFT
- Error Detection Techniques
  - EDDI + CFCSS + Watchdog Timer
- Checking for Undetected Errors
  - Sort check
  - Checksum of FFT results

Error Detection Coverage

Test Program	Num. of Errors	Errors Detected			Undetected Errors
		EDDI	CFCSS	Watchdog Timer	
Insert Sort – Int.	156	156	–	–	–
Insert Sort – FP	21	21	–	–	–
Quick Sort – Int.	43	31	5	6	1
FFT – FP	102	99	1	2	–
Total	322	307	6	8	1

- **99.7% Detection Coverage**
- **98.8% Successful Recovery**

Throughput Comparison

- Hard Board
  - 10 MHz, no cache memory
- COTS Board
  - 25 MHz, 4KB I-cache, 16KB D-cache
    - 25 times faster without SIHFT
  - SIHFT Overhead
    - EDDI & CFCSS: 170%
    - Software EDAC: 3%
  - One order of magnitude faster

Outline

- Motivation and Background
- ARGOS Space Experiment Setup
- Software-Implemented Hardware Fault Tolerance
- Experiment Results
- Conclusion

ARGOS Conclusions

- Rad-Hard Board
  - Failures despite all hardware FT techniques
    - Single points of failure
- COTS Board
  - Effective software FT techniques
    - Error detection, correction and recovery
- COTS + SIHFT
  - Viable techniques
    - Low radiation environments (such as LEO)

The Challenge

- **Determine to what extent commercial electronics, e.g., microprocessors and RAMs, can be used in space**

The Answer

- **COTS + SIHFT**
  - **Viable for low radiation environments**

Demonstration

- **Successful operation of COTS + SIHFT in ARGOS**
  - **Inspite of 5.55 SEUs/MByte per day**

References

- [Dale 95] Dale, C.J., et al., “Fiber Optic Data Bus Space Experiment on Board the Microelectronics and Photonics Test Bed (MPTB),” *Proc. of the SPIE – The Intr’l Society for Optical Eng.*, Vol. 2482, pp. 285-293, April 1995.
- [Oh 02-1] Oh, N., P.P. Shirvani and E.J. McCluskey, "Error Detection by Duplicated Instructions In Super-scalar Processors," *IEEE Trans. on Reliability*, Vol. 51-1, pp. 63-75, Mar. 2002.
- [Oh 02-2] Oh, N., P.P. Shirvani and E.J. McCluskey, "Control Flow Checking by Software Signatures," *IEEE Trans. on Reliability*, Vol. 51-1, pp. 111-122, Mar. 2002.
- [Shirvani 00] Shirvani, P.P., N. Saxena and E.J. McCluskey, "Software-Implemented EDAC Protection Against SEUs," *IEEE Trans. on Reliability*, Special Section on Fault-Tolerant VLSI Systems, Sep. 2000.
- [Takano 96] Takano, T., et al., “In-orbit experiment on the fault-tolerant space computer aboard the satellite Hiten,” *IEEE Trans. on Reliability*, Vol.45, No. 4, pp. 624-631, Dec. 1996.
- [Underwood 98] Underwood, C.I., “The Single-Event-Effect Behavior of Commercial-Off-The-Shelf Memory Devices – A Decade in Low-Earth Orbit,” *IEEE Trans. Nucl. Sci.*, Vol. 45, No. 6, pp. 1450-1457, Dec. 1998.

Publications

- **Shirvani, P.P., “Fault-Tolerant Computing for Radiation Environments," *CRC-TR 01-6* (Ph.D. Thesis), Stanford University, Stanford, CA, June, 2001.**
- **Shirvani, P.P., N. Oh, E.J. McCluskey, D. Wood and K.S. Wood, "Software-Implemented Hardware Fault Tolerance Experiments; COTS in Space," *Proc. International Conference on Dependable Systems and Networks* (FTCS-30 and DCCA-8), Fast Abstracts, pp. B56-7, New York, NY, June 25-28, 2000.**
- **Shirvani, P.P., and E.J. McCluskey, "Fault-Tolerant Systems in a Space Environment: The CRC ARGOS Project," *CRC-TR 98-2* (CSL TR No. 98-774), Stanford University, Stanford, CA, December 1998.**
- **Lovellette, M.N., K.S. Wood K.S., D.L. Wood, J.H. Beall, P.P. Shirvani, N. Oh, E.J. McCluskey*, “*Strategies for fault-tolerant, space-based computing: Lessons learned from the ARGOS testbed” *Proc. Aerospace Conf.*, 2002. IEEE , Volume: 5 , pp. 2109-19, 2002.**

Acronyms

ARGOS	Advanced Research and Global Observation Satellite
CFCSS	Control Flow Checking by Software Signatures
COTS	Commercial Off-The-Shelf
EDAC	Error Detection And Correction
EDDI	Error Detection by Duplicated Instructions
FT	Fault Tolerance
LEO	Low Earth Orbit
MBU	Multiple-Bit Upset
OS	Operating System
SAA	South Atlantic Anomaly
SEU	Single Event Upset
SIHFT	Software-Implemented Hardware Fault Tolerance