The CRC ARGOS Project: Results

Nahmsuk Oh, Philip P. Shirvani, Edward J. McCluskey
Center for Reliable Computing, Stanford University

Daniel L. Wood
Praxis, Inc.

Michael N. Lovellette, Kent S. Wood
Naval Research Laboratory

REE University Kickoff Meeting
Pasadena, CA, Sep. 6-7, 2000

Outline

- Error Detection Techniques in ARGOS
  - Software-implemented error detection
- Space Experiment Results
- Conclusion
Software-Implemented Error Detection

● Time Redundancy
  ○ Error Detection by Duplicated Instructions (EDDI)
● Control Flow Checking
  ○ Control Flow Checking by Software Signatures (CFCSS)
● Compiler Support
● Software-Implemented EDAC

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

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, ErrHandler ; compare master & shadow results
ST 0(SP), R4 ; store master result
ST offset(SP), R24 ; store shadow result
```
Previous Work

- Redundant operation in VLIW [Blough 92]
  - Hardware approach
  - Error detection by redundant operation in VLIW
- Replicated instructions [Holm & Banerjee 92]
  - Error detection by replicated computation
  - Suitable for VLIW
  - Source register corruption undetected
- Stutter step mode execution [Shirvani 98]
  - Extended to EDDI

Overhead Reduction by Instruction Level Parallelism (ILP)

- Irredundant Programs
  - Limited ILP
  - Idle resources
- Duplicated Instructions
  - No dependency between:
    - Master & shadow instructions
  - More ILP
  - Use idle resources
The CRC ARGOS Project: Results

CFCSS

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

Previous Work

- Structural Integrity Checking (SIC) [Lu 82]
  - Send SIC label to watchdog
- Path Signature Analysis [Namjoo 82]
  - Derived signature for basic block or path
- Signatured Instruction Stream [Shen 83]
  - Cyclic code signature
- Watchdog & Signature analysis [Mahmood 85]
  - Error coverage analyzed mathematically
- Watchdog task [Ersoz 85]
  - Assertion
  - Multitasking OS
- Continuous Signature Monitoring [Wilken 90]
  - Horizontal signature for reducing latency
Software Tool Flow

C source code $\rightarrow$ gcc $\rightarrow$ Assembly code $\rightarrow$ Post processor $\rightarrow$ Assembly code w/ EDDI/CFCSS $\rightarrow$ asm $\rightarrow$ Object code

Add EDDI / CFCSS

Software-Implemented EDAC

- ARGOS Project
  - Continuous error collection (ECC and others)
  - Automatic recovery
- Code Segments
  - Periodic scrubbing
- Data Segments
  - Per request
- Interleaving
  - Correct multiple errors
    - Special attention to multiple-bit upsets (MBUs)
Results: Hard Board

- Special Test Programs
  - Check for undetected errors (Assertion)
  - Memory test, Sine table
- 9 Errors Detected in 336 Days

<table>
<thead>
<tr>
<th>Error Detection Technique</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assertion</td>
<td></td>
</tr>
<tr>
<td>Memory test</td>
<td>6</td>
</tr>
<tr>
<td>Sine table</td>
<td>3</td>
</tr>
<tr>
<td>Hardware</td>
<td></td>
</tr>
<tr>
<td>Bus parity</td>
<td>0</td>
</tr>
<tr>
<td>Main memory EDAC</td>
<td>0</td>
</tr>
<tr>
<td>Self-checking pairs</td>
<td>0</td>
</tr>
</tbody>
</table>

Results: COTS Board (1)

- Application Programs
  - Sample algorithms
  - Quick sort, Insert sort, FFT
- EDDI + CFCSS + Watchdog
- Check for Undetected Errors (Assertion)
  - Sort check
  - Checksum for FFT results
Results: COTS Board (2)

- 203 Errors in 136 Days
  - Shorter period than Hard board
  - Due to lower availability

<table>
<thead>
<tr>
<th>Error Detection Technique</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDDI</td>
<td>195</td>
</tr>
<tr>
<td>CFCSS</td>
<td>3</td>
</tr>
<tr>
<td>Watchdog timer only</td>
<td>4</td>
</tr>
<tr>
<td>Assertion only</td>
<td>1</td>
</tr>
</tbody>
</table>

Recovery in COTS Board

- Application Programs
  - Report errors to main control program
- Main Control Program
  - Call Diagnostic function
  - Call Software EDAC to correct a bit flip: in code segment
  - Kill the corrupted process & restart it
- More Than 95% Successful
The CRC ARGOS Project: Results

Results: Software-Implemented EDAC

- 638 Corrections in 183 Days
  - Errors in OS and control program code
  - No uncorrectable errors reported
  - ~3% MBUs
  - OS code segment error rate ~3 SEUs/day

- Availability Improvement
  - At least one order of magnitude

Conclusions

- Rad-Hard Board
  - Failures despite all hardware FT techniques
  - Single points of failure
- Error Detection, Correction and Recovery
  - Effective software techniques
- COTS + SIHFT
  - Viable techniques
  - Low radiation environments
References