Software for Error Detection in ARGOS

Overview
- Center for Reliable Computing (CRC)
  - Fault tolerance & Adaptive Computing
  - Testing
  - Design & synthesis for testability
- CRC in ARGOS
  - Real space environment experiment
  - Rad-hard v.s. COTS hardware
  - Only software techniques in COTS board

Motivation
- Compare Rad-Hard & COTS Components
- Investigate Software Error Detection Technique
- Correct error data

Goals
- Error Data Collection
  - Processor & memory errors in space
  - Compare different orbital positions
  - Software error detection techniques
- Error Recovery Techniques

ARGOS Experiment
ARGOS satellite
- Sun-synchronous
- 450 nautical miles from earth
- Mission life of 3 years
Software for Error Detection in ARGOS

Rad-Hard Board (RH3000)

- CPU
- FPA
- MD
- I-Cache Bus
- D-Cache Bus
- I-RHSC
- D-RHSC
- I-Cache 32KB
- D-Cache 32KB
- RAM 2 MB
- ROM 128 KB

COTS Board (IDT R3081)

- Processor
- CPU core
- RPU
- FPGAs
- RAM 2 MB
- ROM 256 KB
- Dual Port RAM 8K * 16

Requirements

- Error Detection
  - Exercise functional units
  - Programs with error detection capability
- Fault Tolerance
  - Log errors safely
- Correct Transmission of Error Log

Status

- Launched on Feb. 23
- Software Development
  - Testing on brass board from remote site
  - Uploading scheduled for late April

Software Error Detection

- Transient Error Detection
- Enhance Error Detection in Software
  - Assertions
- Algorithm-Based Fault Tolerance (ABFT)
- Control Flow Checking (SAI)
- Software TMR
- Error Detecting Instructions
- Error Recovery in Logging Error Data
Software for Error Detection in ARGOS

Assertions
- Insert Assertion statement
- Example
  If ....
  else if...
  else if...
  else
  Assert_Not_Reached();

Software Error Detection
- Assertions
  - Algorithm-Based Fault Tolerance (ABFT)
  - Control Flow Checking (SAI)
  - Software TMR
  - Error Detecting Instructions

Algorithm Based Fault Tolerance for FFT

\[
X = A_n x \\
rv = (W_3^{BR(0)}, W_3^{BR(1)}, ..., W_3^{BR(N-1)}) \\
ws = rv \times An \\
X(N) = ws \times x' = (rw^*An)^*x' = rw^*(An\times x') = rw^*X'
\]


Software Error Detection
- Assertions
- Algorithm-Based Fault Tolerance (ABFT)
- Control Flow Checking (SAI)
- Software TMR
- Error Detecting Instructions

Control Flow Checking
Software for Error Detection in ARGOS

Signature Analysis by Instructions
- Assigned Signature Method
- Unique signature for each basic block
- Interblock Control Flow Checking
- Correct sequence followed
- Signature Comparison
- Instructions vs. hardware

Software Error Detection
- Assertions
- Algorithm-Based Fault Tolerance (ABFT)
- Control Flow Checking (SAI)
- Software TMR
- Error Detecting Instructions

Software TMR
- Run three processes in parallel
- Round-Robin scheduling

Error Detecting Instruction (EDI)
- Shadow instruction
- Duplicated instruction
- Comparison instruction
- Compare master & shadow results
- Signature analysis instruction
- Control flow checking between basic blocks

Example
- Original source code
  ```assembly
  ADD R3, R1, R2 ; R3 <- R1 + R2.
  ```
- Add error detecting instruction
  ```assembly
  ADD R3, R1, R2 ;master instruction
  ADD R23, R21, R22 ;shadow instruction
  BNE R3,R23,gotoError ;comparison instruction
  ```
Software for Error Detection in ARGOS

Comparison Point
- Right before store instruction
- Example

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD R3, R1, R2 ;R3 &lt;- R1 + R2</td>
<td></td>
</tr>
<tr>
<td>MUL R4, R1, R3 ;R4 &lt;- R1 * R3</td>
<td></td>
</tr>
<tr>
<td>ST 0(SP), R4 ;store R4 where SP points to.</td>
<td></td>
</tr>
</tbody>
</table>

ADD R3, R1, R2 ;an error corrupts addition here
MUL R4, R1, R3 ;R3 contains incorrect value
ADD R24, R21, R23 ;R4 != R24
BNE R4, R24, gotoError ;branch to error handler

Storage Organization
- Grouped
  - Preserve the data structure of original source code
  - Easy address calculation
- Interleaved
  - Useful for passing parameters between caller and callee
  - Variable number of arguments

LZW Wrong Result Example
- Correct
  - Compressed: a b c d
  - Expanded: abcd abcd abcd abcd

- Incorrect result
  - Fault location: 2163 / 7680
  - Bit location: 2 / 8

Post Processor for EDI
- Add EDI to an assembly program
- Targeted faults
  - Bit flip in the memory (Code & Data)
  - Transient errors in hardware

Flow for Adding EDI
Software for Error Detection in ARGOS

Undetected Faults (with Wrong Results)

Superscalar v.s. Non Superscalar

Execution time overhead

Summary
- ARGOS experiment
- Collect data in space
- Pure software techniques for error detection
- Compare hardware & software schemes
- Rad-Hardened & COTS components