ACS Implementation of A Second-Order Linear Control Algorithm

Catherine Shu-Yi Yu
Stanford University
March 1, 1999

Outline
- Second-Order Linear Control System
- Fault Tolerance
  - Multi-threading
  - System diagram
- Emulation Result
- Fault Injection
- Different Number of Bits
- Retry
- Future Work

Second-Order Linear System

Fault Tolerance
- Original
- Multithreading

System Diagram

Emulation Results

<table>
<thead>
<tr>
<th>Threads</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Bits</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td># of Retries</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>CRT's Memory</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Memory</td>
<td>8R3Wx2</td>
<td>12R6Wx2</td>
<td>8R3Wx2+</td>
</tr>
<tr>
<td>Memory Area</td>
<td>25204</td>
<td>90086</td>
<td>88153</td>
</tr>
<tr>
<td>Total Area</td>
<td>54255</td>
<td>61787</td>
<td>123836</td>
</tr>
<tr>
<td># of FPGAs</td>
<td>37</td>
<td>36</td>
<td>85</td>
</tr>
<tr>
<td>f (MHz)</td>
<td>1.6</td>
<td>2.5</td>
<td>--</td>
</tr>
</tbody>
</table>
ACS Implementation of A Second-Order Linear Control Algorithm

**Fault Injection**

- Address Decoder
- Memory Cell Array
- Mem Cell Fault Inject
- XOR
- Output Driver/Decoder

**Simulation Results (I)**

1 thread v.s. 3 threads (p=0.00001)

**Simulation Results (II)**

<table>
<thead>
<tr>
<th>Prob of bit flip</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10⁻³</td>
</tr>
<tr>
<td>1 thread</td>
<td>98.9%</td>
</tr>
<tr>
<td>3 threads tradition voting</td>
<td>98.9%</td>
</tr>
<tr>
<td>3 threads median voting</td>
<td>98.9%</td>
</tr>
</tbody>
</table>

**Different Number of Bits**

- Problem: Overflow
- Solution
  - Different decimal point position
  - Changing parameters

**Results**

<table>
<thead>
<tr>
<th>Bits</th>
<th>32</th>
<th>24</th>
<th>16</th>
<th>12</th>
<th>8</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fraction bits Robot</td>
<td>16</td>
<td>12</td>
<td>8</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td>16</td>
<td>12</td>
<td>9</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overshoot</td>
<td>43%</td>
<td>42%</td>
<td>41%</td>
<td>47%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steady state error</td>
<td>1.0%</td>
<td>1.0%</td>
<td>2.9%</td>
<td>3.6%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stable time (cycles)</td>
<td>&lt;5%</td>
<td>15</td>
<td>15</td>
<td>31</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>Response time (cycles) (90%)</td>
<td>27</td>
<td>27</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Retry**

- Mem
- Mem
- Mem
- Output
- Control
- Error
Retry
- Assume Single Error
- When One Thread Has An Error at Nth cycle
  - Use other thread's stored value after Nth cycle
  - If still wrong
    - Diagnosis
    - Reconfiguration

Summary
- Multithreading
- Performance
- Fault tolerance – 1e-4 < p < 1e-5
- Different Number of Bits
  - For higher number of bits
    - Less response time
    - More overshoot
    - Smaller steady state error
  - From QuickTurn's Synthesis: number of FPGAs not significantly reduced

Future Work
- Adaptive Recovery Techniques
  - Retry
  - Dynamic reconfiguration
- Experiments
- Automating Scheduling
- More Control Algorithms