**FATAL+HEX: Fault-tolerant Self-Stabilizing Clock Generation+Distribution**

**End Goal:**
- Highly dependable architecture

**Self-Stabilization**

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure</td>
<td>Stabilization</td>
<td>Stable</td>
<td></td>
</tr>
</tbody>
</table>

**masks transient Byzantine faults**

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable</td>
<td>Failure</td>
<td>Stabilization</td>
<td>Stable</td>
</tr>
</tbody>
</table>

**Byzantine fault-tolerance**

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Nodes 1-3 are stable, Node 4 is faulty</td>
</tr>
</tbody>
</table>

**FATAL+: Clock Generation**

- Can now generate pulse
- Proposes pulse
- Wait for $P$ neighbors to propose
- Wait for $P$ neighbors to stabilize
- If $j$ propose signal
- If $j$ propose signal
- Communicate topology of FATAL+ core

**HEX: Clock Distribution**

- Pulse forwarding algorithm for HEX-nodes.
- Once receives trigger messages from (left and lower left) or (lower right and lower right) neighbors do broadcast trigger messages; if local clock pulse sleep for same time within $T_x$ - $T_y$
- Forget previously received trigger messages

- Can tolerate one Byzantine fault in each neighborhood.
- Triggers pulse once both neighbors on previous layer have
- If one of them failed, neighbors on same layer can fill in
- Self-stabilizing: directed pulse propagation “flushes out” false residual states from transient faults
- Local oscillators drive high-frequency “fast clocks”
- Resynchronized with every pulse flooded through the grid
- Can be leveraged for fast and efficient communication within a small number of clock cycles

**Future Work**

- Develop novel hardware building blocks to:
  - Increase operational frequency
  - Have cheap self-stabilizing low-level building blocks
  - Bottom-to-top formal verification of FATAL+HEX compound system
  - Provide fault-tolerant communication and application logic
  - Build and test fully functional ASIC prototype

**Acknowledgements**

Danny Dolev
Matthias Fugger
Markus Hofbauer
Christoph Lenzen
Martin Perner
Markus Posch
Ulrich Schmid
Martin Sigl
Andreas Steininger

Hebrew University of Jerusalem
Vienna University of Technology
Massachusetts Institute of Technology
Vienna University of Technology
Vienna University of Technology
Vienna University of Technology
Vienna University of Technology

Fault-tolerant Algorithms for Tick-Generation in Asynchronous Logic: Robust Pulse Generation

Under submission to Journal of the ACM (JACM), first revision.

HEX: Scaling Honeycombs is Easier than Scaling Clock Trees


FATAL+; An Ultra-Robust Clocking Scheme for Systems-on-Chip

Under submission to Journal of Computer and System Sciences (JCSS). Byzantine Self-Stabilizing Clock Distribution with HEX: Implementation, Simulation, Clock Multiplication

6th Conference on Dependability (DEFEND), 2013.

Efficient Construction of Global Time in SoCs despite Arbitrary Faults


Fault-tolerant Algorithms for Tick-Generation in Asynchronous Logic: Robust Pulse Generation