Enabling RTR for Industry

Oliver Diessel & Shannon Koh

Computer Science & Engineering
The University of New South Wales

Embedded, Real-Time and Operating Systems Program
National ICT Australia
What is RTR?

Partial restructuring of a system while it is active

– At what scale?
  Gate/wire/block/module/device/board/system

– When?
  At startup/mode switch (externally or internally triggered)

– How long should it take?
  Limited by device or optimization algorithms
What is the promise of RTR?

• Adaptive hardware
  – Better performance; smaller area; lower power
  – Multifunctional/virtual hardware
  – More with less

• Greater computational power?
How does industry want to use it?

- **Industry’s goal is to maximize profit**
  - Equivalent to *minimizing costs* (development, production, unit delivery, upgrade) and *maximizing sales* (utility, affordability, desirability)

- **How does RTR help?**
  - RTR can potentially help with utility, affordability, upgradability & desirability
    - But these are not easily measurable
  - Low NRE for configurable systems, but what about RTR systems?
    - Design for (most (successful)) RTR is probably still a long way off being commoditized
  - And what about the price/performance niche?
    - When does RTR confer a benefit?
SIREN: Satnav Interference Rejection Engine

• Use emerging GNSS signal diversity to mitigate effects of interference...

• Second generation project to develop open FPGA-based satellite navigation/timing solutions

1 Work led by Andrew Dempster, SNAP@UNSW
Standard Zarlink GPS receiver design
"NAMURU"

Eora: *ngamuru* means ‘to see the way’ or ‘compass’

Navigation Aid Made at UNSW for Reconfiguration Use
Effect of CW interference

Number of visible satellites vs. Time

With interference
Without interference
Figure 1: Galileo Frequency Plan
SIREN concept

• Predict/detect when signals are affected by interference
• Swap compromised channels for ones that are not
• Derive a navigation solution from the resulting set

• Aiming for positioning/timing estimates at 100Hz
• Goal is to integrate techniques into a single chip solution
SIREN RTR challenges

- Develop correlator layouts that can rapidly be reconfigured
- Develop a systems architecture that supports partial dynamic reconfiguration
- Integrate processor-based control and internal reconfiguration into a single-chip solution
- Testing
L1 correlator design
L2c correlator design
L5 correlator design
## Resource estimates

<table>
<thead>
<tr>
<th>Tracking Loop Elements</th>
<th>4-LUTs/FFs Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L1</td>
</tr>
<tr>
<td>Carrier mixer</td>
<td>8</td>
</tr>
<tr>
<td>Code mixer</td>
<td>6</td>
</tr>
<tr>
<td>Carrier NCO</td>
<td>60</td>
</tr>
<tr>
<td>Code NCO</td>
<td>51</td>
</tr>
<tr>
<td>EPL shift register</td>
<td>3</td>
</tr>
<tr>
<td>Code Generator</td>
<td>90</td>
</tr>
<tr>
<td>Integrators</td>
<td>96</td>
</tr>
<tr>
<td>NH code mixer</td>
<td>------</td>
</tr>
<tr>
<td>Sum</td>
<td>320</td>
</tr>
</tbody>
</table>
Reconfiguration delay estimates

<table>
<thead>
<tr>
<th>Configuration Delay (μs)</th>
<th>CPL &lt;&lt; CP</th>
<th>CPL ≈ CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>400</td>
<td>710</td>
</tr>
<tr>
<td>L2</td>
<td>472</td>
<td>787</td>
</tr>
<tr>
<td>L5</td>
<td>787</td>
<td>1,023</td>
</tr>
<tr>
<td>L1 ↔ L2</td>
<td>80</td>
<td>315</td>
</tr>
</tbody>
</table>
The COMMA Approach
Module Placement

- Module slots occupy a V4 "page" 16 CLBs x ½ width
- Slots may be subdivided
- May also be aggregated

1 Shannon Koh’s work
The COMMA Approach

Wiring Harness

- Provides interconnect between module pins and device pins
- Connect modules to wires via slice macros
- Tailored to the application
The COMMA Approach
Pin Virtualisation

- Connect module pins to slice macros via Reconfigurable Data Ports
COMMA Design: Configuration Epochs
COMMA Design: Optimisation
What support does industry need to make more/better use of RTR?

- Motivation to explore the applicability of RTR
- Design exploration tools that allow RTR to be rapidly modelled and assessed at a high level
- Synthesis tools
  - Optimize across configurations
    - Minimize area & power; maximize performance
  - Automate RTR infrastructure provision
    - Minimize overheads at the various levels in the design hierarchy
      - Resource management, communications, controllers, OS, run-time environments
- Validation & verification tools
- Vendor support
What support are they getting?
Where are all the RC companies?
What support are they getting?
Why aren`t they looking at our research?
Can we make a difference?

YES, but we might have to change…

• Grand challenges
  – Companies can’t take the risk to do much explorative research, so academics have an opportunity
  – We may need to aggregate efforts and plan to make an impact

• Collaborate
  – Applications are the driver for innovation in support technology
  – Do one project per year with industry and publish results in trade/industry journals
Can we make a difference?

• Commoditize
  – Don’t stop at developing an algorithm and benchmarking your prototype – take it as far as you can… is it commercializable? What needs to change to make a product out of your design?

• Benchmarks
  – Develop benchmarks that allow improvements to be measured and efforts to be compared

• Vendor support
  – Bridge the gap between vendors and end users
  – Get our enhancements supported by Xilinx/Altera