FPGA Overlays for High Performance Computing

Davor Capalija and Tarek S. Abdelrahman

Department of Electrical and Computer Engineering
University of Toronto
Research Goal

• Design FPGA overlays that deliver high performance for application acceleration
  – **Opportunity**: alleviate the need for hardware design and long synthesis times
  – **Challenge**: deliver high performance that scales with increasing device size

• An enabler for widespread use of FPGAs
  – Rapid prototyping
  – Bridge solutions
  – Portability across FPGA platforms
  – Lower entry threshold
Research Issues

- What programming model to support?
  - Pipelined (streaming) execution of dataflow graphs
Research Issues – Cont’d

- What should the overlay architecture be?

**Overlay Cell**
- the overlay is a 4-NN connected array of cells

**Functional Unit**: ADD, SUB, MUL, DIV, etc.
- realizes a DFG operation

**Pipeline, Routing and Synchronizing logic**
- makes the cell/FU a stage in a pipeline of cells
- establishes connections that realize data flow
• Software tool chain
  – A place-and-route algorithm
Research Issues

• How to maintain frequency as the overlay grows
  – Push-button approach fails
  – A bottom-up tile-based methodology
  – Reusable tiles
Research Issues – Cont’d

• Managing resource overhead
  – How to measure overhead?
  – How to reduce overhead?

• System integration
  – How to attach to a memory system?
  – How to integrate with a processor?
  – How to virtualize in a system?
Prototype Overlays

- Two prototype overlays on a Stratix IV device
  - Each uses about 75% of the device’s resources

<table>
<thead>
<tr>
<th>Overlay</th>
<th>Nodes</th>
<th>Shape</th>
<th>$f_{\text{MAX}}$ (MHz)</th>
<th>Peak Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP</td>
<td>288</td>
<td>18x16</td>
<td>312</td>
<td>32.4 GFLOPS</td>
</tr>
<tr>
<td>INT</td>
<td>384</td>
<td>24x16</td>
<td>355</td>
<td>60.4 GOPS</td>
</tr>
</tbody>
</table>

- Scalable
  - No frequency change from a 5x4 FP overlay to 18x16
  - 7% frequency drop from a 5x4 INT overlay to 24x16

- Programmability overhead over FU-only circuit
  - 10.6X for INT
  - 3.4X for FP
<table>
<thead>
<tr>
<th>DFG</th>
<th>Size</th>
<th>GOPS</th>
<th>Compile Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-Body</td>
<td>44</td>
<td>6.24</td>
<td>0.11</td>
</tr>
<tr>
<td>n-Body (2x)</td>
<td>81</td>
<td>12.48</td>
<td>0.24</td>
</tr>
<tr>
<td>n-Body (4x)</td>
<td>125</td>
<td>18.72</td>
<td>0.44</td>
</tr>
<tr>
<td>BlackSholes</td>
<td>131</td>
<td>21.22</td>
<td>1.33</td>
</tr>
<tr>
<td>MatMul</td>
<td>96</td>
<td>19.66</td>
<td>1.05</td>
</tr>
<tr>
<td>MatMulAdd</td>
<td>114</td>
<td>22.46</td>
<td>3.80</td>
</tr>
<tr>
<td>RGB2YIQ</td>
<td>38</td>
<td>7.46</td>
<td>0.07</td>
</tr>
<tr>
<td>RGB2YIQ (2x)</td>
<td>65</td>
<td>14.91</td>
<td>0.15</td>
</tr>
<tr>
<td>RGB2YIQ (4x)</td>
<td>130</td>
<td>29.82</td>
<td>0.91</td>
</tr>
<tr>
<td>SAD</td>
<td>80</td>
<td>16.69</td>
<td>0.17</td>
</tr>
<tr>
<td>SAD (2x)</td>
<td>132</td>
<td>33.37</td>
<td>0.41</td>
</tr>
<tr>
<td>Gaussian Blur</td>
<td>81</td>
<td>17.40</td>
<td>0.19</td>
</tr>
<tr>
<td>Gaussian Blur (2x)</td>
<td>136</td>
<td>34.79</td>
<td>5.33</td>
</tr>
<tr>
<td>MatMul</td>
<td>96</td>
<td>22.37</td>
<td>2.11</td>
</tr>
<tr>
<td>MatMulAdd</td>
<td>114</td>
<td>25.56</td>
<td>2.32</td>
</tr>
</tbody>
</table>

**GOPS**: Giga Operations Per Second
Summary

• Overlay architectures can enable widespread use of FPGAs in application acceleration

• Two prototypes of one such overlay architecture shows promise

• Many remaining challenges
  – Scaling to million LUT FPGAs
  – The CAD tools
  – FPGA architectural features
  – System integration