Floating Point Synthesis From Model-Based Design

M. Langhammer, M. Jervis, G. Griffiths, M. Santoro
Floating Point On FPGA

- Many new applications require floating point dynamic range performance; e.g.
  - Beam-forming, large FFTs, DPD feedback, ...

- FPGAs can deliver:
  - A powerful mix of fixed and floating point performance
  - Extensive hard DSP capabilities
    - IEEE754 single and double precision specifically supported
      - 100s 36x36 multipliers
      - ~100 54x54 multipliers
  - Superior computational density per Watt than other solutions
  - Sustained peak performance

- Tools Enable:
  - Schematic design entry using operator blocks
  - Fused data-path compilation
DSP Builder / Floating Point Compiler

1. Develop & design schematically with primitive, math.h & core functions; in fixed and floating point domains

2. Tool applies global data-path optimizations to floating point domains and many fixed-point domain optimizations

3. Tool generates integrated and optimized high-performance HDL on each simulation

- Slightly larger, wider operands
- True floating mantissa, not just [1,2)
- Remove Normalization
- Do not apply special and error conditions here
Density and Flexibility

Compiler Advantages

- Fused data-path synthesis allows maximum density at 250MHz system performance
  - No system degradation for floating point
  - 50% device logic resources available for system design
- Over 600 single precision operators can be supported in a mid-range device
- Optimized MATH.H library in development
  - Multiplier based algorithms give low latency, high performance, consistent results
  - EXP, LOG, SQRT, INV SQRT, SIN, COS available now

Floating Point IP

- Matrix Multiplier Core
  - Parameterized
- Example design:
  - 200 single precision operators, 295 MHz fit, 100 GFlops
  - ¼ mid range device
Compiler vs. Cores

- Compiled Data-path Example
  - Matrix Decomposition

- Compiled data-path is about 50% the size of core based design
  - DSP resources same
  - Latency also 50%

<table>
<thead>
<tr>
<th>Matrix Size</th>
<th>Logic</th>
<th>DSP</th>
<th>Vector Logic</th>
<th>Core Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>12x12</td>
<td>5197 (sp)</td>
<td>75</td>
<td>4587 (sp)</td>
<td>7800 (sp)</td>
</tr>
<tr>
<td></td>
<td>8652 (dp)</td>
<td></td>
<td>7855 (dp)</td>
<td>9000 (dp)</td>
</tr>
<tr>
<td>64x64</td>
<td>21457 (sp)</td>
<td>283</td>
<td>20681 (sp)</td>
<td>41600 (sp)</td>
</tr>
<tr>
<td></td>
<td>27346 (dp)</td>
<td></td>
<td>26004 (dp)</td>
<td>48000 (dp)</td>
</tr>
</tbody>
</table>

- Vector Logic:
  - compiled data-path

- Logic:
  - compiled data-path + application

- Core Logic:
  - equivalent data-path constructed from discrete cores