Fast Fourier Transform on FPGA: Design Choices and Evaluation
Peter A. Milder, Franz Franchetti, James C. Hoe, Markus Püschel

**Problem Statement**
- The discrete Fourier transform (DFT) is among the most important tools in signal processing.
- DFT has many algorithms (FFTs) and design choices.
- How to represent, generate, and evaluate the design space for given user constraints?

**Results**: 1) FFT IP core generator: “point and click”

2) FFT implementation guidelines

<table>
<thead>
<tr>
<th>Architecture level</th>
<th>Option</th>
<th>Objective / Optimization</th>
<th>Suggestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithmic</td>
<td>algorithm</td>
<td>variable-latency or cost</td>
<td>Pass FFT</td>
</tr>
<tr>
<td></td>
<td>reduction</td>
<td>variable-latency or cost</td>
<td>Pass FFT</td>
</tr>
<tr>
<td></td>
<td>reduction</td>
<td>fixed by application</td>
<td>fixed by application</td>
</tr>
<tr>
<td></td>
<td>retention</td>
<td>2 or 4</td>
<td>2 or 4</td>
</tr>
</tbody>
</table>

**Algorithmic Level**

- Discrete Fourier Transform (DFT)
  \[ y[n] = DFT_x y \]
  \[ y[n] = \exp\left(\frac{2\pi i n}{M}\right) \]
- Fast Fourier Transform (FFT) Algorithms
  - Matrix factorization
    \[ DFT_3 = \begin{bmatrix} DFT_2 & 0 & 0 \\ 0 & DFT_2 & 0 \\ 0 & 0 & DFT_2 \end{bmatrix} \]
  - Representation as matrix formula
    \[ DFT_2 = \begin{bmatrix} X_0 & Y_0 \\ X_1 & Y_1 \\ X_2 & Y_2 \end{bmatrix} \]
- Pease FFT [2]:
  \[ DFT_3 = \begin{bmatrix} X_0 & Y_0 & 0 \\ X_1 & Y_1 & 0 \\ X_2 & Y_2 & 0 \end{bmatrix} \]
- Iterative FFT [3]:
  \[ DFT_3 = \begin{bmatrix} X_0 & Y_0 & Z_0 \\ X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \end{bmatrix} \]

**Architectural Level**

- Formal View of Streaming
  \[ I_m \otimes A_r \]
  \[ I_m \otimes A_r' \]
  \[ I_m/n \otimes M(1/r \otimes A_r) \]

- FPGA Mapping
  - Stride permutation
    - Method 1: RAM-Based [4]
      - storage
      - proper cost
      - execution time
    - Method 2: FIFO-Based [5]
      - storage
      - proper cost
      - execution time

**Evaluation**

- Synthesis: Xilinx ISE version 8.1
- Spiral generated FFT IP cores vs. Xilinx LogiCore FFT 3.2
- Gap (1/4 throughput) versus area
- Para-optimal points

**Other FPGA-Mapping Options**
- Complex multiplication (2 options)
- Twiddle factor storage (2 options)

**Cost / performance comparable to benchmarks.**

High degree of control over tradeoffs.

**References**

© 2007 Peter A. Milder

This work was supported by DARPA under DOD grant N00014-10-50099 and by NSF awards ACI-0424395 and ITR/ACI-0352687.