A Systolic FFT Architecture for Real Time FPGA Systems

Preston Jackson, Cy Chan, Charles Rader, Jonathan Scalera, and Michael Vai

HPEC 2004

29 September 2004

This work was sponsored by DARPA ATO under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.
Outline

• Introduction
  – Motivation
  – Evaluation metrics

• Parallel architecture

• Systolic architecture

• Performance summary

• Conclusions
Radar Processing Application

\[ \text{Corr}_{x,y}[m] = \sum_{n} x[n]y^*[n-m] \]

- **ADC 1.2 GSPS**
- **Correlation**
- **32K**

8K FFT bottleneck
- Real-time
- Complex
- 0.6 GSPS input (16-bits)
- 1.2 GSPS output (12-bits)
The design changes will be scored based on the following metrics:

- Length of FFT
- IO pins
- Butterflies
- Multipliers
- Adder/subtractors
- Shift registers
Outline

• Introduction

• Parallel architecture
  – Data flow graph
  – Effects of serial input

• Systolic architecture

• Performance summary

• Conclusions
Baseline Parallel Architecture

Parallel FFT
- Butterfly structure
- Removes redundant calculation

<table>
<thead>
<tr>
<th></th>
<th>Size</th>
<th>Pins</th>
<th>Fly</th>
<th>Mult</th>
<th>Add</th>
<th>Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
<td>8192</td>
<td>Δ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pins</td>
<td>448</td>
<td>229K</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fly</td>
<td>32</td>
<td>53K</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shift</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Complex Butterfly

- Butterfly contains
  - 1 complex addition
  - 1 complex subtraction
  - 1 complex, constant multiply
Complex Addition

- Complex addition adds the real and imaginary parts separately:

\[(a + jb) + (c + jd) = (a + c) + j(b + d)\]

2 adds

\[\begin{align*}
\text{Size} & : 16, 8192, \Delta \\
\text{Pins} & : 448, 229K \\
\text{Fly} & : 32, 53K \\
\text{Mult} & : 128, 213K \\
\text{Add} & : 0, 0 \\
\text{Shift} & : 0, 0
\end{align*}\]
Complex Multiply

- The FOIL method of multiplying complex numbers:

\[(a + jb)(c + jd) = (ac - bd) + j(ad + bc)\]

4 multiplies and 2 adds
Another approach requires fewer multiplies:

\[(ad + bc) = c(a + b) - a(c - d)\]
\[(ac - bd) = d(a - b) + a(c - d)\]

3 multiplies and 5 adds
Parallel-Pipelined Architecture

A pipelined version
• IO Bound
• 100% Efficient

<table>
<thead>
<tr>
<th>Size</th>
<th>16</th>
<th>8192</th>
<th>△</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pins</td>
<td>448</td>
<td>229K</td>
<td></td>
</tr>
<tr>
<td>Fly</td>
<td>32</td>
<td>53K</td>
<td></td>
</tr>
<tr>
<td>Mult</td>
<td>96</td>
<td>159K</td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td>288</td>
<td>480K</td>
<td></td>
</tr>
<tr>
<td>Shift</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Systolic Architecture-11
PAJ 9/29/2004

MIT Lincoln Laboratory
Serial Input

A serial version
- IO-rate matches A/D
- 6.25% Efficient

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>16</td>
<td>8192</td>
<td>△</td>
<td></td>
</tr>
<tr>
<td>Pins</td>
<td>28</td>
<td>28</td>
<td>.01%</td>
<td></td>
</tr>
<tr>
<td>Fly</td>
<td>32</td>
<td>53K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mult</td>
<td>96</td>
<td>159K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td>288</td>
<td>480K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shift</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

• Introduction

• Parallel architecture

• Systolic architecture
  – Serial implementation
  – Application specific optimizations

• Performance summary

• Conclusions
Serial Architecture

- The parallel architecture can be collapsed
  - One butterfly per stage
  - Consumes 1 sample per cycle
  - Same latency and throughput
  - More efficient design

<table>
<thead>
<tr>
<th></th>
<th>Size</th>
<th>Pins</th>
<th>Fly</th>
<th>Mult</th>
<th>Add</th>
<th>Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
<td>28</td>
<td>4</td>
<td>12</td>
<td>36</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>28</td>
<td>13</td>
<td>39</td>
<td>117</td>
<td>12K</td>
</tr>
</tbody>
</table>

50% Efficiency
High Level View

- Replace complex structure with an abstract cell which contains:
  - FIFOs
  - Butterfly
  - Switch network

<table>
<thead>
<tr>
<th>Size</th>
<th>16</th>
<th>8192</th>
<th>△</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pins</td>
<td>28</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Fly</td>
<td>4</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Mult</td>
<td>12</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td>36</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td>Shift</td>
<td>22</td>
<td>12K</td>
<td></td>
</tr>
</tbody>
</table>

Stage 1 Stage 2 Stage 3 Stage 4
8192-Point Architecture

- Requires 13 stages
- Fixed point arithmetic
- Varies the dynamic range to increase accuracy
- Overflow replaced with saturated value

- Multipliers limit design to 18-bits and 150 MHz
- Achieves 70 dB of accuracy

<table>
<thead>
<tr>
<th>Size</th>
<th>16</th>
<th>8192</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pins</td>
<td>28</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Fly</td>
<td>4</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Mult</td>
<td>12</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td>36</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td>Shift</td>
<td>22</td>
<td>12K</td>
<td></td>
</tr>
</tbody>
</table>
Increase Parallelism

Add more pipelines
- Design limited to 150 MHz by multipliers
- I/Q module generate 600 MSPS
- Meets real-time requirement through parallelism

<table>
<thead>
<tr>
<th>Size</th>
<th>16</th>
<th>8192</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pins</td>
<td>112</td>
<td>112</td>
<td>400%</td>
</tr>
<tr>
<td>Fly</td>
<td>16</td>
<td>52</td>
<td>400%</td>
</tr>
<tr>
<td>Mult</td>
<td>48</td>
<td>156</td>
<td>400%</td>
</tr>
<tr>
<td>Add</td>
<td>144</td>
<td>468</td>
<td>400%</td>
</tr>
<tr>
<td>Shift</td>
<td>16</td>
<td>12K</td>
<td>100%</td>
</tr>
</tbody>
</table>
Simplification

Target application allows a specific simplification
- Pads a 4096-point sequence with 4096 zeros
- Removes 1st stage multipliers and adders
- Achieves **100% efficiency** in steady state

<table>
<thead>
<tr>
<th>Size</th>
<th>16</th>
<th>8192</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pins</td>
<td>160</td>
<td>160</td>
<td>143%</td>
</tr>
<tr>
<td>Fly</td>
<td>16</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>Mult</td>
<td>36</td>
<td>144</td>
<td>92%</td>
</tr>
<tr>
<td>Add</td>
<td>108</td>
<td>432</td>
<td>92%</td>
</tr>
<tr>
<td>Shift</td>
<td>4</td>
<td>8K</td>
<td>67%</td>
</tr>
</tbody>
</table>
Outline

• Introduction

• Parallel architecture

• Systolic architecture

• Performance summary
  – Power, operations per second
  – FPGA resources, frequency
  – Latency, throughput

• Conclusions
Results

The current implementation has been placed on a Virtex II 8000 and verified at 150 MHz

- Power: 22 Watts @ 65 C
- GOPS: 86 total @ 3.9 GOPS/Watt

- FPGA resources (XC2V8000)
  - Multipliers: 144 (85%)
  - LUTs and SRLs: 39,453 (42%)
  - BlockRAM: 56 (33%)
  - Flip flops: 35,861 (38%)

- Frequency: 150 MHz
- Latency: 1127 cycles
- Throughput: 1.2 GSPS
Outline

• Introduction

• Parallel architecture

• Systolic architecture

• Performance summary

• Conclusions
  – Applicability to other platforms
  – Future work
Conclusions

• Created a high performance, real-time FFT core
  – Low power (3.9 GOPS/Watt)
  – High throughput (1.2 GSPS), low latency (7.6 µsec/sample)
  – Fixed-point (18-bits), high accuracy (70 dB)

• General architecture
  – Extendable to a generic FPGA core
  – Retargetable to ASIC technology

• Future work
  – Develop a parameterizable IP core generator