Transforming a Linear Algebra Core To An FFT Accelerator

Ardavan Pedram*, John McCalpin*, Andreas Gerstlauer*
★Electrical and Computer Engineering
✪Texas Advanced Computing Center

The University of Texas at Austin
The Era of Heterogeneous Computing

• Physical limits of technology scaling
  • Power/utilization/… walls and dark silicon
    – Only a fragment of a chip may be active at any given time
  • Efficiency/optimality vs. flexibility/generality
    – GFLOPS/W (energy per operation)

➢ Opportunity and need for specialization
  ➢ Heterogeneous multi-core / Asynchronous CMP
  ➢ On-chip accelerators
    ➢ GP-GPUs
    ➢ Programmable, reconfigurable or hardcoded?

Nvidia Tegra 2 System on Chip
Implementation Spectrum

Base Architecture: Linear Algebra Core (LAC)

- **Scalable 2-D array of** $n_r \times n_r$ **processing elements (PEs) [ASAP’11]**
  - Up to 50 GFLOPS/W @ 45nm
  - Specialized floating-point units w/ 1 MAC/cycle throughput
  - Broadcast busses (no need to pipeline up to $n_r=16$)
  - Distributed memory architecture
  - Distributed, PE-local control
  - Level-3 BLAS [ASAP’12], Matrix Factorizations [ARITH21]
GEMM vs. FFT

- **GEMM**
  - High ratio of computation to communication
    - $O(N^3)/O(N^2)$
  - Demonstrates the maximum sustainable FLOPS
  - Balanced: # additions $=$ # multiplications

- **FFT**
  - Modest ratio of computation to communication
    - $O(N \log N)/O(N)$
  - Typically memory BW limited
  - Non-Balanced: # additions $>$ # multiplications
Outline

✓ Introduction
  ✓ Motivation and vision

• Related work
• FFT Algorithm and Mapping
• Architecture Tradeoffs
• Experimental Results
• Conclusions and future work
Related Work

• **CPUs: Poor utilization ~ 40% Peak (Effective)**
  - Powers-of-2 strides in FFT interact badly
    - set-associative caches,
    - set-associative address translation mechanisms
    - power-of-2-banked memory subsystems

• **GPUs: Even poorer utilization ~ 20% peak**
  - More computation units
  - Weaker Memory subsystem

• **FPGAs and ASIC:**
  - Automatic RTL generation [Spiral]

• **A complete comparison in** [Chung2010]
Outline

✓ Introduction
  ✓ Motivation and vision

✓ Related work
  • FFT Algorithm and Mapping
  • Architecture Tradeoffs
  • Experimental Results
  • Conclusions and future work
Outline

✓ Introduction
  ✓ Motivation and vision

✓ Related work
  • FFT Algorithm and Mapping
  • Architecture Tradeoffs
  • Experimental Results
  • Conclusions and future work
FMA Optimized Radix-4

\[
\begin{pmatrix}
  x(j) \\
  x(j + L/4) \\
  x(j + L/2) \\
  x(j + 3L/4)
\end{pmatrix} \times \begin{pmatrix}
  1 & 1 & 1 & 1 \\
  1 & -j & -1 & j \\
  1 & -1 & 1 & -1 \\
  1 & j & -1 & -j
\end{pmatrix} \text{diag}(1, \omega_L^j, \omega_L^{2j}, \omega_L^{3j})
\]

- **Radix-4 butterfly operation**
  - Three complex multiplication
  - Eight complex addition
  - 34 real floating-point operations

\[
\begin{pmatrix}
  x(j) \\
  x(j + L/4) \\
  x(j + L/2) \\
  x(j + 3L/4)
\end{pmatrix} \times \begin{pmatrix}
  1 & 0 & \omega & 1 \\
  0 & 1 & -\omega & 0 \\
  1 & 0 & -\omega & 0 \\
  0 & 1 & 0 & -i\omega
\end{pmatrix} \begin{pmatrix}
  1 & \omega^2 & 0 & 0 \\
  1 & -\omega^2 & 0 & 0 \\
  0 & 0 & 1 & \omega^2 \\
  0 & 0 & 1 & -\omega^2
\end{pmatrix}
\]

- **FMA optimized Radix-4 butterfly operation**
  - 24 FMA operations
  - Reduce the loads for twiddle factors
### Radix-4 Butterfly

#### Non-optimized

For $j = 0 : L/4 - 1$

- **FLOPs**
  - $a := x(j)$;  
  - $b := \omega^j_L x(j + L/4)$;  
  - $c := \omega^{2j}_L x(j + L/2)$;  
  - $d := \omega^{3j}_L x(j + 3L/4)$;  
  - $\tau_0 := a + c$;  
  - $\tau_1 := a - c$;  
  - $\tau_2 := b + d$;  
  - $\tau_3 := b - d$;  
  - $x(j) := \tau_0 + \tau_2$;  
  - $x(j + L/4) := \tau_1 - i\tau_3$;  
  - $x(j + L/2) := \tau_0 - \tau_2$;  
  - $x(j + 3L/4) := \tau_1 + i\tau_3$;

#### FMA Optimized

For $j = 0 : L/4 - 1$

- **FLOPs**
  - $a := x(j)$;  
  - $b := x(j + L/4)$;  
  - $c := x(j + L/2)$;  
  - $d := x(j + 3L/4)$;

- **FMA OPs**
  - $b := a - \omega^{2j}_L b$;  
  - $a := 2a - b$;  
  - $d := c - \omega^{2j}_L d$;  
  - $c := 2c - d$;  
  - $x(j + L/2) := c = a - \omega^{j}_L c$;  
  - $x(j) := 2a - c$;  
  - $x(j + L/4) := d := b - i\omega^{j}_d$;  
  - $x(j + 3L/4) := 2b - d$;

34 end for

24 end for
Radix-4 Butterfly on a PE

- **Dependencies**
  - Multiplication
  - Addition

- **FP-MAC unit**
  - 5-9 stages of pipelining
    - Possible Hazards
  - Single cycle accumulator
    - Only track Multiplication

- **24 Cycles**

- **No Dependence Hazard**
  - Careful scheduling of operations
Fast Fourier Transform

- **LAC Architecture**
  - Broadcast buses
  - Floating-Point MAC in PEs

- $n_r$ be a power of 2
  - Limit communication pattern

- **Butterfly operations on PEs**
  - Optimize operations for FP-MAC units
64 Point Radix-4 FFT on a 4x4 LAC

- **Stage one**
  - No communication
- **Stage two (PE(0,0))**
  - Neighbors of distances $1+4^0$, $2+4^0$, $3+4^0$
  - $n_r(n_r-1)x2=24$ row bus transactions
64 Point Radix-4 FFT on a 4x4 LAC

• **Stage one**
  - No communication

• **Stage two (PE(0,0))**
  - Neighbors of distances $1 \times 4^0, 2 \times 4^0, 3 \times 4^0$
  - $n_r(n_r-1) \times 2 = 24$ row bus transactions

• **Stage three (PE(0,0))**
  - Neighbors of distances $1 \times 4^1, 2 \times 4^1, 3 \times 4^1$
  - $n_r(n_r-1) \times 2 = 24$ column bus transactions
FFT on LAC

- Stage 1 inside PEs
- Stage 2 only row buses
- Stage 3 only column buses
- Stage ≥ 4 only intra PE accesses

- Cycle count:
  - $6N/n_r^2 \log_4 N$

- Effective Bandwidth:
  - $n_r^2/(\log N - 1)$
Larger Out of Core 1D and 2D FFTs

- **1D:** Four step algorithm
  1. $N_1$ DFTs of size $N_2$
     - Write back in bit reverse
  2. Multiply the result by $N_1N_2$ twiddle factors
  3. $N_2$ DFTs of size $N_1$
  4. Transpose the result

- **2D:** Can do any order
  - $N_1/N_2$ DFTs of size $N_2/N_1$
    - Write back in bit reverse
  - Same set of twiddle factors for all rows/columns
  - $N_1/N_2$ DFTs of size $N_2/N_1$
Outline

✓ Introduction
  ✓ Motivation and vision

✓ Related work
✓ FFT Algorithm and Mapping
  • Architecture Configurations
  • Experimental Results
  • Conclusions and future work
Architecture Configurations

- **Communication vs. computation**
  - No overlap
    - 12 KB/ PE
    - 50% Utilization
  - Full overlap
    - 16 KB/ PE
    - 83% Utilization

- **Required BW to core**
  - LAC is limited to 4 Doubles/Cycles
  - FFT core needs more BW for small problems
Hybrid LA/FFT Design

• Core
  • Double the off-core BW
  • Expand memory interface to support rows and columns
    – Symmetric
    – Natively support transpose

• PE
  • 8-Words RF (temporary values)
  • 4-Words RF (twiddle factors)
  • Divide MEMA into two halves
  • Symmetric data-path to both SRAMs
Outline

✓ Introduction
  ✓ Motivation and vision

✓ Related work
✓ FFT Algorithm and Mapping
✓ Architecture Configurations
  • Experimental Results
  • Conclusions and future work
Power and Area Analysis

• **Power and Area**
  • CACTI
  • [Galal’10]
  • @ 1GHz (sweet spot)

• **Power**
  • dominated by FPMAC

• **Efficiency**
  • Higher is better
  • Up to 10% loss
Comparison to Other Designs

- Double-precision 1D FFT performance
- Scaled to 45nm technology

**Proposed FFT:**
- Two orders of magnitude better power efficiency than CPUs and GPUs
- High (83%) effective utilization
- An order of magnitude better area efficiency

<table>
<thead>
<tr>
<th></th>
<th>Problem fits in</th>
<th>GFLOPS</th>
<th>W</th>
<th>GFLOPS/mm²</th>
<th>GFLOPS/W</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xeon E3-1270 core</td>
<td>L2 Cache</td>
<td>12.0</td>
<td>28</td>
<td>0.33</td>
<td>0.43</td>
<td>44%</td>
</tr>
<tr>
<td>ARM Cortex A9</td>
<td>L1 Cache</td>
<td>0.6</td>
<td>0.28</td>
<td>0.45</td>
<td>2.13</td>
<td>60%</td>
</tr>
<tr>
<td>PowerXCell 8i SPE</td>
<td>SPE Local SRAM</td>
<td>12.0</td>
<td>64</td>
<td>0.12</td>
<td>0.19</td>
<td>12%</td>
</tr>
<tr>
<td>Nvidia Tesla C2050</td>
<td>L1+L2 Cache</td>
<td>110.0</td>
<td>150</td>
<td>0.21</td>
<td>0.73</td>
<td>21.3%</td>
</tr>
<tr>
<td>Hybrid Core</td>
<td>On-Core SRAM</td>
<td>26.7</td>
<td>0.66</td>
<td>12.2</td>
<td>40.50</td>
<td>83+%</td>
</tr>
<tr>
<td>Hybrid Core+ SRAM 2MB</td>
<td>Off-Core SRAM</td>
<td>26.7</td>
<td>1.02</td>
<td>1.71</td>
<td>26.30</td>
<td>83+%</td>
</tr>
</tbody>
</table>
Outline

✓ Introduction
  ✓ Motivation and vision

✓ Related work

✓ Base Architecture
✓ FFT Algorithm and Mapping
✓ Architecture Configurations
✓ Experimental Results

• Conclusions and future work
Summary & Conclusions

• **Hybrid FFT/Linear Algebra Core**
  • Algorithm/architecture co-design
  • Power and efficiency estimation

• **Results @ 1GHz for FFT core**
  • DP: 26.7 GFLOPS, 40 GFLOPS/W
  • 0.7 Watts
  • 2.2 mm² in 45nm
  • 83% utilization
  • Orders of magnitude improvement in efficiency

➢ **On-going and future work**
  • Memory hierarchy study of a multi-core FFT design