Architecting an LTE Base Station with Graphics Processing Units

Qi Zheng*, Yajing Chen*, Ronald Dreslinski*, Chaitali Chakrabarti+, Achilleas Anastasopoulos*, Scott Mahlke*, Trevor Mudge*

*University of Michigan, Ann Arbor
+Arizona State University, Tempe

SiPS ’13
Oct 17, 2013
Wireless Base Station
Baseband SDR in a Base Station

GOPS Throughput

CDMA2000

LTE

TD-SCDMA

University of Michigan
Technology Evolution

Peak Data Rate (Mbps)

1.0E+03

1.0E+02

1.0E+01

1.0E+00

1.0E-01

Computation complexity
Computing throughput

CDMA2000

EDGE

HSPA+

LTE
Processor for Wireless Base Station

- Performance -- High computing throughput
- Flexibility -- Good programmability
Processor for Wireless Base Station

- Performance -- High computing throughput
- Flexibility -- Good programmability
Graphics Processing Unit

- High Throughput:
  - GOPS/TOPS-level peak throughput

- Good Programming Support
  - CUDA
  - OpenCL

- High Efficiency

<table>
<thead>
<tr>
<th>Processor</th>
<th>GFLOP/dollar</th>
<th>GFLOP/watt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia GeForce GTX680</td>
<td>6.192</td>
<td>15.848</td>
</tr>
<tr>
<td>Intel Xeon E7-8837</td>
<td>0.037</td>
<td>0.656</td>
</tr>
<tr>
<td>Intel Itanium 8350</td>
<td>0.007</td>
<td>0.150</td>
</tr>
</tbody>
</table>
**GPU Architecture**

- **SIMT – Single Instruction Multiple Threads**
  - Thousands of cores on a GPU
  - Explore Data-Level Parallelism

![GPU Architecture Diagram](image-url)
GPU Architecture

- SIMT – Single Instruction Multiple Threads
  - Thousands of cores on a GPU
  - Explore Data-Level Parallelism

- Multithreading
  - Hide long memory latency

Thread Group 0:

```
Id R0 [R1+Offset]
sub R2, R0, #2
add R0, R2, R3
```
GPU Architecture

- SIMT – Single Instruction Multiple Threads
  - Thousands of cores on a GPU
  - Explore Data-Level Parallelism

- Multithreading
  - Hide long memory latency

Thread Group 1:

\[
\begin{align*}
&\text{mul R3,R4,R5,R6} \\
&\text{ld R0 [R1+Offset]} \\
&\text{sub R2, R0, #2} \\
&\text{add R0, R2, R3}
\end{align*}
\]
GPU Architecture

- SIMT – Single Instruction Multiple Threads
  - Thousands of cores on a GPU
  - Explore Data-Level Parallelism
- Multithreading
  - Hide long memory latency

SIMT – Single Instruction Multiple Threads

Multithreading

GOPS/TOPS-level peak throughput
GPU Mapping Challenge

- Core underutilization
  - Over 1000 cores on a commercial GPU
- Pipeline stall
  - Long memory access latency

![Diagram](image-url)
Our Contribution

- Previous works
  - Mapped only a single kernel onto a GPU
  - Base station study on traditional platforms, such as DSP, FPGA, etc

- In this work
  - Key Kernel Parallelization
  - Kernel runtime performance
  - Minimum number of GPUs needed
  - System Power consumption
Outline

- Motivation
- GPU Architecture
- Key Kernels Parallelization
- Experimental Results
- Conclusion
List of Key Kernels

- PHY Layer
  - SC-FDMA demodulation (FFT)
  - Transform decoder (IDFT)
  - Channel estimation
  - MIMO detection
  - Modulation demapper

- Turbo decoder
## Parallelism in PHY Layer Kernels

<table>
<thead>
<tr>
<th>Parallelism</th>
<th>Description</th>
<th>Num of threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-level</td>
<td>Process data from different users in parallel</td>
<td>#thread = $N_{usr}$</td>
</tr>
<tr>
<td>Antenna-level</td>
<td>Process data from different receiver antennae in parallel</td>
<td>#thread = $N_{ant}$</td>
</tr>
<tr>
<td>Symbol-level</td>
<td>Processing SC-FDMA symbols in a subframe in parallel</td>
<td>#thread = $N_{sym}$</td>
</tr>
<tr>
<td>Subcarrier-level</td>
<td>Each subcarrier in a symbol of is processed in parallel</td>
<td>#thread = $N_{sub}$</td>
</tr>
<tr>
<td>Algorithm-level</td>
<td>parallelism inherent in each algorithm</td>
<td>Varies based on kernels</td>
</tr>
</tbody>
</table>
## Parallelism in PHY Layer Kernels

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Parallelism</th>
<th>Number of threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT/IFFT</td>
<td>User-level</td>
<td>(N_{usr} \times N_{ant} \times N_{sym} \times N_{FFT})</td>
</tr>
<tr>
<td></td>
<td>Antenna-level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Symbol-level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Algorithm-level</td>
<td></td>
</tr>
<tr>
<td>Channel Estimation</td>
<td>User-level</td>
<td>(N_{usr} \times N_{ant} \times N_{sub})</td>
</tr>
<tr>
<td></td>
<td>Antenna-level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subcarrier-level</td>
<td></td>
</tr>
<tr>
<td>MIMO detector</td>
<td>User-level</td>
<td>(N_{usr} \times N_{sym} \times N_{sub})</td>
</tr>
<tr>
<td></td>
<td>Symbol-level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subcarrier-level</td>
<td></td>
</tr>
<tr>
<td>Modulation demapper</td>
<td>User-level</td>
<td>(N_{usr} \times N_{ant} \times N_{sym} \times N_{sub} \times N_{Mod})</td>
</tr>
<tr>
<td></td>
<td>Antenna-level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Symbol-level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subcarrier-level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Algorithm-level</td>
<td>(N_{usr} \times N_{ant} \times N_{sym} \times N_{sub} \times \log_2(N_{Mod}))</td>
</tr>
</tbody>
</table>

University of Michigan
Parallelism in Turbo Decoder*

- Total number of threads

\[ N_{\text{thread}} = N_{\text{packet}} \cdot N_{\text{subblock}} \cdot \text{Thread}_{\text{trellis}} \]

- Implementation performance tradeoff

<table>
<thead>
<tr>
<th>Parallelism Scheme</th>
<th>Throughput</th>
<th>Latency</th>
<th>Bit Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet-level</td>
<td>Better</td>
<td>Worse</td>
<td>No Change</td>
</tr>
<tr>
<td>Subblock-level</td>
<td>Better</td>
<td>No Change</td>
<td>Worse</td>
</tr>
<tr>
<td>Trellis-level</td>
<td>Better</td>
<td>No Change</td>
<td>No Change</td>
</tr>
<tr>
<td>Subblock+NII</td>
<td>Worse</td>
<td>No Change</td>
<td>Better</td>
</tr>
<tr>
<td>Subblock+TS</td>
<td>Worse</td>
<td>No Change</td>
<td>Better</td>
</tr>
</tbody>
</table>

Experimental Setup

- Nvidia GeForce GTX680
  - 8 Streaming Multiprocessors
  - 1536 Streaming Processors
  - 64KB L1 cache + shared memory
  - 512KB L2 cache
  - 2GB DRAM

- GPU Runtime measure
  - CUDA event record

- GPU Power measure
  - GPU-Z
## Experimental Setup

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turbo decoder</td>
<td>Code Rate = 1/3,</td>
</tr>
<tr>
<td></td>
<td>Codeword Length = 6144,</td>
</tr>
<tr>
<td></td>
<td>Iteration Number = 5</td>
</tr>
<tr>
<td>Modulation demapper</td>
<td>16QAM/64QAM</td>
</tr>
<tr>
<td>SC-FDMA FFT</td>
<td>2048</td>
</tr>
<tr>
<td>Decoding IFFT</td>
<td>1200</td>
</tr>
<tr>
<td>MIMO</td>
<td>1x1, 2x2, 4x4</td>
</tr>
</tbody>
</table>
PHY Layer Kernel Runtime

- One LTE subframe on a GPU
- Different antenna configurations
<table>
<thead>
<tr>
<th>Schemes</th>
<th>Subblock Num</th>
<th>Codeword Num</th>
<th>Throughput (Mbps)</th>
<th>Worst-case Codeword Latency (ms)</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tellis-level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State-level</td>
<td>512</td>
<td>2</td>
<td>77.6</td>
<td>0.7</td>
<td>1.6×10⁻³</td>
</tr>
<tr>
<td>State-level</td>
<td>256</td>
<td>4</td>
<td>78.2</td>
<td>1.7</td>
<td>4.1×10⁻³</td>
</tr>
<tr>
<td>State-level</td>
<td>256</td>
<td>2</td>
<td>78.3</td>
<td>0.7</td>
<td>4.1×10⁻³</td>
</tr>
<tr>
<td>State-level</td>
<td>128</td>
<td>7</td>
<td>80.6</td>
<td>3.1</td>
<td>2.0×10⁻³</td>
</tr>
</tbody>
</table>
## Turbo Decoder Performance

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Throughput (Mbps)</th>
<th>Worst-case Codeword Latency (ms)</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tellis-level</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subblock Num</td>
<td></td>
<td>Codeword Num</td>
<td></td>
</tr>
<tr>
<td>State-level</td>
<td>512</td>
<td>2</td>
<td>77.6</td>
</tr>
<tr>
<td>State-level</td>
<td>256</td>
<td>4</td>
<td>78.2</td>
</tr>
<tr>
<td>State-level</td>
<td>256</td>
<td>2</td>
<td>78.3</td>
</tr>
<tr>
<td>State-level</td>
<td>128</td>
<td>7</td>
<td>80.6</td>
</tr>
</tbody>
</table>
Number of needed GPUs

- Meet latency and throughput requirements
  - $\sum (t_k) \leq 1\text{ms}$ for a subframe
  - $Th_{\text{turbo}} \geq Th_{\text{sys}}$
Number of needed GPUs

- Meet latency and throughput requirements
  - \( \sum (t_k) \leq 1\text{ms} \) for a subframe
  - \( Th_{turbo} \geq Th_{sys} \)
Number of needed GPUs

- Meet latency and throughput requirements
  - $\sum (t_k) \leq 1ms$ for a subframe
  - $Th_{turbo} \geq Th_{sys}$
Support 75Mbps

- Two GTX680 GPUs + One Intel Core 2 CPU
- Total power is 188W
Conclusion

- Highly parallel GPU implementations of all key kernels

- Kernel runtimes under different configurations

- Up to four GTX680 GPUs needed for $\leq 150$Mbps
  - Can fit into a motherboard with low latency

- Dual-GPU solution consumes 188W for 75Mbps
  - Competitive with a commercial solution
Thanks!

Any questions?
Backup
Our Contribution

- **Previous works**
  - Mapped only a single LTE kernel onto a GPU
  - Base station study on traditional platforms, such as DSP, FPGA, etc

- **In this work**
  - Parallel implementations of LTE key signal processing kernels on GPU
  - Kernel runtime performance under different system configurations
  - The number of GPUs needed for the baseband subsystem in an LTE base station
  - Power consumption of the GPU-based solution
Parallelism in PHY Layer Kernels

- User-level Parallelism
  - \#thread = N_{usr}

- Antenna-level Parallelism
  - \#thread = N_{ant}

- Symbol-level Parallelism
  - \#thread = N_{sym}

- Subcarrier-level Parallelism
  - \#thread = N_{sub}

- Algorithm-level Parallelism
Parallelism in PHY Layer Kernels

- User-level Parallelism
  - \#thread = N_{usr}

- Antenna-level Parallelism
  - \#thread = N_{ant}

- Symbol-level Parallelism
  - \#thread = N_{sym}

- Subcarrier-level Parallelism
  - \#thread = N_{sub}

- Algorithm-level Parallelism
Parallelism in PHY Layer Kernels

- **User-level Parallelism**
  - \#thread = \( N_{usr} \)

- **Antenna-level Parallelism**
  - \#thread = \( N_{ant} \)

- **Symbol-level Parallelism**
  - \#thread = \( N_{sym} \)

- **Subcarrier-level Parallelism**
  - \#thread = \( N_{sub} \)

- **Algorithm-level Parallelism**
Parallelism in PHY Layer Kernels

- User-level Parallelism
  - $\#thread = N_{usr}$

- Antenna-level Parallelism
  - $\#thread = N_{ant}$

- Symbol-level Parallelism
  - $\#thread = N_{sym}$

- Subcarrier-level Parallelism
  - $\#thread = N_{sub}$

- Algorithm-level Parallelism
Parallelism in PHY Layer Kernels

- User-level Parallelism
  - \#thread = N_{usr}

- Antenna-level Parallelism
  - \#thread = N_{ant}

- Symbol-level Parallelism
  - \#thread = N_{sym}

- Subcarrier-level Parallelism
  - \#thread = N_{sub}

- Algorithm-level Parallelism
Parallelism in PHY Layer Kernels

- **User-level Parallelism**
  - \#thread = \(N_{usr}\)

- **Antenna-level Parallelism**
  - \#thread = \(N_{ant}\)

- **Symbol-level Parallelism**
  - \#thread = \(N_{sym}\)

- **Subcarrier-level Parallelism**
  - \#thread = \(N_{sub}\)

- **Algorithm-level Parallelism**
## Number of needed GPUs

- The minimum number of GTX680 GPUs needed for the baseband system of an LTE base station

<table>
<thead>
<tr>
<th>Peak Data Rate (Mbps)</th>
<th>Number of GPUs</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PHY</td>
<td>Turbo</td>
<td>Total</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>5</td>
<td>4</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>
Processor for Wireless Base Station

- Good programmability
- High computing throughput

CDMA2000

LTE

High computing throughput

Good programmability
Processor for Wireless Base Station

- High computing throughput
- Good programmability
Support 75Mbps
- two GTX680 GPUs + one Intel Core 2 CPU
- Total power is 188W

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Power (W)</th>
<th>Energy (J/subframe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turbo decoder</td>
<td>63.3</td>
<td>144.0</td>
</tr>
<tr>
<td>SC-FDMA FFT</td>
<td>56.7</td>
<td>3.4</td>
</tr>
<tr>
<td>Decoding IFFT</td>
<td>56.9</td>
<td>5.7</td>
</tr>
<tr>
<td>Modulation demapper</td>
<td>56.3</td>
<td>26.5</td>
</tr>
<tr>
<td>Channel estimation</td>
<td>61.8</td>
<td>1.2</td>
</tr>
<tr>
<td>MIMO detector</td>
<td>57.7</td>
<td>1.3</td>
</tr>
</tbody>
</table>