Efficient VLSI architectures for baseband signal processing in wireless base-station receivers

Sridhar Rajagopal
Srikrishna Bhashyam, Joseph R. Cavallaro, and Behnaam Aazhang

This work is supported by Nokia, TI, TATP and NSF
Motivation

Computationally complex algorithms for base-stations

– multiple users, high data rates
– matrix inversions, floating point accuracy needed
– DSP solutions infeasible for real-time [S.Das’99]

Real-time implementations for baseband receiver?

– multiuser channel estimation

*S.Das et al., “Arithmetic Acceleration Techniques for Wireless Base-station Receivers”, Asilomar 1999
Contributions

New estimation scheme

- designed from an implementation perspective
- bit-streaming, fixed-point architecture
- reduced complexity, same error rate performance

Real-time architecture design

- exploit bit-level parallelism
- area-constrained, time-constrained
- real-time with minimum area
Baseband signal processing
Channel estimation

Estimates unknown fading amplitudes and asynchronous delays.
Need for multiuser channel estimation

Detector performance depends on estimation accuracy

Best estimator: Maximum Likelihood

=> jointly estimate parameters for all users

=> Multiuser channel estimation

Single-user sliding correlator used for implementation
Multiuser channel estimation algorithm

\[ R_{bb} \ast A = R_{br} \]

\[ R_{bb} = \sum_L b_i b_i^T \]

\[ R_{br} = \sum_L b_i r_i^H \]

- \( b_i \in \{ -1, 1 \}^{2K} \)
- \( r_i \in \mathbb{C}^N \)
- \( R_{bb} \in \mathbb{R}^{2K \times 2K} \)
- \( R_{br} \in \mathbb{C}^{2K \times N} \)
- \( A \in \mathbb{C}^{2K \times N} \)

- \( b_i \) - Training/Tracking bits
- \( r_i \) - Received signal
- \( N \) - Spreading gain (typically fixed, e.g.: 32)
- \( K \) - Number of users (variable, \( \leq N \))
- \( A \) - Maximum Likelihood channel estimate
Outline

Background

**Channel Estimation - An implementation perspective**

VLSI architectures

- Area-constrained, Time-constrained, Area-Time efficient

DSP Comparisons and Conclusions
Iterative scheme for channel estimation

\[
A^{(i)} = A^{(i-1)} - \mu (A^{(i-1)} * R_{bb}^{(i)} - R_{br}^{(i)})
\]

\[
R_{bb}^{(i)} = R_{bb}^{(i-1)} + b_L * b_L^T - b_0 * b_0^T
\]

\[
R_{br}^{(i)} = R_{br}^{(i-1)} + b_L * r_L^H - b_0 * r_0^H
\]

Bit-streaming, method of gradient descent

Stable convergence behavior with \( \mu \)

Simple fixed-point architecture
Comparison of Bit Error Rates (BER)

Signal to Noise Ratio (SNR)

- Iterative Channel Est.
- Original Channel Est.

\[ O(K^2N) \]
\[ O(K^3 + K^2N) \]

SINR = 0 dB
Paths = 3
Training = 150 bits
Spreading N = 31
Users K = 15
Outline

Background

Channel Estimation - An implementation perspective

VLSI architectures
  - Area-constrained, Time-constrained, Area-Time efficient

DSP Comparisons and Conclusions
Design specifications

32 Users (K)

32 spreading code length (N)

Target = 128 Kbps

- 4000 cycles available at 500 MHz

Single cycle addition/multiplication
Task decomposition

Channel Estimate to Detector

TIME

Correlation Matrices (Per Bit)

\[ R_{br} \quad \mathcal{O}(2KN, 8) \]

\[ R_{bb} \quad \mathcal{O}(2K^2, 8) \]

Iterate

\[ A \quad \mathcal{O}(4K^2N, 8) \]

Tracking Window

\[ b_L(2K, 1) \]

\[ b_0(2K, 1) \]

\[ r_L(N, 8) \]

\[ r_0(N, 8) \]
Architecture design

\[ R_{bb}^{(i)} = R_{bb}^{(i-1)} + b_L * b_L^T - b_0 * b_0^T \]

XNOR gates, UP/DOWN counters

\[ R_{br}^{(i)} = R_{br}^{(i-1)} + b_L * r_L^H - b_0 * r_0^H \]

8-bit adders

\[ A^{(i)} = A^{(i-1)} - \mu (A^{(i-1)} * R_{bb}^{(i)} - R_{br}^{(i)}) \]

8-bit multipliers [Schulte’93]

* Schulte, Swartzlander “Truncated Multiplication with Correction Constant”, Workshop on VLSI Signal Processing,1993
Area-constrained: Min. area, not real-time

\[ R_{bb}^{(i)} = R_{bb}^{(i-1)} + b_L * b_L^T - b_0 * b_0^T \]

\[ R_{br}^{(i)} = R_{br}^{(i-1)} + b_L * r_L^H - b_0 * r_0^H \]

\[ A(i) = A^{(i-1)} - \mu(A^{(i-1)} * R_{bb}^{(i)} - R_{br}^{(i)}) \]
## Area-constrained: Hardware used

<table>
<thead>
<tr>
<th>Blocks</th>
<th>Quantity</th>
<th>Full Adder Cells</th>
<th>Complex</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter</td>
<td>1*8</td>
<td>8</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>Multiplier</td>
<td>1*8</td>
<td>64</td>
<td>*2</td>
<td>128</td>
</tr>
<tr>
<td>Adders</td>
<td>3<em>8 + 2</em>16</td>
<td>56</td>
<td>*2</td>
<td>112</td>
</tr>
<tr>
<td><strong>Total Area</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>248 FA cells</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Time (N=K=32)</th>
<th>4K²N</th>
<th>128,000 cycles</th>
</tr>
</thead>
</table>

Time-constrained: Real time, large area

\[ R^{(i)}_{bb} = R^{(i-1)}_{bb} + b_L * b_L^T - b_0 * b_0^T \]

\[ A^{(i)} = A^{(i-1)} - \mu (A^{(i-1)} * R^{(i)}_{bb} - R^{(i)}_{br}) \]

\[ R^{(i)}_{br} = R^{(i-1)}_{br} + b_L * r_L^H - b_0 * r_0^H \]
# Time-constrained : Hardware used

<table>
<thead>
<tr>
<th>Blocks</th>
<th>Quantity</th>
<th>Full Adder Cells</th>
<th>Complex</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter</td>
<td>$2K^2*8$</td>
<td>$16K^2$</td>
<td>-</td>
<td>$16K^2$</td>
</tr>
<tr>
<td>Multiplier</td>
<td>$4K^2N*8$</td>
<td>$256K^2N$</td>
<td>*2</td>
<td>$512K^2N$</td>
</tr>
<tr>
<td>Adders</td>
<td>$2KN<em>16 + 2KN</em>8 + 4K^2N*16$</td>
<td>$48KN + 64K^2N$</td>
<td>*2</td>
<td>$96KN + 128K^2N$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Total Area (N=K=32)</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Time</th>
<th>Log$_2$(2K)</th>
<th>6 cycles</th>
</tr>
</thead>
</table>
Area-Time efficient architecture design

Area - constrained

– single 8-bit multiplier
– \(4K^2N\) cycles \((128,000)\) [3.81 Kbps, 248 FA Cells]

Time-constrained

– \(4K^2N\) 8-bit multipliers
– \(\log_2(2K)\) cycles \((6)\) [83.33 Mbps, 20,000,000 FA Cells]

Goal: real-time with minimum area

Different parallelism levels for multipliers
Area-Time efficient: Real-time, min. area

\[
R_{bb}^{(i)} = R_{bb}^{(i-1)} + b_L \ast b_L^T - b_0 \ast b_0^T
\]

\[
A^{(i)} = A^{(i-1)} - \mu (A^{(i-1)} \ast R_{bb}^{(i)} - R_{bb}^{(i)})
\]

\[
R_{br}^{(i)} = R_{br}^{(i-1)} + b_L \ast r_L^H - b_0 \ast r_0^H
\]
# Area-Time efficient: Hardware used

<table>
<thead>
<tr>
<th>Blocks</th>
<th>Quantity</th>
<th>Full Adder Cells</th>
<th>Complex</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter</td>
<td>2K*8</td>
<td>16K</td>
<td>-</td>
<td>16K</td>
</tr>
<tr>
<td>Multiplier</td>
<td>2K*8</td>
<td>128K</td>
<td>*2</td>
<td>256K</td>
</tr>
<tr>
<td>Adders</td>
<td>2K<em>16 + 2</em>8 + 1*16</td>
<td>32K + 32</td>
<td>*2</td>
<td>64K + 64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Total Area (N=K=32)</th>
<th>10,000 FA cells</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Time</td>
<td>2KN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,000 cycles</td>
</tr>
</tbody>
</table>
Outline

Background

Channel Estimation - An implementation perspective

VLSI architectures
  – Area-constrained, Time-constrained, Area-Time efficient

DSP Comparisons and Conclusions
**DSP comparisons**

DSPs unable to exploit bit-level parallelism
Inefficient storage of bits
Unable to replace bit-multiplications by add/sub.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Clock Rate</th>
<th>Full Adder Cells</th>
<th>Data Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>C67 DSP</td>
<td>166 MHz</td>
<td>-</td>
<td>1.02 Kbps</td>
</tr>
<tr>
<td>Area</td>
<td>500 MHz</td>
<td>248</td>
<td>3.81 Kbps</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Area-Time</td>
<td>500 MHz</td>
<td>$10^4$</td>
<td>256 Kbps</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Time</td>
<td>500 MHz</td>
<td>$2 \times 10^7$</td>
<td>83.33 Mbps</td>
</tr>
</tbody>
</table>
Scalability of architectures

Design for maximum number of users in the system

Fewer users

- turn off functional units to reduce power
- reconfigure hardware for higher data rates (FPGA)

Investigating K-user design using K/2-user designs.

Investigating DSP extensions
Conclusions

New estimation scheme
- designed from an implementation perspective
- bit-streaming, fixed-point architecture
- reduced complexity, same error rate performance

Real-time architecture designs
- exploit bit-level parallelism
- area-constrained, time-constrained
- real-time with minimum area

=> Real-time architectures for base-band signal processing