DSPs for Future Wireless Base-Stations

Sridhar Rajagopal and Joseph R. Cavallaro

May 8, 2000

This work is supported by Texas Instruments, Nokia, Texas Advanced Technology Program and NSF
Outline

- Background
- DSP Implementation and Task Partitioning
- Reduced Complexity Algorithms
- VLSI Architecture
- Extensions for DSPs
Evolution of Wireless Communications

First Generation
Voice

Second/Current Generation
Voice + Low-rate Data (9.6Kbps)

Third Generation +
Voice + High-rate Data (2 Mbps) + Multimedia
W-CDMA
Communication System Uplink

- Direct Path
- Reflected Paths
- Noise + MAI
- User 1
- User 2
- Base Station
Main Processing Blocks

- Channel Estimation
- Detection
- Decoding

Baseband Layer of Base-Station Receiver
Current DSP Implementation

- Current DSP Implementation
  - C67 at 166MHz; Spreading Code = 31
  - Matched Filter followed by 3-stage Multiuser Detector
  - Target Data Rate = 128 Kbps

Graph showing achieved data rates vs. number of users with different data rates.
Reasons for Poor Performance

- Sophisticated, Compute-Intensive Algorithms
- Need more MIPs/FLOPs performance
- Unable to fully exploit pipelining or parallelism
- Bit - level computations / Storage
- Task Partitioning
  - Multiple DSPs, FPGAs
Iterative Scheme for Estimation

\[
\begin{align*}
R_{bb} &= \sum b_i b_i^T \\
R_{br} &= \sum b_i r_i^H \\
R_{bb} \ast A_i &= R_{br}
\end{align*}
\]

\[
\begin{align*}
R_{bb} &= R_{bb} + b_L \ast b_L^T - b_0 \ast b_0^T \\
R_{br} &= R_{br} + b_L \ast r_L^H - b_0 \ast r_0^H \\
A &= A - \mu(A \ast R_{bb} - R_{br})
\end{align*}
\]

- Tracking
- Method of Gradient Descent
- Stable convergence behavior
  - Symmetric, Positive Definite $R_{bb}$
  - Condition number - MAI, SNR, Preamble length
- $\mu$ - reciprocal of maximum eigenvalue
Simulations - AWGN Channel

Comparison of Bit Error Rates (BER)

Signal to Noise Ratio (SNR)

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

\[ O(K^2N) \]

MF – Matched Filter
ML – Maximum Likelihood
ACT – using inversion
Fading Channel with Tracking

Doppler = 10 Hz, 1000 Bits, 15 users, 3 Paths

SNR vs BER for different methods:
- MF - Static
- MF - Tracking
- ML - Static
- ML - Tracking

The graph shows the BER performance with respect to SNR for the specified conditions.
VLSI Implementation

- Channel Estimation as a Case Study
- Area - Time Efficient Architecture
- Real - Time Implementation
  - Minimum Area Overhead
- Bit- Level Computations - FPGAs
- Core Operations - DSPs
Area-Time Tradeoffs

- **Area-Constrained Architecture**
  - Pico-cells; lower data rates

- **Time-Constrained Architecture**
  - Maximum achievable data rates

- **Area-Time Efficient Architecture**
  - Real-Time with minimum area overhead

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Adder Cells</th>
<th>Memory</th>
<th>Time</th>
<th>Data Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>248</td>
<td>16 K</td>
<td>0.262 ms</td>
<td>3.81 Kbps</td>
</tr>
<tr>
<td>Time</td>
<td>$2 \times 10^7$</td>
<td>4 K</td>
<td>12 ns</td>
<td>83.33 Mbps</td>
</tr>
<tr>
<td>Area-Time</td>
<td>$10^4$</td>
<td>16 K</td>
<td>4 μs</td>
<td>256 Kbps</td>
</tr>
<tr>
<td>C67 DSP</td>
<td>-</td>
<td>128 K</td>
<td>0.97 ms</td>
<td>1.02 Kbps</td>
</tr>
</tbody>
</table>
Characteristics of Wireless Algorithms

- Massive Parallelism
- Bit-level Computations
- Matrix Based Operations
  - Memory Intensive
- Complex-valued Data
- Approximate Computations
Instruction Set Extensions

- To accelerate Bit level computations in Wireless
  - Integer - Bit Multiplications
    - Multiuser Detection, Decoding, Cross Correlation
  - Bit - Bit Multiplications
    - Auto-Correlation, Channel Estimation
- Useful in other Signal Processing applications
  - Speech, Video, etc.
SIMD Parallelism

64-bit Register A

64-bit Register B

64-bit Register C
Integer - Bit Multiplications

For $i = 1..8$, $j = 1..8$

$$D[i][j] = D[i][j] + b[i] \times C[j] \quad \text{(Cross-Correlation)}$$
Computational Savings

- Avoid bit multiplications and control structures
- 4 8-bit Multiply
  - Latency 3
- 8 8-bit Add
  - Latency 1
- Cross-Correlation Example
  - 64 multiply, 64 add

<table>
<thead>
<tr>
<th>Original SIMD Instruction Set</th>
<th>12<em>3 + 8</em>1 = 54 cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>With Extensions</td>
<td>8*1 = 8 cycles</td>
</tr>
</tbody>
</table>
Conclusions

- Architecture and Algorithms to meet real-time
- Task Decomposition
  - Real Time with Multiple Processing Elements
- Iterative Algorithms
  - Reduce Complexity, Simpler Implementation
- VLSI Implementation
  - Real-Time with minimum Area Overhead
- Extensions to DSPs for acceleration