Implementation of MIMO Receiver Algorithms

Cooperative MIMO Techniques for Cellular System Evolution
CoMIT
WP #2: Algorithms and Architectures

Janne Janhunen, Johanna Ketonen, Tuomo Hänninen, Shahriar Shahabuddin
firstname.lastname@ee.oulu.fi
Outline

- Introduction
- System model
- Algorithms
- Platforms
- Design flow
- Results
- Conclusions
Introduction

- Realistic scenarios and data rate targets
- State-of-the-art algorithms
- Implementation aspects of receiver algorithms
- Performance vs. complexity
System Model

- Orthogonal frequency division multiplexing (OFDM) and single carrier FDM
- Multiple antennas at the transmitter and receiver (MIMO)
Receiver algorithms

- A multitude of receiver algorithms has been studied

- Least squares (LS)
- Minimum mean square error (MMSE)
- Decision directed estimators

- Co-channel interference suppression

- MMSE
- Successive Interference Cancellation (SIC)
- K-best List Sphere Detector
- Selective spanning with fast enumeration (SSFE)
- Layered ORthogonal lattice Detector (LORD)

- Turbo decoding
Platforms

- Programmable platforms
  - Digital signal processor (DSP)
    - Texas Instruments C67x, C55x C64x
    - Optimum Semiconductor Technologies Sandblaster 3011 and 3500
  - Transport Triggered Architecture (TTA)
  - Graphics Processing Unit (GPU)

- Reconfigurable platforms
  - FPGA
    - Xilinx Virtex-6 XC6VLX240T

- Application-specific integrated circuits (ASIC)
Design Flow

Matlab code

C code

Simulator

Matlab

TTA Processor Design

DSP Programming

High Level Synthesis (HLS)

Hand written RTL

TTA Codedesign environment

Code Composer Studio / Sandblaster IDE

Xilinx AutoESL

Xilinx ISE/EDK

FPGA Synthesis

Synopsys Design Vision

FPGA Synthesis

Synopsys Design Vision

C / TTA Assembly Programming

Mapping on FPGA

Xilinx ISE/EDK

Xilinx ISE/EDK

ASIC Synthesis

Synopsys Design Vision

ASIC Synthesis

Synopsys Design Vision

Synopsys Design Vision

Xilinx ISE/EDK

Xilinx ISE/EDK
## Results

Transport triggered architecture processor results

<table>
<thead>
<tr>
<th>TTA processor 130 nm CMOS technology</th>
<th>12-bit floating point processor</th>
<th>16-bit fixed point processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>SSFE</td>
<td>SSFE</td>
</tr>
<tr>
<td>Antenna configuration</td>
<td>2x2 / 4x4</td>
<td>2x2 / 4x4</td>
</tr>
<tr>
<td>Modulation</td>
<td>16-QAM / 64-QAM</td>
<td>16-QAM / 64-QAM</td>
</tr>
<tr>
<td>Processor clock frequency (MHz)</td>
<td>200</td>
<td>277</td>
</tr>
<tr>
<td>Core size (gate equivalents)</td>
<td>65,550</td>
<td>70,730</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>36.8 / 43.1</td>
<td>- / 64.0</td>
</tr>
<tr>
<td>Energy (nJ / detected bit)</td>
<td>1.04 / 0.89</td>
<td>- / 0.95</td>
</tr>
<tr>
<td>Single core throughput (Mbps)</td>
<td>35.5 / 48.5</td>
<td>49.2 / 67.0</td>
</tr>
</tbody>
</table>

FPGA results

<table>
<thead>
<tr>
<th>K-best Sphere Detector (K=8)</th>
<th>One high throughput block (Il=16)</th>
<th>Four average throughput blocks in parallel (Il=64)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUT (look-up table)</td>
<td>69,383</td>
<td>34,476</td>
</tr>
<tr>
<td>FF (flip-flop)</td>
<td>97,676</td>
<td>50,044</td>
</tr>
<tr>
<td>DSP48</td>
<td>228</td>
<td>216</td>
</tr>
<tr>
<td>BRAM</td>
<td>287</td>
<td>28</td>
</tr>
<tr>
<td>Max frequency (MHz)</td>
<td>231</td>
<td>247</td>
</tr>
<tr>
<td>Throughput (Mbps)</td>
<td>347</td>
<td>372</td>
</tr>
</tbody>
</table>
Results

- Implementation results for a sphere detector and SIC detector for a 4x4 16-QAM system

<table>
<thead>
<tr>
<th></th>
<th>K-best, list size 8</th>
<th>SIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gates</td>
<td>309 k</td>
<td>367 k</td>
</tr>
<tr>
<td>Power (W)</td>
<td>0.78</td>
<td>0.63</td>
</tr>
<tr>
<td>Max frequency</td>
<td>280 MHz</td>
<td>100 MHz</td>
</tr>
<tr>
<td>Detection rate</td>
<td>140 Mbps</td>
<td>400 Mbps</td>
</tr>
<tr>
<td>Goodput at 24 dB</td>
<td>113 Mbps</td>
<td>21 Mbps</td>
</tr>
</tbody>
</table>

- Results include channel preprocessing and log-likelihood ratio calculation
Conclusions

✓ A wide range of receiver algorithms achieving real-time requirements has been implemented using programmable and reconfigurable platforms and application-specific circuits

✓ For programmable platforms the transport triggered architecture has been proved to be very promising

✓ A floating-point arithmetic applying half precision or shorter word length is feasible for programmable platforms in order to reduce programming burden, improve the code legacy and gain in latency and energy consumption.

✓ Both the error rate performance and implementation results have to be considered when comparing algorithms
Thank you!


93.
