Detector Implementations Based on Software Defined Radio for Next Generation Wireless Systems

Janne Janhunen
janne.janhunen@ee.oulu.fi
Outline

- Introduction
- Benefits and Challenges of Programmability
- System Model
- Unlinear Detector Algorithms
- Programmable Platforms and Architectures
- Results
- Conclusions
Introduction

- MIMO technique combined with OFDM (MIMO-OFDM) has been introduced to 3GPP LTE and WiMAX and proposed to LTE-A.
- High data rate requirements cause challenges to the real-time implementations.

- A software defined radio (SDR) is a radio communication system where components are implemented using software on a computing device.

- Algorithm study and development
  - K-best list sphere detection (LSD) algorithm
  - Layered ORthogonal lattice Detector (LORD)
  - Selective Spanning with Fast Enumeration (SSFE)

- Programmable platforms
  - Digital Signal Processors (DSP) such as TMS320C6711 (floating point), TMS320C55x (fixed-point) and TMS320C6455 (fixed-point).
  - System-on-a-chip such as Sandbridges SB3011 and SB3500 devices which employ multi-threading and multiple cores.
  - Application-specific instruction-set processor (ASIP) which is based on the transport triggered architecture (TTA)
Benefits and Challenges of Programmability

- Programmability = reuse of hardware
- Programmable platform provides an opportunity to exploit the silicon more efficiently than a pure hardware implementation in a multi-standard world.
- In addition, software design and time-to-market is faster than in hardware design.
- To improve performance, programmable core can be accelerated with fine grained accelerators.
- However, programmability increases power consumption (possible up-to 20-50x compared to corresponding hardware accelerator) and computational overhead.
  - Instruction fetch/decode
  - Caches
  - Registers
  - Control

<table>
<thead>
<tr>
<th>Platform</th>
<th>Power consumption/operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware accelerator</td>
<td>~5-10pJ</td>
</tr>
<tr>
<td>90 nm CMOS</td>
<td></td>
</tr>
<tr>
<td>Embedded processor</td>
<td>~125-500pJ</td>
</tr>
<tr>
<td>General purpose processor</td>
<td>~10-20nJ</td>
</tr>
</tbody>
</table>

Silvén 2008

Embedded processor energy consumption breakdown.
Dally et al 2008
The MIMO-OFDM system model requirements are based on the 3G LTE standard.

The received signal can be described with the equation

\[ y_s = H_s x_s + \eta_s, \quad s = 1, 2, \ldots, S, \]

where \( S \) is the number of subcarriers, \( x \) is the transmitted signal, \( \eta \) is the Gaussian noise vector and \( H \) is the channel matrix.
Unlinear Detector Algorithms and Simulations
Unlinear Detector Algorithms

- All algorithms are based on the tree type of search
- An example: 2x2 antenna system, 16-QAM, real system model

K-best, K=4

+ Fixed computational complexity
+ Fixed throughput
+ Amount of control is small

- Wasted partial Euclidean distance (PED) computation
- Large list size increases the computational complexity fast
- Expensive sorter operation
- Limited possibility to parallelize tree search between levels

Layered ORthogonal lattice Detector (LORD)

+ Fixed computational complexity
+ Fixed throughput
+ Rather simple slicing operation replaces the expensive sorting
+ No unnecessary PED computation
- In typical case, high number of nodes (constellation points) are required on the top level of the tree
- Final list size might be high

Selective Spanning with Fast Enumeration (SSFE), m=[2 1 2 2]
Simulation
Platforms and Architectures
Platforms and Architectures

**Digital Signal Processors (DSP)**  
TMS320C6711 (floating point VLIW (Very Long Instruction Word))  
TMS320C6455 (fixed-point VLIW)  
TMS320C55x (fixed-point, low-power processor)

**System-on-a-chip (SoC)**  
Sandbridge SB3500 (multi-threading and multiple cores, resembles VLIW)

**Application-specific instruction-set processor (ASIP)**  
Transport Triggered Architecture (TTA)

VLIW (Fischer 1983)  
TTA (Corporaal 1991)
Transport Triggered Architecture

- TTA resembles a VLIW architecture
  - TTA instruction word consists of multiple moves -> one for each bus
  - Each move determines the data transport on the corresponding bus
  - Very fine-grained control
  - Allows optimization which is not available in a conventional processors, e.g. data moves between functional units without using registers
- Finite State Machine of a hardware accelerator is replaced by the transport program in TTA
  - About the same number of control bits are required as in FSM based data path control
  - Depending on the design, it is possible to achieve the same energy efficiency with TTA as ASIC.

```
add R0, R1 → R2
R0 → adder.operand
R1 → adder.trigger
adder.result → R2
adder.result → mul.operand
```
Transport Triggered Architecture

- The bypass network of the processor is exposed to the programmer/compiler
  - Software has complete control over the internal transports
  - Operations are side-effects of data transports: only one instruction – MOVE!
  - Writing data into a *triggering port* of a functional unit starts computation
  - The latencies of functional units are visible to programmer/compiler

- TCE (TTA Codesign Environment)
  - C-compiler available in a toolset

- Mapping TTA on platform: FPGA and ASIC
Transport Triggered Architecture
OSEd - Operation Set Editor

With OSEd it is possible to add, simulate and delete operation definitions
Four implementations of $K$-best list sphere detector

- $K=16$

<table>
<thead>
<tr>
<th></th>
<th>Clock frequency (MHz)</th>
<th>Throughput (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMS320C6455</td>
<td>1200</td>
<td>1.8</td>
</tr>
<tr>
<td>Sandblaster 3500</td>
<td>1800 (3x600)</td>
<td>3.4</td>
</tr>
<tr>
<td>Sandblaster 3500 + instruction set extension for sorter</td>
<td>1800 (3x600)</td>
<td>32.0</td>
</tr>
<tr>
<td>ASIP based on TTA</td>
<td>280</td>
<td>7.6</td>
</tr>
</tbody>
</table>
Results(2)

- A design example of $K$-best-LORD algorithm
  - TTA assembly hand coded -> tight scheduling, all the function units are kept busy
  - 2x2 16-QAM system: 35 clock cycles per tree search for LORD, two searches per symbol vector in LORD algorithm. Therefore 70 clock cycles required for symbol vector.

<table>
<thead>
<tr>
<th>FU</th>
<th># of FUs</th>
<th>Latency (cc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUL</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>SLICER</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>ADD/SUB</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>SORTER</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>REG BANKS (8x16bit, 1024 bits)</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clock rate [MHz]</th>
<th>Decoding rate [Mbps]</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>11.4</td>
</tr>
<tr>
<td>300</td>
<td>34.4</td>
</tr>
<tr>
<td>500</td>
<td>57.1</td>
</tr>
</tbody>
</table>
Conclusion

- MIMO technique combined with OFDM (MIMO-OFDM) provides an opportunity for higher data rates but real-time implementation has to be pushed on the edge.

- Digital signal processors require (fine grained) accelerators to achieve expected data rates.

- Because there are multiple (wireless communication) standards to be supported, programmable platforms are of interest.

- Software defined radio is an old concept. However, not until now techniques have become mature enough to start responding to the expectations what has been built on it.
Thank you!