Multicore DSP Architecture and Programming

O. Dahl\textsuperscript{1}

\textsuperscript{1}Electrical Engineering, Linköping University, Linköping, Sweden

Guest lecture in TDDD56 Multicore and GPU Programming, LiU, December 5, 2011
Personal background

- At LiU since 2011-01-01, at ISY (Institutionen för Systemteknik) - Associate Professor in System Integration (a new subject at the department)
  http://www.da.isy.liu.se/~olad/

- Moved from ST-Ericsson

- Started at Ericsson November 2006 - worked with applications, software architecture, LTE design, simulation for software development

- Before that: engineer, consultant, manager, associate professor in Computer Science and Automatic Control

- Experience in software development, system engineering, system development, simulation, real-time systems, control

- Ph D Automatic Control, Lund, 1992
Problem to solve

How to make a wireless modem for 3GPP LTE (and older standards as well e.g. WCDMA, GSM)?
Challenges

Meet requirements in
- high-speed wireless mobile communication (> 100Mb/s)
- standards compliance (3GPP)
- competitiveness
- silicon size (square millimeters)
- power consumption (mW to W)
- flexibility (many standards, backwards compatibility)
The Smartphone Disruption [Gustafsson, 2011]
ST-Ericsson M7400 [ST-Ericsson, 2011]
3gpp [3GPP, 2011]
ePUMA [ePUMA, 2011] - with contributions from Joar Sohl and Andreas Karlsson
Coresonic [Coresonic, 2011]
ST-Ericsson EVP [ST-Ericsson, 2009]
System-C and TLM - http://www.systemc.org (temporarily down due to merger with Accellera - see e.g. [Doulos, 2011] until December 7)
Virtual platforms e.g. [Corleto, 2009]
Wikipedia
Outline

1 System requirements
   • 3GPP
   • LTE - basic concepts

2 System design
   • DSP
   • DSP - ePUMA
   • ASIC
   • Control processors

3 System development

4 Summary
Outline

1. System requirements
   - 3GPP
   - LTE - basic concepts

2. System design
   - DSP
   - DSP - ePUMA
   - ASIC
   - Control processors

3. System development

4. Summary
3GPP LTE

- Specifications from [3GPP, 2011], e.g. 36.201, 36.211, 36.212
- Increased data rates e.g. 100-300 Mbit/s downlink, > 50 MBit/s uplink
- Scalable channel bandwidth
- OFDM, MIMO
- Packet-switched all-IP solution (no circuit switching)
- Sub-5ms latency

Overview e.g. in [Agilent, 2009]
Digital modulation

Map sequence of bits to a complex number

QAM - Quadrature Amplitude Modulation [Wikipedia, 2011a]
I and Q - complex numbers

- Introduce the carrier frequency $\omega_c$
- Send $s(t) = I(t) \cos(\omega_c t) + Q(t) \sin(\omega_c t)$
- Receive, with disturbance $n(t)$, $\hat{s}(t) = s(t) + n(t)$

Define $s_1(t) = \hat{s}(t) \cos(\hat{\omega}_c t)$ and calculate

$$s_1(t) = \hat{s}(t) \cos(\hat{\omega}_c t)$$
$$= I(t) \cos(\omega_c t) \cos(\hat{\omega}_c t) + Q(t) \sin(\omega_c t) \cos(\hat{\omega}_c t) + n(t) \cos(\hat{\omega}_c t)$$
$$= I(t) \frac{1}{2} (\cos((\omega_c - \hat{\omega}_c) t) + \cos((\omega_c + \hat{\omega}_c) t)) +$$
$$Q(t) \frac{1}{2} (\sin((\omega_c + \hat{\omega}_c) t) + \sin((\omega_c - \hat{\omega}_c) t)) + n(t) \cos(\hat{\omega}_c t)$$

- Low-pass filtering and $\hat{\omega}_c \approx \omega_c$ gives $2s_1(t) \approx I(t)$
Similarly, define $s_2(t) = \hat{s}(t) \sin(\hat{\omega}_c t)$ and calculate

$$s_2(t) = \hat{s}(t) \sin(\hat{\omega}_c t)$$
$$= I(t) \cos(\omega_c t) \sin(\hat{\omega}_c t) + Q(t) \sin(\omega_c t) \sin(\hat{\omega}_c t) + n(t) \sin(\hat{\omega}_c t)$$
$$= I(t) \frac{1}{2} (\sin((\hat{\omega}_c + \omega_c) t) + \sin((\hat{\omega}_c - \omega_c) t)) +$$
$$Q(t) \frac{1}{2} (\cos((\hat{\omega}_c - \omega_c) t) - \cos((\hat{\omega}_c + \omega_c) t)) + n(t) \sin(\hat{\omega}_c t)$$

Low-pass filtering and $\hat{\omega}_c \approx \omega_c$ gives $2s_2(t) \approx Q(t)$
LTE - basic concepts

OFDM

- Send data on multiple frequencies
- Send during a symbol interval $T_u$
- Use subcarrier spacing $\Delta f = \frac{1}{T_u}$
- In LTE, $\Delta f = 15kHz$ (mostly), i.e. $T_u \approx 66.7\mu s$
OFDM - orthogonality

Fourier transform of a pulse [Wikipedia, 2011b]
Orthogonality, since signals on two subcarriers

\[ x_1(t) = a_1 e^{j2\pi k_1 \Delta ft}, \quad x_2(t) = a_2 e^{j2\pi k_2 \Delta ft} \]

fulfil

\[ \int_{mT_u}^{(m+1)T_u} x_1(t)x_2^*(t)dt = \int_{mT_u}^{(m+1)T_u} a_1 a_2^* e^{j2\pi (k_1 - k_2) \Delta ft} dt = 0 \]

for \( k_1 \neq k_2 \)
OFDM - implementation using FFT

*OFDM can be implemented using FFT (Fast Fourier Transform) at receiver side and IFFT (Inverse FFT) at sender side*
OFDM and modulation - sender

[Wikipedia, 2011c]
OFDM and modulation - receiver

[Wikipedia, 2011c]
Coding and Decoding

- Main coding algorithm is Turbo coding with a coding rate $R = 1/3$
- Convolutional coding (for BCH - broadcast channel)

Turbo encoder [Wikipedia, 2011d]
Parallel signal processing

OFDM symbols received in series from the radio interface, processed in parallel, processing stages include e.g. FFT, demodulation, control decoding, data decoding.

Uplink processing proceeds in parallel
Channel estimation

- Estimate properties of channel
- Compensate for channel effects
- Communication with base station
- Reference signal (pilot symbols)
LTE - basic concepts

MIMO

- Multiple-antennas
- Diversity techniques
- Spatial multiplexing (send more than one data stream)
And there is more ...

- synchronization (time, frequency)
- cell search
- receive system information
- power control
- uplink synchronization (timing advance)
- FDD and TDD
- Random access
- Paging
- HARQ

and ... this is only L1 ... we have to make a complete protocol stack ... and it has to be mobile (handover etc.)
LTE - basic concepts

What speed do we get?

- 20Mhz bandwidth, 1200 subcarriers
- 14 OFDM symbols in one subframe (1 ms)
- 64QAM - 6 bits per resource element
- \(14 \times 6 \times 1200 / 1e-3 = 100800000\) (without coding, control information, but also without MIMO)
Outline

1. System requirements
   - 3GPP
   - LTE - basic concepts

2. System design
   - DSP
   - DSP - ePUMA
   - ASIC
   - Control processors

3. System development

4. Summary
Building blocks

- DSP
- ASIC
- Control processors
And there is more ...

- Application processors
- Radio, radio interface
- Interconnect, buses
- Memory, caches
- Power management, thermal management
- Imaging, video, graphics, display
- Storage, e.g. flash, memory card
Leocore

Information from [Coresonic, 2011, Anjum et al., 2011]

- Leocore
- ASIP for baseband processing
- Identify common operations in baseband processing - domain specific architecture
- Coresonic developer studio
- SIMT™ - Single Instruction-flow Multiple Tasks
- Units for complex calculations, control unit (RISC), accelerators for FEC (Viterbi, Turbo)
- DFE interface, MAC interface
DSP

Master thesis proposal - Parallel Simulation of Multicore DSP Systems for Software Defined Radio

- Develop parallel version of simulation tool
- Utilize multicore on the host
- Threads - partitioning, synchronization, interaction
- Static analysis, dynamic analysis
- Requires competence in concurrent programming and hardware/software interaction. Knowledge of DSP hardware and software is beneficial, but not strictly required
- C++, some Python

more info at [Computer Engineering, 2011]
Information from [ST-Ericsson, 2009]

- EVP
- Vector processor (SIMD)
- VLIW instructions - 6 parallel vector operations, 4 parallel scalar operations
- C control structures
- Code generator for scrambling and generating channelization codes
- $< 0.5 mW/MHz$
ePUMA

Research project at Division of Computer Engineering, cooperation also with Information Coding (ISY) and IDA (parallel programming)

- Overview
- Master thesis proposals
Highly parallel processor for predictable DSP tasks

Heterogenous design:
- 1 master control processor
- 8 slave processor cores

Exploited parallelism:
- Task-parallelism (several processor cores)
- Data-parallelism (SIMD instructions on slave processors)
Applications

Some example applications:

- Baseband processing.
- Media processing.
- Radar.

Often in constrained environments, such as phones. Ordinary processors often fail because of

- high power consumption.
- high cost.
- low performance.
Properties of DSP algorithms

Most DSP algorithms share some common traits.

- Predictable addressing. i.e the addresses of the accessed values are not data dependant.
- Few branches other than back jumps in loops.
- Constant iteration counts.

Application Specific Instruction set Processors (ASIPs) for DSP take advantage of this to solve the previous problems.
System overview
DSP - ePUMA

Memory hierarchy

Off chip main memory

On chip interconnection

Master LS
PM DM 0 DM 1

Sleipnir 0 LS
PM CM LVM 1 LVM 2 LVM 3

Sleipnir 7 LS
PM CM LVM 1 LVM 2 LVM 3

Level 1

Level 2

Level 3

Master Core
Registers

Sleipnir Core
Registers
Sleipnir features

- Scratchpad memory based programming - no data cache
- Up to 16-way SIMD datapath (operates on 128 bit data vectors)
- Up to 16 real or 4 complex multiplications per cycle (16 bit data)
- Supported datatypes:
  - Real fixed-point data: 8, 16, 32 bits
  - Complex fixed-point data: 16, 32 bit real and imaginary parts
  - Single precision floating-point (32 bits)
- Special purpose instructions: DCT, butterflies, sort...
Sleipnir customization

Many parameters can be customized:

- Instruction set
- Local memory and register file sizes
- AGU capabilities
- Accelerators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local vector memory (LVM) size</td>
<td>Up to 8k 128-bit vectors (128kB)</td>
</tr>
<tr>
<td>Register file size</td>
<td>8-32 vectors (0.125 - 0.5 kB)</td>
</tr>
<tr>
<td>Constant memory size</td>
<td>Up to 256 vectors (4kB)</td>
</tr>
<tr>
<td>Program memory size</td>
<td>Typically 8-16 kB</td>
</tr>
</tbody>
</table>
Normally many cycles are wasted on rearranging data with shuffle-instructions. This is often due to issues with data alignment and bank-conflicts.
Consider the following address layout in a single bank memory. The only vectors of length four that can be accessed in one cycle is the row vectors \{0, \ldots, 3\}, \{4, \ldots, 7\}, \{8, \ldots, 11\} and \{12, \ldots, 15\}. Accessing one of the colored column vectors take 4 cycles.
By splitting the memory into different banks (which increases the area cost somewhat), the only constraint is that no two elements reside in the same bank. So while we may now access e.g. vectors \{x, \ldots, x+3\} in one cycle, the columns still take four cycles to access.
Multi-bank and permutation

Given that the access patterns are known in advance, as is common in DSP algorithms, we may reorder the physical addresses of the logical addresses. An example of a permutation that allows single cycle access for the columns can be seen below. No two elements of any column reside in the same memory bank.
## ePUMA system benchmarks

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Sleipnir</th>
<th>Cell</th>
<th>GTX280</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 × 8 DCT/Q</td>
<td>5</td>
<td>≈ 59</td>
<td></td>
</tr>
<tr>
<td>8 × 8 DCT</td>
<td>4</td>
<td></td>
<td>≈ 66</td>
</tr>
</tbody>
</table>

**Table:** Average required clock cycles

Execution time for DCT/Quantization for ePUMA @ 300 MHz ≈ Cell @ 3.2 GHz.
Master thesis proposal - Sleipnir accelerator interface design

• Some problems are not well handled by processors

• One solution: Design an accelerator

• Thesis proposal:
  • Design an accelerator interface to Sleipnir
  • Implement and evaluate accelerators for Sleipnir
Master thesis proposal - Memory architecture evaluation

- Evaluate different memory configurations for our multicore architecture
  - Single-bank
  - Multi-bank
  - Multi-bank with permutation
  - Multi-port
  - Cache
  - ...

- Investigate impact of memory architecture for different applications

- Interesting aspects:
  - Performance
  - Power-consumption
  - Chip area
Master thesis proposal - FPGA Board Demo of ePUMA

- Setting up an FPGA board demo of ePUMA to verify the hardware design
- Goals:
  - Setting up demo environment
  - Test some of our existing demo applications (MotionJPEG and MPEG2-decoder) on real hardware
  - Possibility to set up your own demo!

more info at [Computer Engineering, 2011]
 ASIC

- Decide which blocks to be implemented in hardware
- Decide on programmability
- Power consumption
Control processors

Control processor(s)

- Modem control
- Power control
- ARM Cortex R, M (A)
- RTOS
Outline

1. System requirements
   - 3GPP
   - LTE - basic concepts

2. System design
   - DSP
   - DSP - ePUMA
   - ASIC
   - Control processors

3. System development

4. Summary
Parallel development

- Concurrent development of hardware and software
- Hardware simulation for software development
- Virtual platform
SystemC

- Event-driven simulation framework
- Handles time and parallel activities
- Standardized by OSCI, IEEE
- C++ class library
TLM

- Transaction level modeling
- Function calls vs. pin-level simulation
- Bit-accurate interfaces
- Varying degrees of timing can be added (loosely timed, approximately timed)
- Hardware modeling for software verification
Virtual platform

- A virtual representation of the system
- SystemC and TLM
- Processor models
- Peripheral models
- Commercial tools
- Model handling - signal processing models, HW verification models, virtual platform models
- Acceptance and usage, finding bugs, early SW development and verification, release of platform, supporting different RATs, software layer dependencies
Outline

1 System requirements
   - 3GPP
   - LTE - basic concepts

2 System design
   - DSP
   - DSP - ePUMA
   - ASIC
   - Control processors

3 System development

4 Summary
Summarizing notes

- LTE as an example of multicore digital signal processing
- Digital signal processors and ASIC blocks
- Control processors
- 3GPP
- Time-to-market
- Hardware and Software as parallel development tracks
3GPP (2011).  
3GPP - specification numbering.  
http://www.3gpp.org/specification-numbering.

Agilent (2009).  
3GPP Long Term Evolution: System overview, product development, and test challenges.  
http://cp.literature.agilent.com/litweb/pdf/5989

EURASIP Journal on Wireless Communications and Networking.

Master thesis proposals.  
http://www.da.isy.liu.se/undergrad/exjobb/open/e
Coresonic - company website.
http://www.coresonic.se/.

Virtual platforms: Enablement challenges.

Doulos (2011).
SystemC TLM 2.0.

ePUMA (2011).
ePUMA embedded parallel DSP platform with unique memory access.
http://www.da.isy.liu.se/research/current/ePUMA/

The smartphone disruption.
In *ELLIIT Workshop, Lund.*
http://www.control.lth.se/media/ELLIIT/Workshop2

**ST-Ericsson (2011).**
THOR M7400 LTE and HSPA+.

**ST-Ericsson (2009).**
Low-power Embedded Vector DSP.
http://www.stericsson.com/sales_marketing_resource

**Wikipedia, C. d. (2011a).**
Constellation diagram.
http://en.wikipedia.org/wiki/Constellation_diagram

**Wikipedia, F. t. (2011b).**
Fourier transform.
http://en.wikipedia.org/wiki/Fourier_transform

**Wikipedia, O. (2011c).**
OFDM.

http://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing
