MULTICORE DIGITAL SIGNAL PROCESSING

Maxime Pelcat – mpelcat@insa-rennes.fr

Slides from M. Pelcat, K. Desnos,
J.-F. Nezan, D. Ménard,
M. Raulet, J. Gorin, F. Pescador
Multicore DSPs – Karol Desnos (kdesnos@insa-rennes.fr)
Introduction: Porting Signal Processing Algorithms to MPSoCs
Motivations

Software Productivity Gap

- Lines of code/chip x2 every 10 months
- Transistors/chip x2 every 18 months
- Lines of code/day x2 every 5 years

Source: ITRS & *Hardware-dependent Software*, Ecker et al., Springer

Multicore DSPs – Karol Desnos (kdesnos@insa-rennes.fr)
Hardware Complexity

Source: ITRS System Drivers 2011

Multicore DSPs – Karol Desnos (kdesnos@insa-rennes.fr)
Typical Multiprocessor System-on-Chip (MPSoC)

Heterogeneous MPSoC

- Customized Processors for Specific Functions
- "Generalist" Processors
- On-Chip Interconnect(s)
- Globally Shared Storage Space
- Interfaces with External World

Introduction
Why MPSoCs?

• Frequency Wall
  • Dynamic power consumption is proportional to $f^2$
  • Energy is difficult to extract $\rightarrow$ overheating

• Energy Efficiency
  • 2 cores @ f/2 is better than 1 core @ f
General Design Constraints for Embedded Systems

• Application Constraints
  • Real-time requirements (e.g. video decoding)
  • Reliability constraints (e.g. Anti-lock Braking System)
  • Limited size and power (e.g. handheld systems)

• Cost Constraints
  • Engineering cost
  • Production cost
  • Maintenance cost
  • Recycling cost

• External Constraints
  • Regulation and Standards (e.g. electromagnetic compatibility)
  • Environmental constraints (e.g. temperature, humidity)
Design Challenges for MPSoC-Based Systems

• Exploit architecture parallelism
  • Express application parallelism
  • Balance computational load on PEs

• Hardware/Software co-design process
  • Complex design-space exploration

• Respect constraints
  • Predict/guarantee application performances
  • Reuse legacy code
Typical Development Flow

- C Code
- Command line options
- Compiler
- Simulator + Debugger + Profiler
- OS Core(s)
- Target

10010 Binaries

Introduction
C Language is:

- The preferred language for embedded software
  - C used in 60% of embedded systems in 2009*
  - C++ used in 22% of embedded systems in 2009*

- Good for abstracting core architecture
  - Amount of registers
  - Number of pipeline stages
  - Instruction parallelism
  - Loop optimizations
  - Cache accesses
  - Data representation

- Bad for expressing coarse-grain parallelism
  - Inspired by Turing Machine
  - Global state in a program
  - Unique activity point
  - What is time ?


Multicore DSPs – Karol Desnos (kdesnos@insa-rennes.fr)
Grail of Heterogeneous MPSoCs Programming

Introduction

Multicore DSPs – Karol Desnos (kdesnos@insa-rennes.fr)
Code porting

- Task management
  - Assignment (mapping)
  - Ordering (scheduling)
  - Timing
Code porting

- **Task management**
  - Assignment (mapping)
  - Ordering (scheduling)
  - Timing
Code porting

• Task management
  • Assignment (mapping)
  • Ordering (scheduling)
  • Timing
Code porting

- Task management
  - Assignment (mapping)
  - Ordering (scheduling)
  - Timing
Code porting

- Task management
  - Assignment (mapping)
  - Ordering (scheduling)
  - Timing

- Communication
  - Direct copy
  - DMA
  - Ordering

- Synchronization
  - Polling
  - Interrupts

• Memory
  - Static allocation
  - Dynamic allocation
Code porting complexity caused by:

- **HW resources heterogeneity**
  - PE / Interconnect / Memory

- **Limited HW resources**
  - PE / Interconnect / Memory

- **SW complexity**
  - Complex control dependency
  - Complex data dependency
  - Low predictability
Code porting objectives

• Optimizing / offering trade-offs between:
  • Latency / Response time
  • Throughput
  • Load balancing
  • Memory footprint
  • Power consumption

• Adaptability to any target architecture:
  • DSP
  • GPU
  • HPC

+ Reliability
Programming MPSoCs: A hot research topic!

- IETR work on Multicore DSP Programming

We look at that:
- CAL code and C + πSDF

Also here:
- S-LAM model

Algorithm

Portability

Multicore Compiler

And there:
- Mapping/scheduling
- Memory allocation

And here too:
- Adaptive resource management

Simulator + Debugger + Profiler
Scope of this course

• Present some **state-of-the-art DSP applications and architectures**

• Introduce **MPSocS software programming methods**

• Illustrate the **challenges of multicore processing**
Course Outline

• Lecture 1 – Maxime Pelcat
  • Introduction to the course
  • Applications for MPSoCs

• Lecture 2 – Karol Desnos
  • Languages and MoCs
  • Programming MPSoCs
  • Dataflow MoCs

• Lecture 3 – Maxime Pelcat
  • Hardware Architectures
Course Outline

• Lecture 4 – Karol Desnos
  • Theoretical Bounds
  • Mapping/Scheduling Strategies

• Lecture 5 – Karol Desnos
  • Lab Session
Applications for Multicore Digital Signal Processing
Applications for MDSP

• Overview
• Standardization Processes
• MPEG HEVC
• 4G
Applications for MDSP

Overview

- **Embedded system applications & High Performance Computing (HPC) applications**
  - Base stations & software-defined radio
  - Image and audio processing
  - Industrial control systems
  - Aeronautics & transportation
  - Radar / Sonar
  - Medical
  - Scientific computing & numerical simulation
  - High Performance Computing (HPC)
  - …
Applications for MDSP

Overview

- Embedded system applications & High Performance Computing (HPC) applications

**Multicore DSP boards**

<table>
<thead>
<tr>
<th>Key Feature</th>
<th>Digital Media Processors</th>
<th>OMAP Applications Processors</th>
<th>C6000 Digital Signal Processors</th>
<th>C5000 Digital Signal Processors</th>
<th>C2000 Microcontrollers</th>
<th>MSP430 Microcontrollers</th>
<th>Stellaris 32-Bit ARM Cortex-M3 MCUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automotive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communications</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Industrial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wireless</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key Feature</td>
<td>Complete tailored video solution</td>
<td>Low power and high performance</td>
<td>High performance</td>
<td>Power-efficient performance</td>
<td>Performance, integration for greener industrial applications</td>
<td>Ultra-low power</td>
<td>Open architecture software, rich communications options</td>
</tr>
</tbody>
</table>

Source: Texas Instruments
Applications for MDSP

Overview

• Types of signals
  • Base stations base layer – 1D complex signals
  • Image and audio processing – 1D and 2D+t signals
  • Industrial control systems – 1D and 2D+t signals
  • Aeronautics & transportation – 1D and 2D+t signals
  • Radar / Sonar – 1D complex signals
  • Medical – 1D/ 2D / 3D / ND (e.g. Hyperspectral)
  • Scientific computing & numerical simulation
    High Performance Computing (HPC) – ND signals
  • …
Overview

- **Common operations**
  - Low-pass, band-pass, high-pass and adaptive filtering (FIR and IIR filters)
  - Cross/auto, linear/circular correlation (similarity between signals)
  - Convolution (equivalent to multiplication in Fourier domain)
  - Transformations between domains (Fast Fourier, DCT, Hadamard, wavelet, Hilbert, Wigner-Ville...)
  - Noise removal
  - Power computation
  - Independent component analysis
  - Expected signal detection and extraction
  - Data prediction (temporal, spatial)
  - Entropy coding
  - Complex, vector and matrix operations
  - Forward error correction
  - ...
Applications for MDSP

• Overview

• Standardization Processes

• MPEG HEVC

• 4G
Standardization Process

Motivation 1: putting some order in spectrum usage
Standardization Process

Motivation 2: providing interoperability

Source: Fernando Pescador - HEVC Broadcast Example: H2B2VS Project
Multicore DSPs – Karol Desnos (kdesnos@insa-rennes.fr)
Standardization processes

• Famous standardization organizations regarding signal processing include:
  • ISO (International Organization for Standardization)
  • IEEE (Institute of Electrical and Electronics Engineers)
  • ITU (International Telecommunication Union)
  • 3GPP (Third Generation Partnership Project)

• MPEG HEVC Video Compression Standard
  • Developed by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC JTC1 Moving Picture Experts Group (MPEG)

• 3GPP LTE Radio Telecommunication Standard
  • Developed by the 3GPP (3rd Generation Partnership Project)
  • Respecting (partially) the ITU-R organization IMT-Advanced specification (4G)
Standardization processes

- ISO/IEC Motion Picture Expert Group:

- ISO « International Organization for Standardization » created in 1947
  - Non-governmental organization
  - Creates all kinds of international standards
  - 165 member countries; headquarters in Geneva

  - « Motion Pictures Expert Group » : ISO/IEC Organization created by Leonardo Chiariglione in 1988:
  - 3 to 4 meetings a year
Standardization processes

• The Motion Picture Expert Group:
  • Organized into « national bodies »
  • FNB managed by AFNOR (Association Française de NORmalisation)
    • Often meetings in Rennes or Paris (Orange, Technicolor, …)
    • French delegation in international meetings
  • AFNOR sells the normative documents
  • Gathering people from industry and university
  • Most people come from electrical engineering (TI, Ericsson…)

Standardization Process

MPEG Standards → Looking for more compression

• **Mpeg-1 (1992)**
  • Data rate: 1.5 Mbit/s (675 MBytes/hour)
  • Example: Video storage on Cdrom, Video-CD

• **Mpeg-2 (1994)**
  • Data rate: 5 to 10 Mbit/s (2.3 to 4.5 GBytes/hour)
  • Example: DVD, Digital Video Broadcast

• **Mpeg-4 (since 1998)**
  • Data rate: 10 Kbit/s to 10 Mbit/s (4.5MBytes/hour to 4.5 GBytes/hour)
  • Natural and synthesis data, 3D scenes (too early for the market)
  • Notion of services (interactivity, intellectual property protection …)
  • Example: Mpeg-4 Part 2 (DivX until v5,Xvid)
  • Extensions: AVC (Advanced Video Coding) and SVC (Scalable Video Coding)

• **Each standard : better compression (HEVC: HD@4Mb/s)**
HEVC and 4G: 2 examples of MDSP

Signal Processing

1. Physical Layer
   4G Physical Layer

2. Data Link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer
   HEVC Compression
Applications for MDSP

- Overview
- Standardization Processes
- MPEG HEVC
- 4G
Applications for MDSP

MPEG HEVC

• **Objective:** avoid redundancies in the coded sequence
  • Predict picture from preceding already decoded ones
  • Predict blobs from already decoded ones
  • Same general principles for MPEG-4 Part 2, AVC, SVC, HEVC
  • I (Intra) image, P (Predicted) image, B (Bipredicted) image

Decoding order
Applications for MDSP

MPEG HEVC

- **Objective:** avoid redundancies in the coded sequence
  - Bloc-based compression

- **Inter-Prediction**
  - Motion vector computation
  - The parrot arrived from the right
  - Each block is predicted from the most resembling one in a previously decoded frame.
Applications for MDSP

MPEG HEVC

• **Objective:** avoid redundancies in the coded sequence
  • Bloc-based compression

• **Intra-Prediction**
  • Each bloc is predicted from the most resembling neighborhood.
MPEG HEVC

- Once a bloc is predicted, residual is transformed and quantized (lossy)
  - Eye is less sensitive to high frequencies
    - Protect more low frequencies
    - Quantize more high frequencies

\[
\begin{array}{cccccccccc}
100 & 28 & 131 & 24 & 122 & 101 & 37 & 8 \\
107 & 220 & 49 & 195 & 110 & 28 & 220 & 44 \\
244 & 174 & 55 & 201 & 7 & 14 & 112 & 175 \\
56 & 236 & 14 & 45 & 178 & 52 & 15 & 199 \\
38 & 116 & 170 & 50 & 120 & 244 & 174 & 33 \\
66 & 17 & 127 & 33 & 200 & 28 & 120 & 16 \\
79 & 23 & 80 & 30 & 255 & 5 & 78 & 19 \\
101 & 5 & 33 & 26 & 210 & 17 & 34 & 240 \\
\end{array}
\]

\[
\begin{array}{cccccccccc}
20 & 3 & 10 & 1 & 5 & 2 & 1 & 0 \\
17 & 22 & 5 & 8 & 4 & 1 & 2 & 0 \\
15 & 16 & 4 & 6 & 0 & 0 & 1 & 2 \\
5 & 12 & 0 & 5 & 3 & 1 & 0 & 2 \\
4 & 6 & 8 & 1 & 2 & 2 & 0 & 0 \\
3 & 1 & 5 & 0 & 2 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 2 & 0 & 0 & 0 \\
6 & 0 & 0 & 1 & 3 & 0 & 0 & 0 \\
\end{array}
\]
Applications for MDSP

MPEG HEVC

• Once a residual is quantized, it is entropy coded
  • VLC: Variable Length Coding
    → use less bits to code 0 than 1 than 2 than 3…
  • Simplest: golomb code
    0 → 0 1 bit
    1 → 10 2 bits
    2 → 110 3 bits
    3 → 1110 4 bits
  • Used in AVC: CAVLC & CABAC
  • Used in HEVC: CABAC (using parameter occurrence probability)

• Per macroblock, a decoder receives:
  • Type (I, P, B),
  • Reference picture(s) and vectors if P or B, direction if I,
  • VLC residual
Advanced Video Coding - Decoder

Bitstream processing

Bitstream

VLC

VLC environment

Macroblock Generation

MB data

MB Image processing

Intra Prediction

Current Decoding Slice

Inter Prediction

Decoded Picture Buffer

Sample Reconstruction

Deblocking Filter

Reconstructed Frame

Motion Vectors Reconstruction

Motion Vectors Neighborhood

Inverse Transform

Dequantize

DC Reconstruction

Rescale
### Advanced Video Coding - Profiles

#### Main/High Profile
- **Interlace**
- **B slices** (bidir)
- **I slices**
- **P slices**
- **SI / SP slices**
- **Data partitioning**
- **Slice Groups**
- **Redundant slices**
- **ASO Arbitrary slice ordering**
- **FMO Flexible Macroblock Ordering**

#### Extended Profile (streaming profile)
- **Cabac**
- **CAVLC**

#### Baseline (low latency)
- **AVC Baseline**
  - Low Delay, Lower Processor Load
- **AVC Main/High**
  - Supports Interlaced video, B-Frames, CABAC encoding
- **AVC Extended**
  - Includes Error Resilience Tools, B-Frames

#### Predictions
- Entropy coding
- Buffer management
High Efficiency Video Coding (HEVC) - 2013

- ITU-T/VCEG and ISO-IEC/MPEG: JCTVC
  - ITU H.265 and ISO MPEG-H Part 2
- +50% compression over H.264/AVC for ~perceived quality
  - Subjective assessment
  - Limited complexity increase
  - Fostered parallel processing
- Prepared extensions:
  - Scalable SHEVC,
  - 3D,
  - bit-depths >10 bits…
- Available software:
  - Reference software HM9
  - Open HEVC decoder and CAL decoder (IETR)

Source: ATEME
Applications for MDSP

AVC vs. HEVC: Same structure, more complexity

**AVC**
- 16x16 pixels macroblocks
- 4x4 or 8x8 transform sizes
- Limited partitioning of prediction blocks
- 9 intra prediction directions
- DCT transform
- 1 loop filter: deblocking

**HEVC**
- 64x64 pixels to 8x8 coding units
- 4x4 to 32x32 transform sizes
- Prediction units and transform units with many possible sizes
- 35 intra prediction directions
- DCT transform/ DST in some intra cases
- Extensions: adaptive motion vector prediction, extended intra prediction
- 2 loop filters: deblocking and SAO
- Enhanced parallelism: wavefront: computing coding unit lines in parallel
  tiles: computing sub-images in parallel

Source: ATEME
Applications for MDSP

HEVC Encoder Bloc description

Source: ATEME
Multicore DSPs – Karol Desnos (kdesnos@insa-rennes.fr)

Applications for MDSP

HEVC Decoder Bloc description

VLC
Decoding
Dequantization

Inverse Transform

Filtering

Source: Hervé Yviquel
HEVC and OSI Layers

- The OSI (Open Systems Interconnection) model divides the communication protocols in layers:
  - Implementing one layer without knowing the other layers inner working is possible
  - High level layers can be reused in many communication systems (example: TCP/IP)

- HEVC is designed for transport:
  - Network Abstraction Layer units, containing compressed data for a part of an image, are sent over a network

7. Application Layer
   NAL Units HEVC

6. Presentation Layer
   RTP Payload / DASH / MPEG-TS

5. Session Layer
   RTP / HTTP

4. Transport Layer
   UDP / TCP

3. Network Layer
   IP

2. Data Link Layer

1. Physical Layer
HEVC development effort for a single-core DSP

- Create and optimize MPEG-4 Part 2 decoder
  - 6 MM (men month) for an AVC base version from the standard
  - + 6 MM to handle all the cases
  - Not feasible from the reference softwares: not made to be optimized
    - TM5 for MPEG-2
    - MoMuSyS for MPEG-4 Part 2
    - JM for MPEG4 – Part 10 (AVC, H264)
    - JSVM for Scalable Video Coding
    - HM for HEVC
  - 4 MM from a PC software (Xvid for MPEG-4 Part2, X264 for AVC)
  - 6 MM to optimize the code correctly for a mono-core DSP (DM642 & c6416)
HEVC development effort for a single-core DSP

• Create and optimize a MPEG encoder
  • More choices left to the programmer
    • Can choose to code only a subset of the cases
  • More computationally complex
    • Motion estimation: usually more than half of the coding time
• We can represent signal processing with dataflow

• Data links are very heterogeneous
Applications for MDSP

- Overview
- Standardization Processes
- MPEG HEVC
- 4G
Applications for MDSP

3GPP Standards

- **1G**: 1980, 10kbps
- **2G**: 1990, 100kbps
- **3G**: 2000, 1Mbps
- **4G**: 2010, 10Mbps
- **LTE Advanced**
- **3GPP Standards**
Frequency allocation

- **LTE Frequency allocations in France (ACERP 2011)**
  - 800 MHz band (previously UHF TV bands)
  - 2.6 GHz band

![Frequency Allocation Diagram](source: Wikipedia)

- **Source:** Wikipedia
Applications for MDSP

3GPP

- Generations of Wireless Systems
  - European Telecommunications Standards Institute (ETSI)
    - 1990: Global System for Mobile Communications (GSM)
  - Third Generation Partnership Project (3GPP)
    - 1998: Third Generation Telecommunication System (3G)
    - Members: ETSI (Europe), ATIS(USA), ARIB (Japan), TTC (Japan), CCSA (China) and TTA (Korea)

- 2006: 3G UMTS systems surpassed 100 million users

- 2010: GSM systems reached 3.5 billion users
Applications for MDSP

• 3GPP Releases
  • Phase 1 (1992): GSM
  • Release 97 (1998): GPRS
  • Release 98 (1999): EDGE
  • Release 99 (2000): 3G UMTS
  • Release 5 (2002): HSDPA
  • Release 7 (2007): HSPA+
  • Release 8 (2008): LTE
  • Release 9 (2009): LTE with interoperability, HSPA+ enhancement
  • Release 10 (2011): LTE-Advanced - 4G
Standard alternatives to 3GPP

• **2G Competition**
  • GSM
  • CDMAOne (IS-95)
    • Northern America and Asia, Qualcomm patents

• **3G Competition**
  • 3GPP UMTS
  • CDMA 2000
    • CDMAOne evolution
    • US and Corea, close to UMTS

• **4G Competition**
  • 3GPP Long Term Evolution (LTE)
  • WiMAX
    • Evolution of Wi-Fi, promoted by IEEE
    • Technologies close to LTE
Applications for MDSP

Core Network

Base Station = eNodeB
3GPP LTE release 9

- **High Data Rates**
  - 50Mbps (UpLink),
  - 100Mbps (DownLink)

- **High User Equipment Speed**
  - Optimized up to 120 km/h

- **Reduced Latency**
  - Quick response time (under 5ms)

- **Optimized for packet-switching**
  - Support for VoIP and data

- **Cheap Roll-out**
  - Bandwidth flexibility

- **Up to 100km radius cells (35km for GSM macrocells)**
- **Up to 100 user per cell**
- **Free to consult: search 36.211 and 36.212 in Google**

Applications for MDSP
LTE implements the low OSI layers

- **1. Physical Layer**
  - RS-232, 802.11a/b/g/n..

- **2. Data Link Layer**
  - Ethernet, PPP...

- **3. Network Layer**
  - IP...

- **4. Transport Layer**
  - TCP, UDP, SSL, TLS...

- **5. Session Layer**
  - SSH...

- **6. Presentation Layer**
  - ASCII...

- **7. Application Layer**
  - FTP, HTTP, SMTP, Telnet...

- **physical signal manipulation**...

- **point-to-point and point-to-multipoint control, error correction**...

- **packet fragmentation, logical addressing**...

- **transparent transfers between end-users**...
LTE Base Station Algorithms

1. RACH

2. Uplink: Up to 50Mbps
   Downlink: Up to 100Mbps
LTE Duplex Mode

- **Time Division Duplex**

- **Frequency Division Duplex**

Uplink

Downlink
Uplink SC-FDMA vs Downlink OFDMA

between 1.4 and 20MHz of shared bandwidth

frequency
Applications for MDSP

PRBs allocation example - Uplink

5 MHz Cell
25 PRBs = 300 subcarriers
Typical Case with 4 PUCCH Regions
(PRB pairs 0 to 3)

0 1 2 3
3 2 1 0
0 1 2 3

Frequency Hopping of 12 subcarriers

PUCCH PRBs

PUSCH PBRs

UE1 PUSCH
UE2 PUSCH
Reference Signal
No Emission
PRACH type 0 Burst
Shared PUCCH
Applications for MDSP

**LTE frequency division: subcarriers**

- **Spectrum flexibility** (both downlink and uplink)
  - A Physical Resources Block (PRB): 180kHz for 1 millisecond
  - 14 symbols per ms

<table>
<thead>
<tr>
<th>Bandwidth (MHz)</th>
<th>1.4</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFDM FFT size</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>1536</td>
<td>2048</td>
</tr>
<tr>
<td>Number of available PRBs (downlink)</td>
<td>6</td>
<td>12</td>
<td>25</td>
<td>50</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td>Number of available subcarriers</td>
<td>72</td>
<td>144</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
</tr>
</tbody>
</table>
LTE Multi-Path

Applications for MDSP

Multicore DSPs – Karol Desnos (kdesnos@insa-rennes.fr)
Applications for MDSP

Multi-path side-effects

- Temporal effect of multipath can be ISI (Inter Symbol Interferences)
  - OFDM generates longer symbols

- Multi-path distortion results in fades at precise frequencies
  - FDM dispatches redundant symbols over a wide frequency band

Source: Freescale
Applications for MDSP

PRBs allocation example

Multicore DSPs – Karol Desnos (kdesnos@insa-rennes.fr)
Applications for MDSP

Application is naturally described with dataflow

- **Downlink Data Encoding**
  - CRC/Turbo Coding
  - Rate Matching
  - Interleaving/Scrambling
  - Modulation
  - Multi-Antenna Precoding
  - OFDMA Encoding

- **Uplink Data Decoding**
  - Channel Estimation
  - SC-FDMA Decoding/Multi-Antenna Equalization
  - Demodulation
  - Descrambling/Deinterleaving
  - Rate Dematching/HARQ
  - Turbo Decoding/CRC

- **Symbol Processing**
  - Channel Decoding
  - Turbo Decoding/CRC

- **Symbol Processing**
  - Channel Coding
  - Rate Matching
  - Interleaving/Scrambling
  - Modulation
  - Multi-Antenna Precoding
  - OFDMA Encoding
Applications for MDSP

Static and Variable parts in the LTE algorithm

- Active Users
- Downlink Data per user
- Uplink Data per user
- Downlink Encoding Load
- Uplink Decoding Load
- Preamble Detection Load
Conclusion of Applications for MDSP part

• Applications are complex!
  • A designer should not need to be an expert in both application and architecture
  • Legacy code reuse between systems is absolutely needed
    When a programmer has generated a functional efficient piece of code, he does want to reuse it
  • A designer should not need to tweak his code for his target architecture

• Applications are naturally specified with schematic blocs
  • A similar programming language would be useful!