Virtual Prototype of MuSIC & Design Flow
Scalable MuSIC Baseband Processor

- 4 SIMD Execution Units
- Long Instruction Word
  - Computation Slot (DSP Pipe & IU)
  - Memory Slot
  - Communication Slot
- 4 Interleaved Threads
- Multi-tasked GP core RC1632 to control SIMD
- Local Data Memories instead of L1 Cache

- 3-Level Memory Hierarchy
  - external DRAM/Flash
  - shared memory
  - local memories
- GP Core for L1 Ctrl & MAC
- reconf. accelerators for FIR and Channel Encoding/Decoding
MuSIC Virtual Prototype - Overview

6-Layer System Bus (128 Bit)

2-Layer AMBA AHB (32 Bit)

- IIRCtrl
- DMA
- Bridge
- Bridge
- Special Registers
- Timers
- GPIO
- EBU
- USB
- ESIF
- ARM926 EJS
- Sync Hardware

- RTL-like SystemC
- VaST Models (cycle accurate)
MuSIC Virtual Prototype - Overview

6-Layer System Bus (128 Bit)

ShM 0
6 x 16 K
ShM 0
6 x 16 K
ShM 0
6 x 16 K
ShM 0
4 x 32 K
ShM 0
6 x 16 K
ShM 0
6 x 16 K
ShM 0
6 x 16 K
ShM 1
4 x 32 K
ShM 0
6 x 16 K
ShM 0
6 x 16 K
ShM 0
6 x 16 K
ShM 2
4 x 32 K
ShM 0
6 x 16 K
ShM 0
6 x 16 K
ShM 0
6 x 16 K
ShM 3
4 x 32 K

IRCtrl S
Timers S
GPIO S
EBU S
USB S
M
ARM926 EJS S
D
Sync Hardware S

Sync Bus (8 Bit)

2-Layer AMBA
AHB (32 Bit)

Special Registers S
Timers M M S
GPIO S
EBU S
USB M S
ESIF M

RTL-like SystemC
VaST Models (cycle accurate)
C++ (cycle accurate)
LISATek (same des. used for VHDL gen.)
MuSIC Virtual Prototype - Tools

- Debugging and Profiling of Processors
  - Multi-Core debugger for RC1632
  - Trace file for activities and states of the PEs in each cycle

- Statistics
  - Utilization of the system (e.g. busses), identification of bottlenecks

- Monitors
  - Cycle accurate observation of system (busses, memory, etc.)

- Tracing of thread states (running, blocking) without any changes to the software

- Semihosting
  - Using host’s I/O capabilities (e.g. file I/O)
Benefits of MuSIC Virtual Prototype

- **Hardware Development**
  - Certification of real-time capabilities of MuSIC before silicon
  - Detailed analysis of system w/o
    - expensive on-chip measurement
    - slow HDL simulation
  - Design space exploration for next generation
  - Executable specification for hardware implementation

- **Software Development**
  - Early availability of platform
  - Certification of real-time behavior of software before silicon
  - Easier debugging than on hardware
Software Design & Verification Process

<table>
<thead>
<tr>
<th>Specification of radio standard (e.g. 3GPP Docs for WCDMA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>build functional system model</strong></td>
</tr>
<tr>
<td><strong>based on inherent functional partitioning</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Executable model of the radio standard as C-Program</td>
</tr>
<tr>
<td>(al'tively: CoCentric/COSSAP, SPW, Simulink)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>C functions for signal processing</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Windows</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>mapping to MuSIC architecture</strong></td>
</tr>
<tr>
<td>(threads, SIMD, VLIW, schedule, and assembly)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Multi-threaded C program of the radio standard using ILTOS API</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>C &amp; PE assembly functions for signal processing</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Windows</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>re-compilation &amp; re-linking for MuSIC</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Multi-threaded C program with PE assembly functions of the radio standard using ILTOS on MuSIC</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Virtual prototype on Windows</td>
</tr>
<tr>
<td>Silicon</td>
</tr>
</tbody>
</table>

Functional verification
Concept verification
Bit error rates
Algorithm performance
Capture Timing

Software profiling & mapping
HW & SW partitioning
Synchronization
Performance profiling
Timing estimate

Refinement of mapping, real-time certification

Ramacher / Raab
23.08.2006
System Design – Level 1

Specification of radio standard (e.g. 3GPP Docs for WCDMA)

Paper

build functional system model based on inherent functional partitioning

Executable model of the radio standard as C-Program
(alternatively: CoCentric/COSSAP, SPW, Simulink)

C functions for signal processing

Windows

direct mapping for early demonstration of functionality

FPGA

functional verification
Concept verification
Bit error rates
Algorithm performance
Capture of timing
System Design & VP – Level 2a

mapping to MuSIC architecture
(threads, SIMD, VLIW, schedule, and assembly)

PE assembly code
cycle count of functions

simulation speed
≈ 10 – 20 kHz

run time
≈ 10 sec/func (30 funcs)

execution time of kernel functions on selected architecture components

PE + asm

ARMx

FIR

Mem

T / V

Mem

Mem

cycle-acc.

idealized/
instr.-acc.

Ramacher / Raab
23.08.2006
System Design & VP – Level 2b

Instrumented Multi-threaded C program of the radio standard using ILTOS API

C functions for signal processing

Concurrent Simulation on Windows / SystemC

annotation of execution times

thread tracing and partitioning
Performance profiling
Analysis of parallelism
Realistic traces

definition of threads

realistic traces of thread states

ILTOS

SystemC

thread 1

thread n

equivalent HW model

ARMxxx

ARMxxx

PE

PE

t / V

FIR

synchronization

ideal mem w/o conflicts

cycle-acc.

idealized/ instr.-acc.

simulation speed
≈ 1 MHz

run time
≈ 30 x 30 sec
(30 test cases)
System Design & VP – Level 2c

<table>
<thead>
<tr>
<th>thread 1</th>
<th>thread n</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILTOS ↔ Windows</td>
<td>ILTOS ↔ Windows</td>
</tr>
</tbody>
</table>

Data synchronization of multiple threads

Functional verification of complete multi-threaded application

Run time

≈ 0
System Design & VP – Level 3

- Timed accelerators, busses, memories
- Simulation speed: ≈ 50 – 100 kHz
- Run time: ≈ 30 x 10 min

Re-compilation & re-linking for MuSIC

Multi-threaded C program with PE assembly functions of the radio standard using ILTOS on MuSIC

Instruction-Accurate VP with Timed Accelerators/Bus/Mem
System Design & VP – Level 4a

- Multi-threaded C program with PE assembly functions of the radio standard using ILTOS on MuSIC
- Cycle-Accurate Virtual Prototype
- Refinement of mapping, real-time certification

Cycle-Accurate Virtual Prototype

Multi-threaded C program with PE assembly functions of the radio standard using ILTOS on MuSIC

Simulation speed: \( \approx 5 \text{ – } 20 \text{ kHz} \)
Run time: \( \approx 30 \times 50 \text{ min} \)
Cycle-accuracy
Multi-threaded C program with PE assembly functions of the radio standard using ILTOS on MuSIC

Cycle-Accurate Virtual Prototype

Silicon
System Design & VP – Level 5

cycle-accuracy

simulation speed
≈ 300 Hz

run time
≈ 30 x 28 h
Throughput of Abstraction Levels

**real-time**
(8 SIMD cores @ 300 MHz,
4 GP cores @ 300 MHz,
1 ARM @ 200 MHz,
accelerators)

WCDMA protocol stack, L1 ctrl, L1 DSP
e.g. 30 test cases, 100 ms each,
for real-time certification

**level 2a – profil. & map.**
components 10 kHz
run time for 30 functions: 30 x 10 sec

**level 2b – thread traces**
SystemC 1 MHz
run time for 30 test cases: 30 x 30 sec

**level 3 – debug & perf.**
instr.-acc. VP 50 kHz
run time: 30 x 10 min

**level 4 – real-time certif.**
cycle-acc. VP 1 MHz
run time: 30 x 30 sec

**level 5 – real-time certif.**
cycle-acc. VP 300 Hz
run time: 30 x 28 h
## Throughput of Abstraction Levels

**real-time**
- (8 SIMD cores @ 300 MHz, 4 GP cores @ 300 MHz, 1 ARM @ 200 MHz, accelerators)

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Example</th>
<th>Run Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2a</td>
<td>profil. &amp; map. components 10 kHz</td>
<td>WCDMA protocol stack, L1 ctrl, L1 DSP</td>
<td>run time for 30 functions: 30 x 10 sec</td>
</tr>
<tr>
<td>2b</td>
<td>thread traces</td>
<td>e.g. 30 test cases, 100 ms each, for real-time certification</td>
<td>run time for 30 test cases: 30 x 30 sec</td>
</tr>
<tr>
<td>3</td>
<td>debug &amp; perf.</td>
<td></td>
<td>run time: 30 x 10 min</td>
</tr>
<tr>
<td>4</td>
<td>real-time certif.</td>
<td></td>
<td>run time: 30 x 30 sec</td>
</tr>
<tr>
<td>5</td>
<td>real-time certif.</td>
<td></td>
<td>run time: 30 x 28 h</td>
</tr>
</tbody>
</table>

Note: BeeCube System Emulator

Ramacher / Raab
23.08.2006
Requirements to a Research Accelerator

- support of different abstraction levels
- generation of cycle-accurate from abstract architecture models and vice versa
- integration of existing IP (SystemC/C++)
  - processor simulators
  - accelerators
  - bus models / protocols
  - memory models
- design space exploration wrt.
  - processor types (SIMD, VLIW) and instruction sets
  - ASIPs
  - accelerators
  - communication architecture