Parallel Programming Models & Platforms
Application to Multimedia

Pierre Paulin, Director
SoC Platform Automation Technologies
STMicroelectronics
Central R&D, Ottawa, Canada
SoC Tools

- Proven, established programming models
- User-defined parallelism

Parallel programming models

- Keep it simple, regular, predictable
- Use industry standards: Processors, NoC, I/O
- Simplify use of legacy architectures

Application S/W

- Lightweight mapping tools, H/W RTOS
- Standard simulation & analysis tools
- User-defined analyses

FlexMP SoC Platform

MultiFlex SoC Tools

MPSoC, July 2004
Outline

- FlexMP architecture platform
- MultiFlex Tools and Methodologies
  - MP-SoC compilation, H/W O/S
- Applications
  - MPEG4 video codec
  - 10 Gb/s IPv4 packet forwarding
  - 2.5 Gb/s traffic manager
  - 3G basestation
FlexMP SoC Platform

- Multi-threaded, multi-processor platform
- Popular processor models w. config. extensions: H/W multithreading and pipeline depth

MPSoC, July 2004
MultiFlex MP-SoC Platform Tools

- Two parallel Programming Models
  - DSOC: Message passing
  - SMP: Shared memory

Application to platform mapping

- H/W message passing, IP Plug and Play
- H/W MP-O/S scheduler accelerators
MultiFlex Message Passing

- MP programming models

Message Passing
H/W Accelerator
Neutral Data Format, Standard NoC I/F

Message passing
IP Plug and Play

Executable Spec

Conf. Proc.
mem
fpga

Conf. Proc.
mem
SoG

NoC

I/O

I/O

eMEM
H/W O/S Schedulers

eSoG/eFPGA

MPSoC, July 2004
MultiFlex H/W O/S

- Manage med-grain concurrency (~100 instr)
  - Fault tolerance
  - Future: Manage power, QoS

Executable Spec
- MP programming models

MPSoC, July 2004
Message Passing Model: DSOC (Distr. System Object Component)

Based on leading distributed S/W concepts
- E.g. CORBA, DCOM

Objects represent application functionality

Inter-object communication via standard I/F
- Use of lightweight Interface Description Language

Platform independent, no mapping assumptions
DSOC to Platform Mapping

- S/W-S/W com
- S/W-H/W com
- Message
- Passing Engine

- Synthesis from IDL:
  - Auto generation of
    - Drivers between different O/S
    - Drivers between S/W PE’s and network-on-chip
    - Glue logic between H/W PE’s and NoC

MPSoC, July 2004
DSOC Platform

Max processor-processor message passing rate:
35 MHz (500 MHz clk)
15 MHz (200 MHz clk)
<15 instructions

MPSoC, July 2004
Programming Model 2: SMP

- Symmetric multi-processing with shared-memory
  - Complement to DSOC programming model
    - DSOC object may have SMP internal implementation
  - SMP is more natural MP model for Multimedia

- SMP Nano-kernel written in C and C++
  - Java/C# style concurrency primitives implemented with C++ API (or C Posix API)
  - Hardware O/S assists in implementation
SMP Platform

Fork 1~256 threads:

10 instructions (50ns @200MHz)

+ 12 cycles/thread (in conc. engine)

H/W Concurrency Engine

Semaphore  Run Queue

Monitor

Entry List

Condition

Network-on-Chip

NoC I/F

MPSoC, July 2004
SMP/DSOC to Platform Mapping

MPSoC, July 2004
Outline

- FlexMP architecture platform
  - Multi-threaded processors, Flexible H/W
  - Network-on-Chip (NoC) interconnect
- MultiFlex Tools and Methodologies
  - Multi-Processor SoC analysis and debug tools
  - MP-SoC compilation, H/W O/S
- Applications
  - MPEG4 video codec
  - 10 Gb/s IPv4 packet forwarding
  - 2.5 Gb/s traffic manager
  - 3G basestation

MPSoC, July 2004
MPEG4 Codec Exploration

- 30 frame/sec, VGA resolution (4.1 GIPS required)
- High-level model using SMP and Message Passing

### Hardware/Software Trade-offs

<table>
<thead>
<tr>
<th>H/W (80% perf.)</th>
<th>S/W: 5 RISC, 4 threads (88% Utilization)</th>
<th>Coproc: Clip Div Abs Sgn</th>
</tr>
</thead>
<tbody>
<tr>
<td>S/W: 15 RISC, 16 threads (75% Utilization)</td>
<td>Coproc: Clip Div Abs Sgn</td>
<td></td>
</tr>
<tr>
<td>H/W (65% perf.): DCT, SAD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **95% Lines of Code in S/W**
- **96% Lines of Code in S/W**

- Off-the-shelf appln. code
- ARM7 RISC @ 200MHz
- Simple memory arch.

**MPSoC, July 2004**
Load balancing

- The total load average is about 88%
- The load is well balanced over the 5 ARM.s thanks to concurrency engine
Execution speed up

![Graph showing execution speed up with different number of ARM threads.]

- **Number of ARM threads:** 2, 4, 8
- **FPS (Frames Per Second):** 5, 10, 15, 20, 25, 30, 35
- **Latency:** 0

**Graph Labels:**
- 8 threads
- Theoretical
- 2 threads
- 4 threads
- Latency = 0
Cache analysis

![Graph showing cache miss ratio and FPS for cache sizes ranging from 0 to 16 KB. The graph includes lines for cache miss, FPS with 8 threads, and FPS with 2 threads.](image)

- **Cache size (KB)**
- **Cache miss ratio (%)**
- **FPS**

- **Cache miss**
- **FPS: 8 threads**
- **FPS: 2 threads**
Use of H/W load balancing engines (CE and HORBA)

- Only 3.8% data bandwidth overhead
RESULTS:

- 85-92% PE utilization
- Msg passing code <20%
3G Base Station Platform Exploration

- DSOC Objects
  - Channel Coding
  - Interleaver / Deinterleaver
  - CRC
  - Code block segmentation
  - 2nd interleaver deinterleaver
  - Spreading

- NOC
  - Configurable number of ARMs and H/S threads

- Hardware (H/W)
  - Turbo decoding
  - Viterbi decoding
  - HORBA
  - MEM
  - DMA
  - I/O
  - StepNP

- Software (S/W)

MPSoC, July 2004
MultiFlex MP-SoC Tools: Summary

Value-added:
- Platform independent eS/W
- Platform scalability
- High PE utilization (85-97%)
- Ease of programming

Executable Spec
- MP programming models

Application to platform mapping

- Multi-media
- Networking
- 3G base station

MPSoC, July 2004