MAPS Project

Weihua Sheng, Jianjiang Ceng,
Jeronimo Castrillon, Anastasia Stulova,
Stefan Schürmans, Rainer Leupers

Software for Systems on Silicon
RWTH Aachen University

Mapping of Applications to MPSoCs
November 27th 2008, Düsseldorf

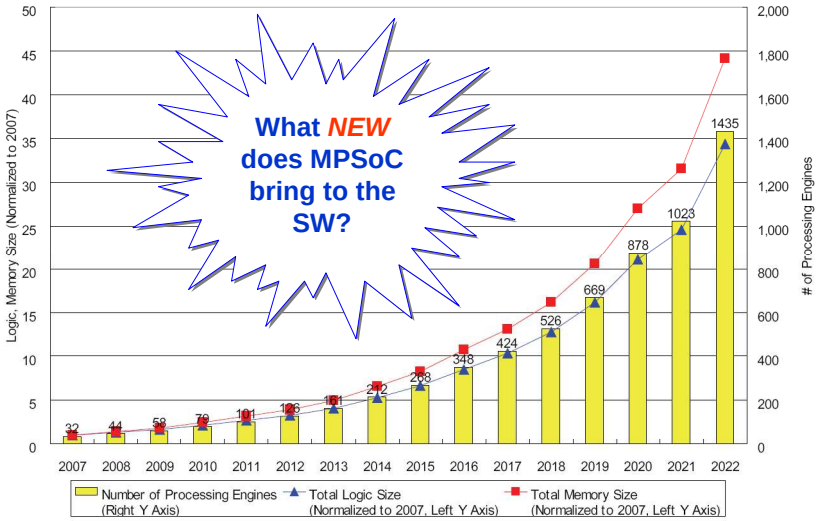
Institute for Integrated Signal Processing Systems

Agenda

MAPS - MPSoC Application Programming Studio

- MAPS Project Motivation/Overview
- MAPS-TCT Framework Demo
- MVP (MAPS Virtual Platform) Demo

MPSoC Age Has Come



SoC Consumer Portable Design Complexity Trends (Source: ITRS 2007)

MPSoC Application Drivers

- **Versatility: we have *MANY* applications running on the MPSoC.**

Portable/Consumer

Source: Chen, NTU, MPSoC 2008

Automotive

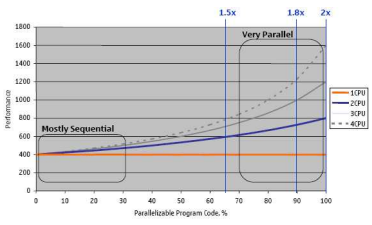
Source: NXP, MPSoC 2008

- **Complexity: and at the *SAME* time.**

Example: Parallel Usage Scenario

"Advanced end-user Ari does multiple things at the same time."

- Beginning:
  1. **Streaming audio** from XM Radio.
  2. **Browsing** website www.CNN.com (very tricky Java/tables/css).
  3. **Recording news video** received from DVB-H.
- Suddenly:
  4. **Push email** downloads 1MB JPEG image (background).
  5. **Voice call** comes in.
- At the same, time many OS features are used in parallel:
  VoIP stack, HTTP/TCP/IP stack, Bluetooth stack, WLAN driver, telephony, MP3 decode, RTP/UDP stack, Java virtual machine, window server, fileserver, etc.
- Requires high performance peak. Overload shows as bad user experience for the foreground application.
- Any system stutter or unresponsiveness considered harmful.



Source: Nokia, MPSoC 2006

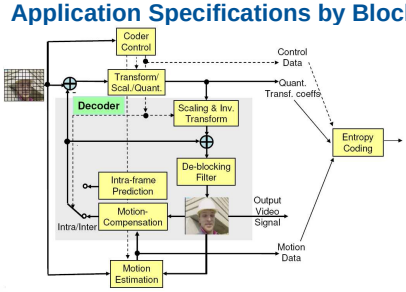
Mixed MPSoC Programming Models/Languages

- **Programming Models/Languages Currently Used/Evaluated (Percentage of Survey Respondents Identifying Each)**

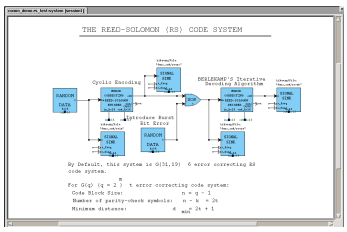
	N	Users of Multiple Discrete Processors, N=108	Users of Multicore, N=115	Users of Both, N=123
Normal C/C++	307	88.0%	88.7%	89.4%
Titanium (Java)	42	13.0%	11.3%	12.2%
OpenMP	27	13.0%	4.3%	6.5%
MPI	18	10.2%	3.5%	8.9%
Unified Parallel C	23	7.4%	3.5%	8.9%
CAF (Fortran)	4	2.8%	0.0%	0.8%
Other	28	7.4%	6.1%	10.6%

Source: VDC (Multicore Expo 2008)

- **Application Specifications by Block Diagrams/etc.**



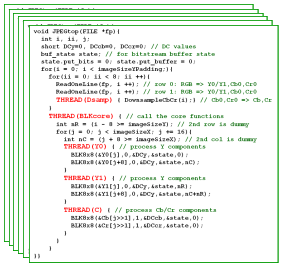
H. 264 Encoder Block Diagram

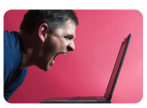


Efficient/Fast Multi-task Processing

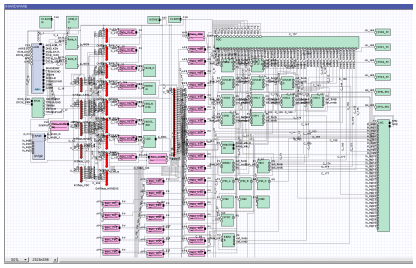
Keys to success of embedded real-time systems

- **Smart spatial/temporal task mapping**
- **Efficient task-scheduling functionalities**

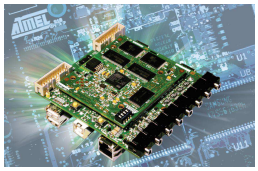




Mapping



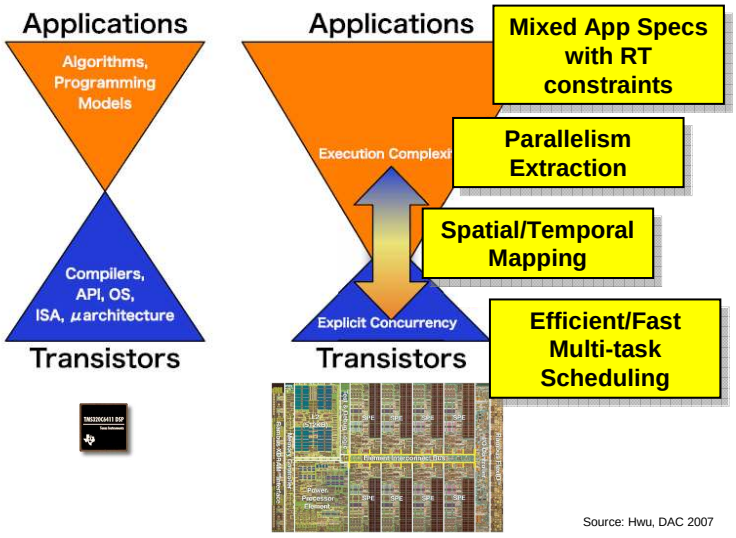
Source: Virtual Platform of Shapes RDT, RWTH Aachen



Source: Diopsis940, Atmel

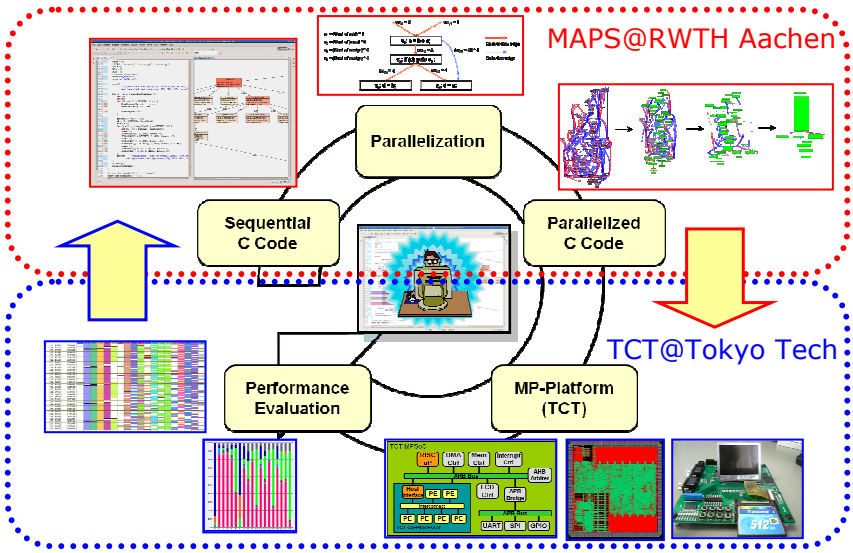
Summary – embedded SW from Uni-core to Multi-core

Multi-core brings more software complexity in multi-orders!



Source: Hwu, DAC 2007

MAPS-TCT Framework

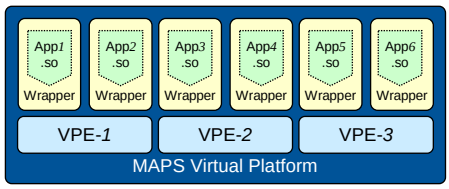


Ceng et al, „MAPS: An Integrated Framework for MPSoC Application Parallelization“, DAC 2008

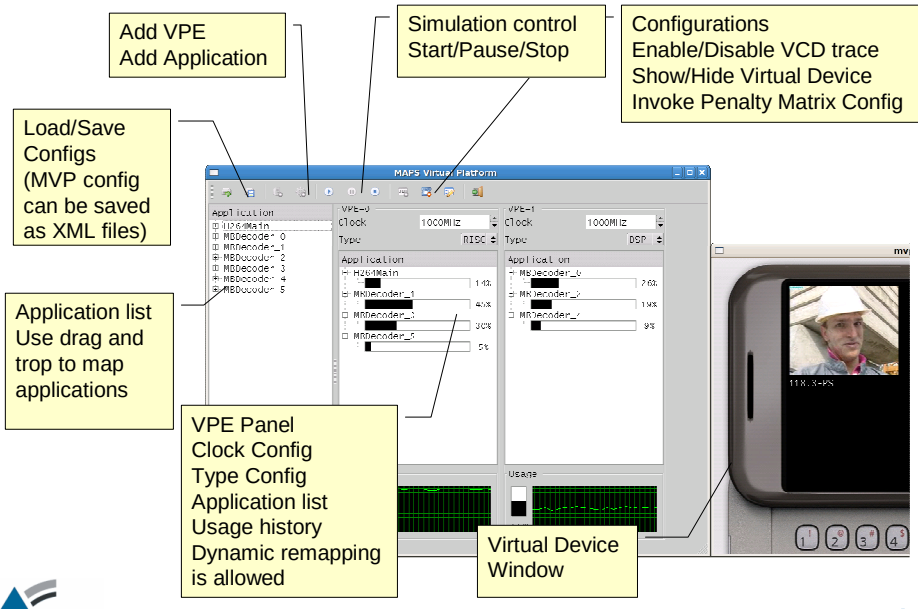
MAPS-TCT Demo

MAPS Virtual Platform (MVP)

- **A SystemC based high-level virtual platform**
  - No instruction-set-simulator is employed
  - Simulate software natively
- **Processing elements are modeled with abstract Virtual Processing Elements (VPEs)**
  - Include a simple scheduler to enable temporal mapping
  - Multiple VPEs can used to model MPSoC
- **Run applications written in C**
  - Application source code is decoupled from the simulator
  - No SystemC code is needed from the MVP user



MVP GUI



MVP Demo

Summary of the MVP

- **A ready-to-use high-level virtual platform is demonstrated**
  - No SystemC coding is needed
- **Applications are developed standalone independent from the simulator**
- **Multiple applications can be simulated simultaneously in the platform**
- **A simple scheduler is currently included to support temporal mapping**
- **Spatial mapping exploration is supported through**
  - Configuration files created by the MVP user
  - Dynamically drag-and-drop tasks during the simulation
- **Still a work in progress**
  - Implement more sophisticated scheduling algorithm
  - Integrate the MVP into the MAPS programming environment

Thank you!