HW-SW Interfaces CoDesign for Multi-Processor SoC

Dr. Ahmed Amine JERRAYA
TIMA Laboratory
46 Avenue Felix Viallet
38031 Grenoble Cedex France
Tel: +33 476 57 47 59
Fax: +33 476 47 38 14
Email: Ahmed.Jerraya@imag.fr
Defining HW-SW Interfaces

- Application SW Designer: A set of system calls used to hide the underlying execution platform. Also Called Programming Model
- HW designer: A set of registers, control signals and more sophisticated adaptors to link CPU to HW subsystems.
- System SW designer: Low level SW implementation of the programming Model for a given HW architecture.
- Assumes HW is ready de start low level SW design
- CPU is the ultimate HW-SW Interface

SOC requirements
- HW-SW interfaces tradeoff
- Programming model Abstracts both HW and SW interfaces in addition to CPU
Classical SW design flow to interface HW

- Programming Model: Abstract HW at Different level

- Discontinuities:
  - Compilation: Generally ignore the CPU environment (Interrupts, Complex I/O)
  - Sys.lib: adapt for different HW
  - MMAP: Adapt to different CPU-memory architecture
  - User.lib: to make the flow efficient for the application
Parallel Programming Models for SW Design

- Explicit concurrency, decomposition, mapping; Implicit communication, synchronization, Interconnection and Interface
  - SDL, compositional C++

- Explicit concurrency, decomposition, mapping, communication, synchronization; Implicit Interconnection and Interface
  - MPI, TLM Message, thread package, concurrent C

- Explicit concurrency, decomposition, mapping, communication, synchronization, Interconnection; Implicit CPU
  - Programming with OS services e.g. POSIX threads.

- Explicit concurrency, decomposition, mapping, communication, synchronization, Interconnection and CPU; Implicit MMAP
  - ISA SW
Joint HW/SW Interfaces abstraction requires different programming Models

- Implicit : CPU SS organization
  - SystemC 3.0: SW Native, BFM interface or CPU SS simulation, TLM or RTL HW

- Implicit CPU organization
  - BCA: SW Bin, BFM interface or CPU ISA simulation, TLM or RTL HW

- Implicit Physical addressing (MMAP, Booting address, …)
  - MAXSIM: SW Bin, CPU ISS cycle accurate with explicit Memory, TLM or RTL HW

- All explicit
  - Bin SW, RTL CPU + RTL HW
System specification is a virtual architecture: virtual modules using specific programming models connected through an execution environment.

Architecture implementation: heterogeneous components and sophisticated HW/SW interfaces.

HW/SW interface codesign requires a unified model for HW, SW and CPU sub-system.
Conclusion

- Classical Programming models separate HW and SW interfaces
- SoC Programming Model abstract CPU in addition to both HW and SW
- Existing HW/SW interface Models
  - Cosimulation execute SW as a HW module
  - Formal methods abstract both SW and HW to a single model, exclude CPU
- HW/SW Interfaces codesign requires to invent a Unified model to abstract HW, SW and CPU [Petrot]
Thank You