Design space exploration in Multi-Processor System-on-Chip platforms

Computer Science Research Seminar - Qualifying Examination

Candidate: Nikolina Frid
Faculty of Electrical Engineering and Computing

Mentor: prof. Vlado Sruk, PhD

June 2014
Contents

• Introduction

• MPSoC Design

• MPSoC Modelling & Synthesis

• Design Space Exploration

• Conclusion
Introduction

• Embedded systems:
  – Ubiquitous, programmable, long-lived
  – Requirements: balance development and production costs and time with performance and functionality gains [Kreutzer, 2000]

• New type of embedded system - MPSoC
  – Heterogeneous set of components – high performance with minimum hardware (cost effective)
  – Key issues [Park, 2009]:
    • determine a target multicore platform [design space exploration (DSE)]
    • exploit the application parallelism

• Automation of design process – lowers cost and time to market
MPSoC Design
Abstraction Levels and Methodologies
System Design

• System design [Gajski, 2009]
  – Define architecture, components, modules, interfaces, and data
  – Satisfy specified requirements
  – Complex task – sequence of steps (specification, implementation, verification, refinement)
  – Refine higher levels of abstraction to lower levels

• Aspects of heterogeneous multiprocessor architecture [Wolf, 2007]:

Productivity Gap
Abstraction Levels - Y Chart

**Behaviour**
- black box

**Structure**
- components & connections

**Physical**
- size & position

**Logic**
- gates & flip-flops

**Processor**
- datapath, interfaces

**System**
- processors, memories, buses
System Design Methodologies

• Bottom – up [Gajski, 2009]
  – Parts are built before assembling the whole product
  – Library has functional, structural and layout models for each component

• Top – down [Gajski, 2009]
  – Does not attempt a component/system layout until the entire design is finished
  – Starts with model of computation (highest level)

• System Level Design [Sangiovanni-Vincentelli, 2007]
  – Key concept: system level specification
  – Platform Based Design - most promising approach to implementation
    • „meet in the middle” approach
    • family of potential solutions
    • key step: mapping functionality to the architecture
MPSoC Modelling & Synthesis
System Level Modelling & Specification
System Level Modelling

• Model [Jantsch, 2004]:
  – A simplification of another entity
  – Does not contain any other characteristics than those relevant for the task

• System level model [Kreutzer, 2000]:
  – Represents application behaviour, architecture characteristics, and the relation (e.g., mapping, hardware-software partitioning) between application(s) and architecture.
  – At a high level of abstraction - allows early verification & estimations on performance, power consumption and cost

• 3 classes of models [Gajski, 2009]:
  – Specification model – behaviour
  – Transaction level model (TLM) – communication & computation
  – Cycle accurate model (CAM) – microarchitecture
Application Specification

- Set of concurrent, hierarchical processes that operate on and exchange data via variables and channels – **functionality** [Marwedell, 2011].

- **MoC** (Model of Computation) [Wolf, 2007]
  - Decomposition of behaviour into pieces and their relationships
  - **State-based**: focus on control flow
    - Driven by actions, events
    - e.g.: Finite State Machines
  - **Process-based** (actor-based): data oriented
    - **Actors** (modules) connected through **channels**
    - Expose flow of data and dependencies between actor executions
    - Untimed, ordered by data dependencies
    - e.g.: Data Flow Graphs, Kahn Process Networks,
  - **Concurrent models**:
    - Describe time as **partially ordered**
    - e.g. Task Graphs, Petri Nets
Kahn Process Network (KPN)

- Set of processes that communicate through a network of unidirectional communication channels [Kahn, 1974]
Task Graph

- Directed Acyclic Graph (DAG) – a generic model of a parallel program consisting of a set of *partially ordered* processes [Kwok, 1999].
Platform Specification

- Set of system components connected by a network of buses – realizes system behaviour \cite{Wolf, 2007}

- Library of components \cite{Gajski, 1996}

- Multiprocessor structure representation:
  - Graph (XML) \cite{Nikolov, 2009}
  - Architecture Analysis & Design Language (AADL) \cite{Feiler, 2006}
Design Space Exploration

Task partitioning and mapping
Design Space Exploration (DSE)

• Find out *near-optimal system architectures* for a given application, considering the design constraints and objectives [Marwedell, 2011].

• Involves:
  – Architecture and processing element selection
  – Task partitioning and mapping
  – Performance estimation before HW prototype is built
Software partitioning and mapping

  - Assignments of task sets to processors
  - Execution order of tasks on the same processor
  - Keep precedence constraints

- Mapping and scheduling in heterogeneous multiprocessor systems - **NP-hard optimization problem** [Branke, 2008]

- Traditionally solved using Linear Programming
  - Complex development, high computational cost
  - Suitable only for small problem instance

- Other solutions [Vivekanandarajah, 2008]:
  - Non-deterministic methods:
    - **Evolutionary algorithms** - take very long time for more complex problems
  - Deterministic methods:
    - **Greedy algorithms** - very fast, but provide suboptimal solution
Existing solutions

- Metropolis (UC Berkeley) [Balarin, 2003]:
  - PBD
  - Application modelling: MMM – proprietary metamodelling language
  - Architecture modelling: processes and resources
  - General framework and language for modelling, simulation, validation and analysis
  - No support for automatic DSE
Existing solutions (2)

- Daedalus (Univ. Amsterdam) [Nikolov, 2009]
  - SLD environment
  - Sesame modelling and simulation environment [Thompson, 2007]
- Y-Chart
- KPN + XML platform specification
- automatic DSE – SPEA2
Conclusion

- Design of heterogeneous MPSoCs
  - Costly & time consuming
  - Automation is a must!

- DSE – key for success
  - Simulation & performance estimation must be done in early stages
  - Automation of partitioning & mapping software in focus
References


References


Q&A

• ...