Introduction to Hardware-dependent Software Design

ASPDAC 2009 Special Session

Hardware-dependent Software for Multi- and Many-Core Embedded Systems

Rainer Dömer
UC Irvine, California
USA

Andreas Gerstlauer
Univ. of Texas at Austin
USA

Wolfgang Müller
University of Paderborn
Germany

Introduction

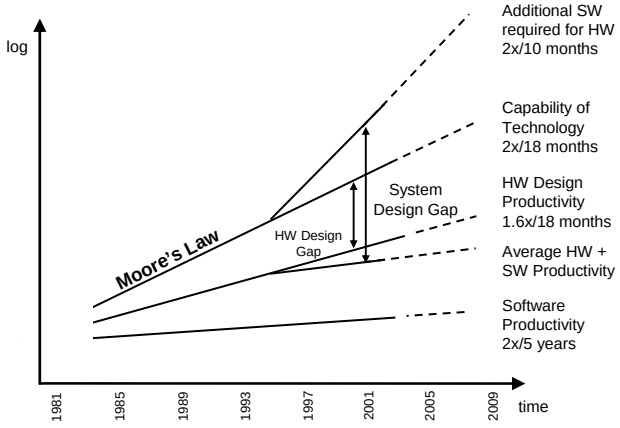
- Embedded System Design
  - Rising system complexities
  - Rapidly increasing software content
  - Domination of embedded software
  - Special attention to hardware/software interface
- Hardware-dependent Software (HdS)
  - Gained relevance in recent years due to
    - Flexibility
    - Possibility of late change
    - Quick adaptability
  - Importance already observed by VSIA in 2002

Motivation

- Design Productivity Gap
  - Hardware productivity gap
    - Capacities in chip size outpace capabilities in chip design
    - Moore's law: chip capacity doubles every 18 months
    - HW design productivity estimated at 1.6x over 18 months
  - Software productivity gap
    - Growth of SW productivity estimated at 2x every 5 years
    - Needs in embedded SW estimated at 2x over 10 months
  - System productivity gap
    - HW gap + SW gap

Motivation

- Design Productivity Gap



(source: Ecker et. al. [3]).

Motivation

- Design Productivity Gap
  - Hardware productivity gap
  - Software productivity gap
  - System productivity gap
    - HW gap + SW gap
- Additional complexity
  - Close interaction and tight dependency between HW and SW

➢ Hardware-dependent Software is at the core of the system design challenge!

Hardware-dependent Software

- Definition:
  Hardware-dependent Software (HdS) is the software in an embedded system that closely interacts with the underlying hardware platform.
- Specifically
  - HdS is built specifically for a particular HW block
    - HdS is meaningless without the HW
  - HdS and HW together implement the core functionality
    - HW is meaningless without the HdS

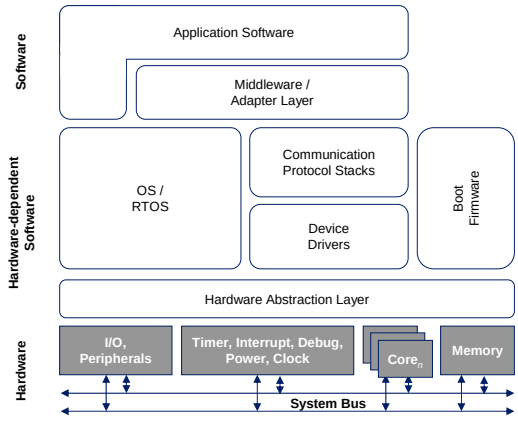
Hardware-dependent Software

- HdS is part of a Layered Software Architecture
  - HdS is low-level software
  - HdS provides application software with an interface to hardware features

➢ HdS is a software layer between the application software and the underlying hardware platform

Hardware-dependent Software

- Layered Software Architecture



(based on Ecker et. al. [3]).

Developing HdS

- Typical Embedded SW development
  - Dedicated C/C++ development environments
  - Targeted tool chains
    - Cross-compiler
    - Target-specific assembler and linkers
    - Debuggers
    - Linters
  - Customization for embedded software
    - Intrinsics, pragmas, inline assembly
- Development
  - Most often manual
  - Tedious
  - Error-prone

Developing HdS

- Goals
  - Move to higher level of abstraction!
  - Utilize automation!
    - Eliminate manual coding, debugging, and validation
- Advanced approaches
  - Model-based design
  - Code generation
    - Automatically generate low-level code from abstract, high-level description
  - Software synthesis
    - Automatically generate device drivers, protocol stacks, and entire application software

HdS for Multi- and Many-Core Platforms

- Moving beyond Single-Core Architectures
  - Venture Develoment Corp. (VDC) projects a 6 times increase of multi-core microprocessors between 2007 and 2011
    - Multi-core (2-10 cores)
    - Many-core (tens, hundreds, thousands of cores)
- Growing variety of system architectures
  - Multi-processing
    - Symmetric, homogeneous
    - Asymmetric, heterogeneous
  - Operating System (OS / RTOS)
    - Single shared OS with common HdS stack
    - Multiple / independent OS

➢ HdS design is a growing challenge!

Special Session Outlook

- Semiconductor Perspective:
  - *"Using a Dataflow abstracted Virtual Prototype for Hardware-dependent Software Design",* Michael Velten et.al, Infineon Technologies AG, Germany
- Viewpoint of a consumer electronics manufacturer
  - *"Needs and Trends in Embedded Software Development for Consumer Electronics",* Yasutaka Tsunakawa, Sony Corp., Japan
- Potential solution
  - *"Hardware-dependent Software Synthesis for Many-Core Embedded Systems",* Samar Abdi et.al., CECS, UC Irvine, USA

Additional Information…

- **Hardware-dependent Software Principles and Practice**
  - Edited by
    - Wolfgang Ecker
    - Wolfgang Müller
    - Rainer Dömer
  - Springer, Feb. 2009
  - ISBN: 978-1-4020-9435-4
  - Approx. 310 p., Hardcover