Hardware-Software Co-Design Introduction

EE8205: Embedded Computer Systems
http://www.ee.ryerson.ca/~courses/ee8205/

Dr. Gul N. Khan
http://www.ee.ryerson.ca/~gnkhan
Electrical and Computer Engineering
Toronto Metropolitan University

Overview

- Traditional Embedded System Design
- Hardware-Software (HW/SW) Co-design
- Co-specification and Cosynthesis
- HW/SW Partitioning

Introductory Articles on Hardware-Software Codesign available at the course webpage, Part of Chapter 7 of the Text by Wolf

Introduction

Embedded computers are the processing devices.

- Home appliances and entertainment units
- Transportation including automobiles
- Medical instrumentation
- Wireless communication devices,
- Jet engines and other aerospace/space application
- Industrial control, nuclear systems and many more

By many estimates embedded computers make up 99% of worldwide computers

Embedded Computer Systems are the ideal candidate for hardware-software codesign.

Embedded System Design

- Separate HW and SW design has been explored and examined very thoroughly
- Joint design remains an area of rapidly growing study
- Old embedded devices always built from scratch
  - within reasonable amount of time
- Components - smaller and faster - IP cores
- Tools required for the product engineer.

Embedded System Architecture Design

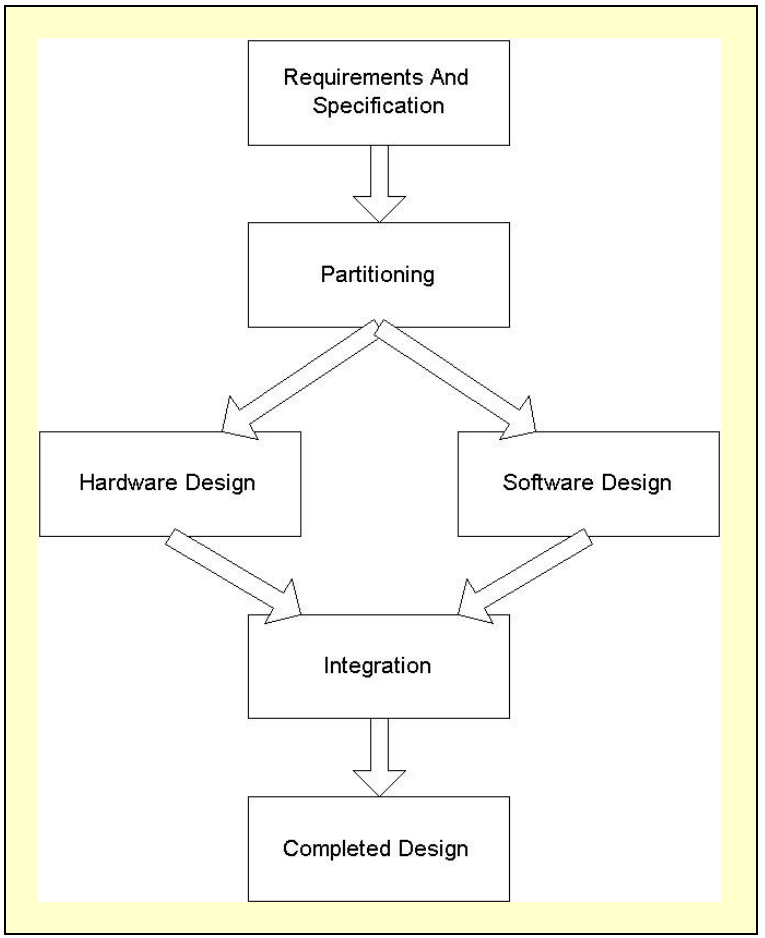
- Real-time System Design
  - Performance analysis
  - Scheduling and allocation
- Accelerated systems
- Use additional computational unit dedicated to some functions?
  - Hardwired Logic e.g., FPGA
  - Multiple processing elements (PEs) or an extra CPU
- **Hardware/software co-design**: a joint design of hardware and software architectures of Embedded System.

Traditional Design Practices

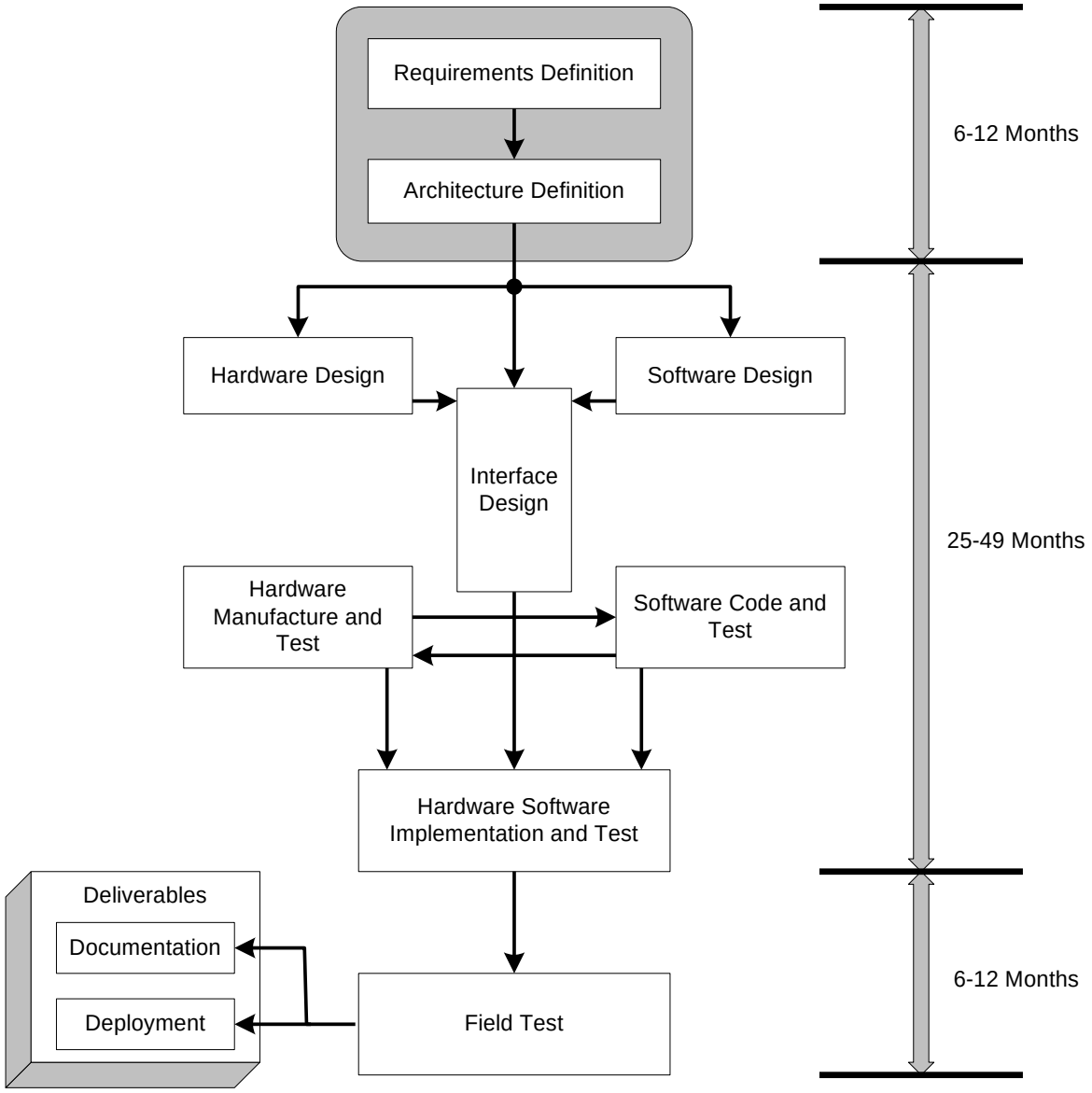
- **Performance Requirements make it impossible to execute entire application in software**
- **Intensive portions are extracted and realized as custom hardware**
- **Early Design Cycle Partitioning**
  - **Design Space is not fully Explored**
  - **High-Cost Design**
  - **Inefficient**

Traditional Embedded System Design

- HW/SW Partitioning performed at an early stage.
- Design mistakes have huge negative effect
- Inability to correct mistakes performed at the partitioning phase



Traditional Design Practice



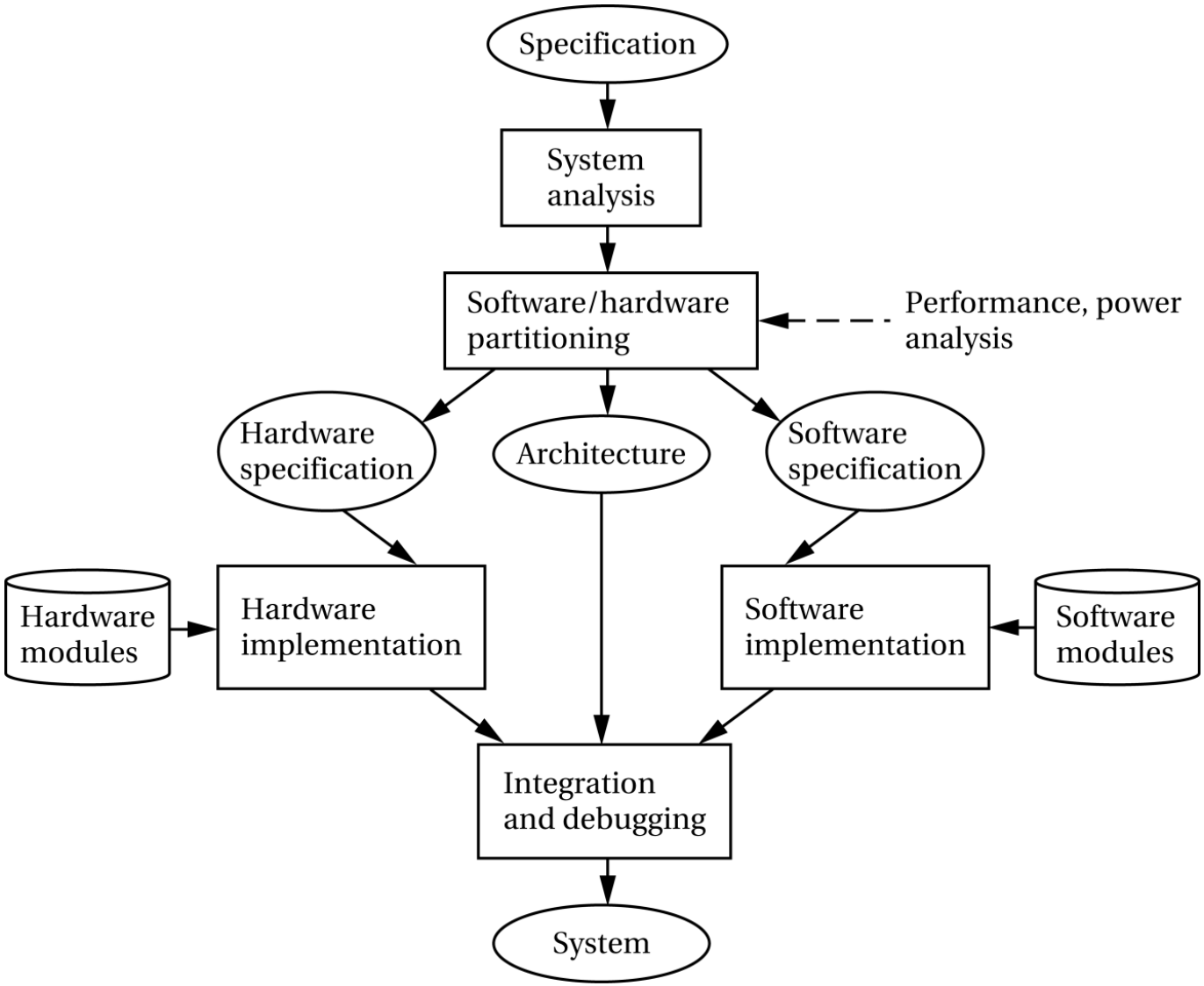
Advancements

- VLSI Technology
  - Smaller, Faster IP Cores
  - Reconfigurable Logic
- Matured Hardware Design Methodology
- Matured Software Design Methodology
- Joint design – Still in Infancy but popular!

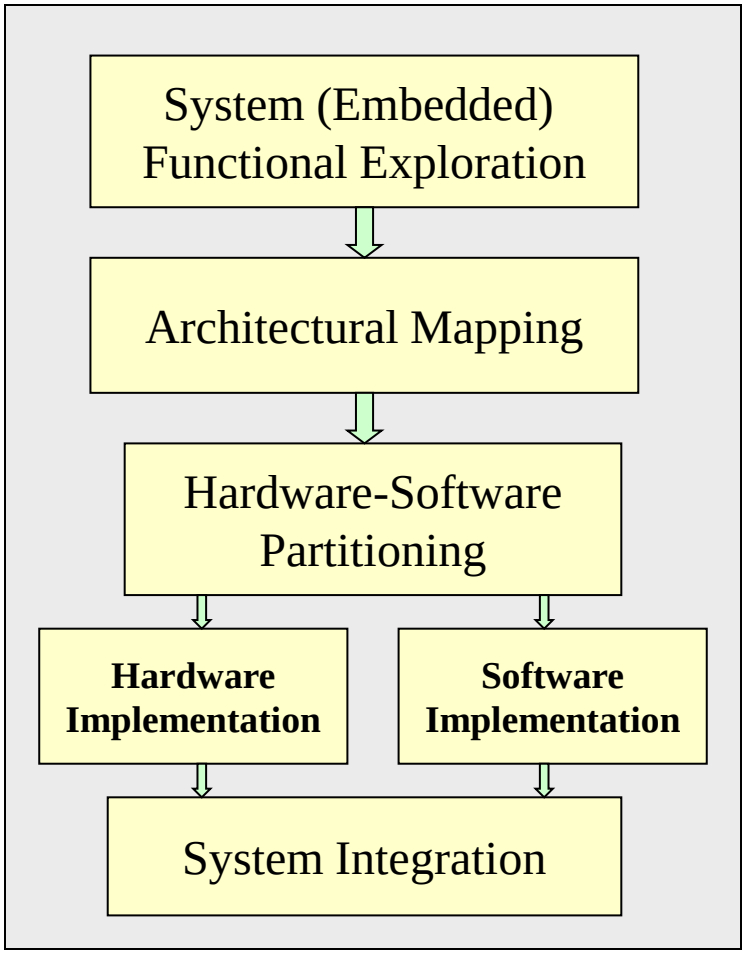
Hardware-Software Co-design

- An approach utilizing the maximum efficiency of Hardware and Software is needed
- Recent developments in CAD Tools
- Result -- Hardware Software Codesign
  - A unified approach
  - Large Design Space Exploration
  - Improved Time to Market

Codesign Methodology



Hardware-Software Codesign



- Functional exploration: Define a desired product's requirements and produce a specification of the system behavior.
- Map this specification onto various hardware and software architectures.
- Partition the functions between silicon and code; and map them directly to hardware or software components.
- Integrate system for prototype test.

HW/SW-Codesign

Co-design of (embedded) computer systems encompassing the following parts:

- Co-Specification: Describe system functionality at the abstract level

  System description is converted into a task graph representation

- HW-SW Partitioning: Take the task graph and decide which components are implemented where/how ?

  i.e. Dedicated hardware,

  Software -- one CPU or multiple CPUs

HW/SW-Codesign

- HW-SW Co-Synthesis: Analyze the task graph and decide on the system architecture.

  (incorporates HW/SW partitioning as heart of co-synthesis process)

- HW-SW Co-Simulation: Simulate embedded device's functionality before prototype construction. Simultaneous simulation of hardware and software.
- Co-Verification: Mathematical or simulation-based verification that device meets requirements.

HW/SW Co-Specification

- Model the (embedded) system functionality from an abstract level.
- Developing system specification that describes hardware, software modules and relationship (interface) between the hardware and software.
- No concept of hardware or software yet.
- Common environment

  SystemC: based on C++.
- Specification is analyzed to generate a task graph representation of the system functionality.

Hardware-Software Co-Synthesis

Four Principal Phases of Co-synthesis:

- **Partitioning**

  Dividing the functionality of an embedded system into units of computation.

- **Scheduling**

  Choosing time at which various computation units will occur.

- **Allocation**

  Determining the processing elements (PEs) on which computations will occur.

- **Mapping**

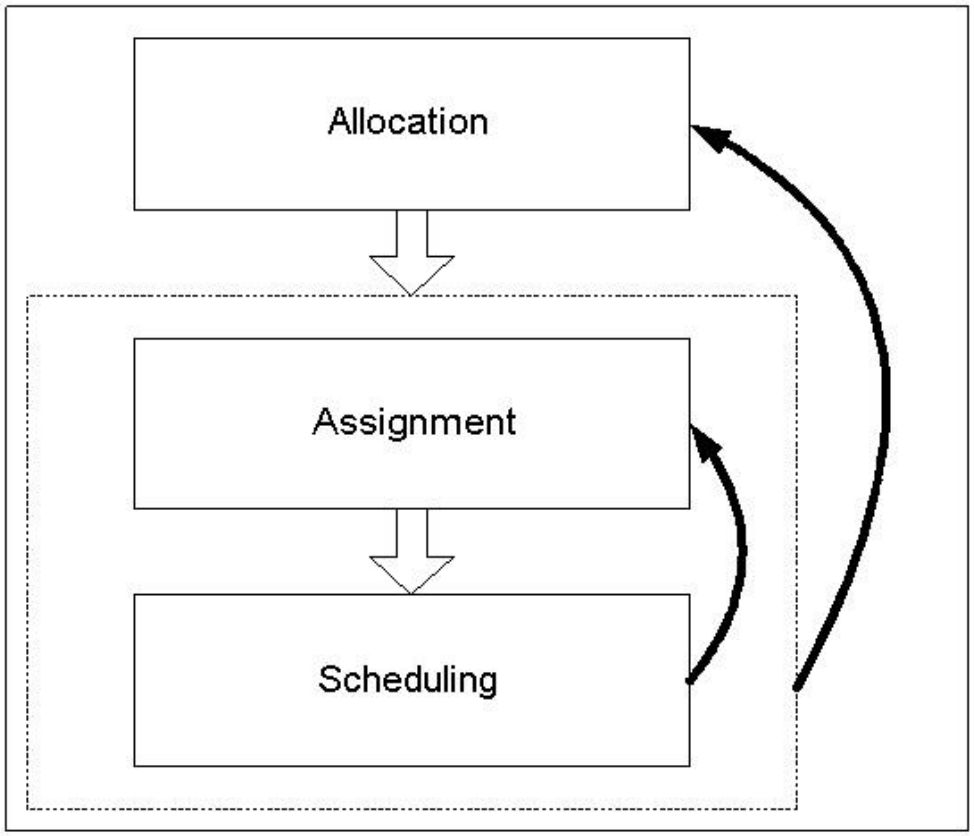
  Choosing particular component types for the allocated units (of computations).

HW/SW Co-Synthesis

Automatic and semi-automatic design of hardware and software modules to meet the specification.
Automatically derive the system architecture. Tightly coupled with HW/SW Partitioning along with:

- **Allocation:** select the number and type of communication links and processing elements for the target system.
- **Assignment (Mapping):** Mapping tasks to processing elements.
- **Scheduling:** Timing of task execution and communications.

Common Co-Synthesis Structure



System Partitioning

Introduces a design methodology that uses several techniques:

- **Partition the system specification into processes/tasks**

  The best way to partition a specification depends on the characteristics of the underlying hardware platform

- **Determine the performance of the function when executed on the hardware platform**

  We usually rely on approximating

- **Allocate processes onto various processing elements**

HW/SW Partitioning

- An area of significant research
- Analyzes task graph to determine each task's placement (HW or SW)
- Many algorithms have been developed.
- Major problem involves the computation time of partitioning algorithm

Hardware-Software Partitioning

Hardware/Software System Design involve: Modeling, Validation and Implementation

- **System implementation involves:**

  Hardware-Software Partitioning

  Finding those parts of the model best implemented in hardware & those best implemented in software.
- **Such partitions can be decided by the designer**

  with successive refinements

  or determined by the CAD tools

Hardware-Software Partitioning

For embedded systems, such partitioning represents a physical partition of the system functionality into:

- **Hardware**
- **Software executing on one or more CPUs**

Various formation of the Partitioning Problem that are based on:

- **Architectural Assumptions**
- **Partitioning Goals**
- **Solution Strategies**

COWARE: A design environment for application specific architectures targets telecom applications

Partitioning Techniques

Hardware-Software Homogeneous System Model => Task Graph

For each node of the task graph, determine implementation choices (HW or SW):

- Keep the scheduling of nodes at the same time
- Meet real-time constraints

- There is intimate relationship between partitioning and scheduling.
- Wide variation in timing properties of the hardware and software implementation of a task.

That effects the overall latency significantly