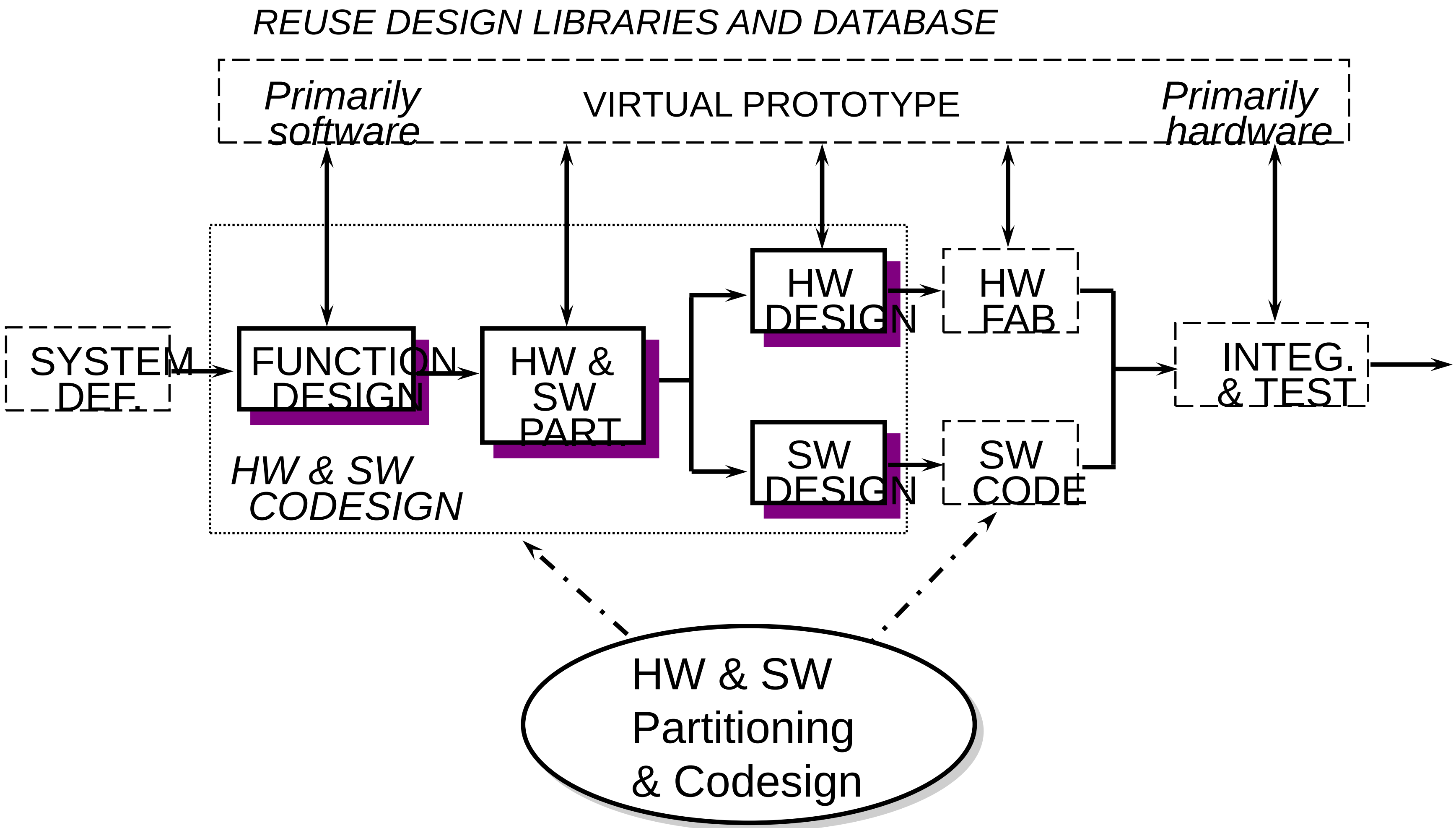
Hardware/Software Codesign

Embedded Systems – 5

Hemant Ghayvat
Department of Computer Science and Media Technology
hemant.ghayvat@lnu.se

Rapid Prototyping Design Process



Lecture’s Goals

Introduce the fundamentals of HW/SW codesign and partitioning concepts in designing embedded systems

- Discuss the **current trends in the codesign** of embedded systems
- Provide information on the goals of and methodology for **partitioning hardware/software** in systems

Show benefits of the **codesign approach** over current design process

- Provide information on how to incorporate these techniques into a general digital design methodology for embedded systems

Illustrate **how codesign concepts** are being **introduced into design methodologies**

- Several example codesign systems are discussed

Introduction to Embedded Systems and Hardware-Software Codesign

Introduction

Unified HW/SW Representations

HW/SW Partitioning Techniques

Integrated HW/SW Modeling Methodologies

HW and SW Synthesis Methodologies

Industry Approaches to HW/SW Codesign

Hardware/Software Codesign Research

Summary

Codesign Definition and Key Concepts

Codesign

- The meeting of system-level objectives by exploiting the trade-offs between hardware and software in a system through their **concurrent design**

Key concepts

- Concurrent: hardware and software developed at the same time on parallel paths
- Integrated: interaction between hardware and software developments to produce designs that meet performance criteria and functional specifications

Motivations for Codesign

Factors driving codesign (hardware/software systems):

- Instruction Set Processors (ISPs) available as cores in many design kits (386s, DSPs, microcontrollers,etc.)
- Systems on Silicon - many transistors available in typical processes (> 10 million transistors available in IBM ASIC process, etc.)
- Increasing capacity of field programmable devices - some devices even able to be reprogrammed on-the-fly (FPGAs, CPLDs, etc.)
- Efficient C compilers for embedded processors
- Hardware synthesis capabilities

Motivations for Codesign (cont.)

The **importance of codesign** in designing hardware/software systems:

- Improves design **quality, design cycle time, and cost**
  - Reduces integration and test time
- Supports growing complexity of embedded systems

Three Categories for Hardware/Software Systems

Application Domain
- Embedded systems
  - Manufacturing control
  - Consumer electronics
  - Vehicles
  - Telecommunications
  - Defense Systems
- Instruction Set Architectures
- Reconfigurable Systems

Degree of programmability

Implementation Features
- Discrete vs. integrated components
- Fabrication technologies

Categories of Codesign Problems

Codesign of embedded systems

- Usually consist of sensors, controller, and actuators
- Are reactive systems
- Usually have real-time constraints
- Usually have dependability constraints

Codesign of ISAs (Instruction set architecture)

- Application-specific instruction set processors (ASIPs)
- Compiler and hardware optimization and trade-offs

Codesign of Reconfigurable Systems

- Systems that can be personalized after manufacture for a specific application
- Reconfiguration can be accomplished before execution or concurrent with execution (called *evolvable* systems)

Components of the Codesign Problem

Specification of the system

Hardware/Software Partitioning

- Architectural assumptions - type of processor, interface style between hardware and software, etc.
- Partitioning objectives - maximize speedup, latency requirements, minimize size, cost, etc.
- Partitioning strategies - high level partitioning by hand, automated partitioning using various techniques, etc.

Scheduling

- Operation scheduling in hardware
- Instruction scheduling in compilers
- Process scheduling in operating systems

Modeling the hardware/software system during the design process

Embedded Systems: Complexity Issues

Complexity of embedded systems is continually increasing

Number of states in these systems (especially in the software) is very large

Description of a system can be complex, making system analysis extremely hard

Complexity management techniques are necessary to model and analyze these systems

Systems becoming too complex to achieve accurate “first pass” design using conventional techniques

New issues rapidly emerging from new implementation technologies

Techniques to Support Complexity Management

Delayed HW/SW partitioning

- Postpone as many decisions as possible that place constraints on the design

Abstractions and decomposition techniques

Incremental development

- “Growing” software
- Requiring top-down design

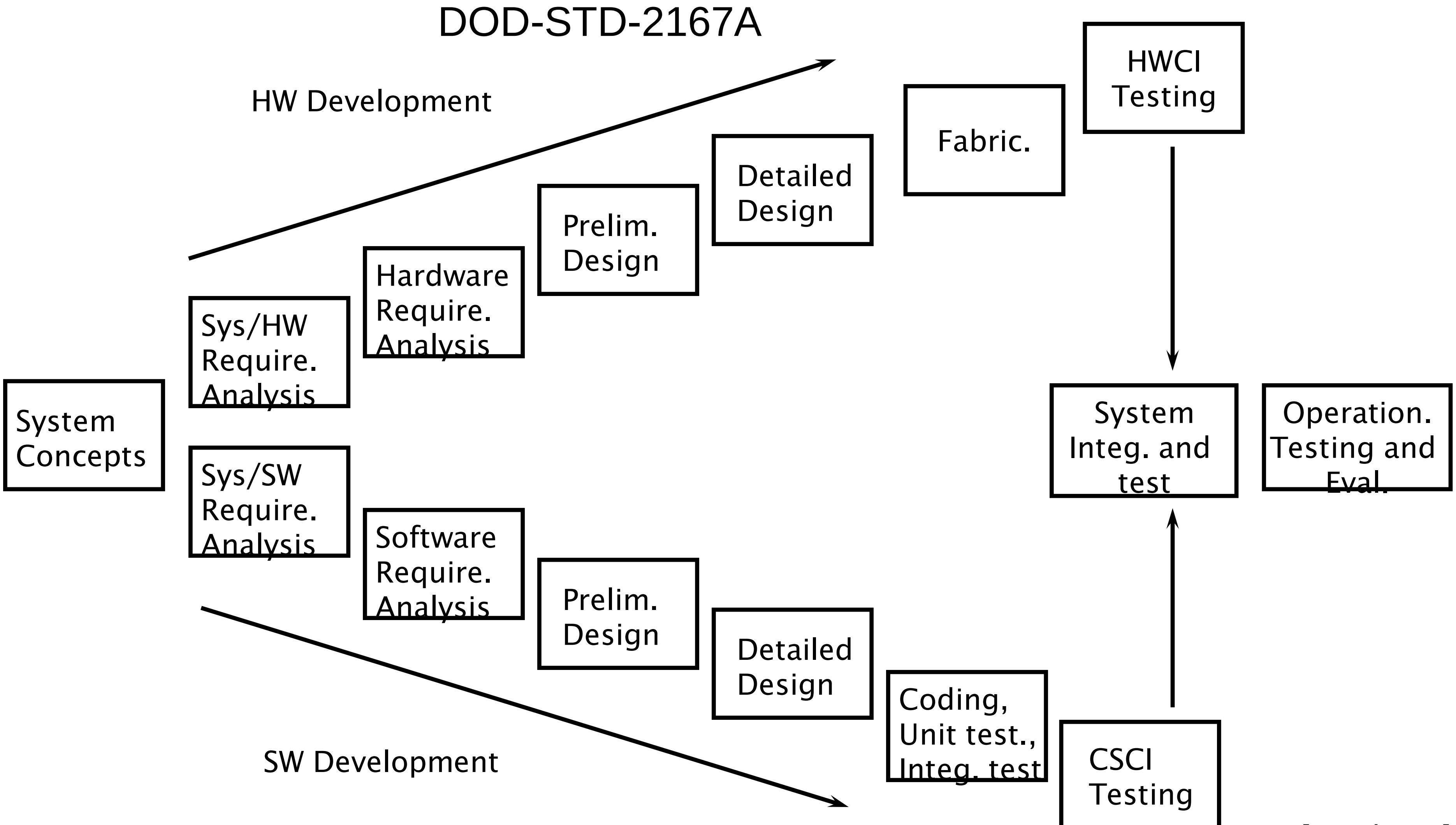
Description languages

Simulation

Standards

Design methodology management framework

A Model of the Current Hardware/Software Design Process



[Franke91]

Current Hardware/Software Design Process

Basic features of current process:

- System immediately partitioned into hardware and software components
- Hardware and software developed separately
- "Hardware first" approach often adopted

Implications of these features:

- HW/SW trade-offs restricted
  - Impact of HW and SW on each other cannot be assessed easily
- Late system integration

Consequences these features:

- Poor quality designs
- Costly modifications
- Schedule slippages

Common Misconceptions in Current Hardware/Software Design Process

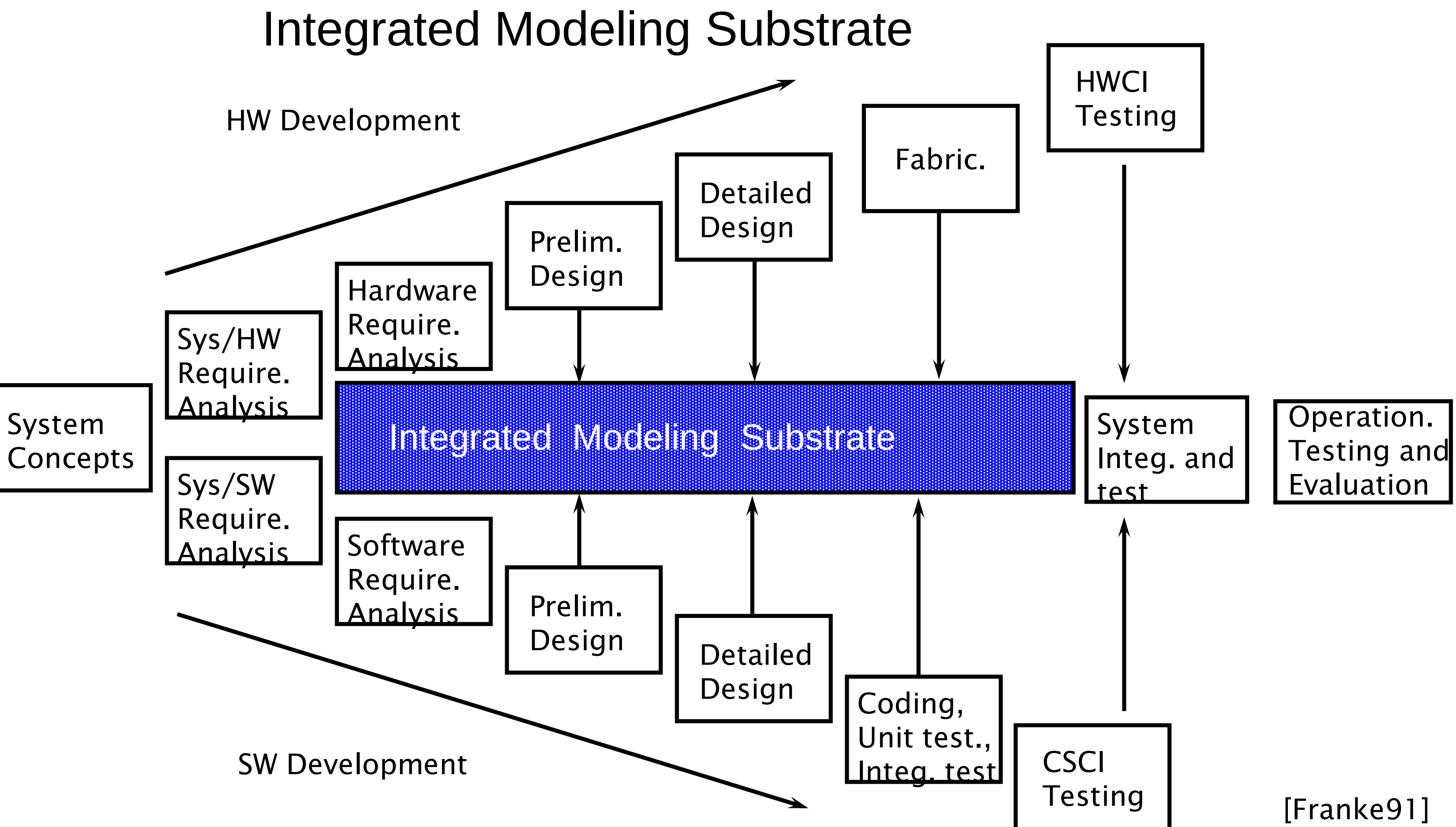
Hardware and software can be acquired separately and independently, with successful and easy integration of the two later

Hardware problems can be fixed with simple software modifications

Once operational, software rarely needs modification or maintenance

Valid and complete software requirements are easy to state and implement in code

Directions of the HW/SW Design Process



Requirements for the Ideal Codesign Environment

Unified, unbiased hardware/software representation

- Supports uniform design and analysis techniques for hardware and software
- Permits system evaluation in an integrated design environment
- Allows easy migration of system tasks to either hardware or software

Iterative partitioning techniques

- Allow several different designs (HW/SW partitions) to be evaluated
- Aid in determining best implementation for a system
- Partitioning applied to modules to best meet design criteria (functionality and performance goals)

Requirements for the Ideal Codesign Environment (cont.)

Integrated modeling substrate

- Supports evaluation at several stages of the design process
- Supports step-wise development and integration of hardware and software

Validation Methodology

- Ensures that system implemented meets initial system requirements

Cross-fertilization Between Hardware and Software Design

Fast growth in both VLSI design and software engineering has raised awareness of similarities between the two

- Hardware synthesis
- Programmable logic
- Description languages

Explicit attempts have been made to “transfer technology” between the domains

Cross-fertilization Between Hardware and Software Design (cont.)

VLSI DESIGN → SOFTWARE ENGINEERING

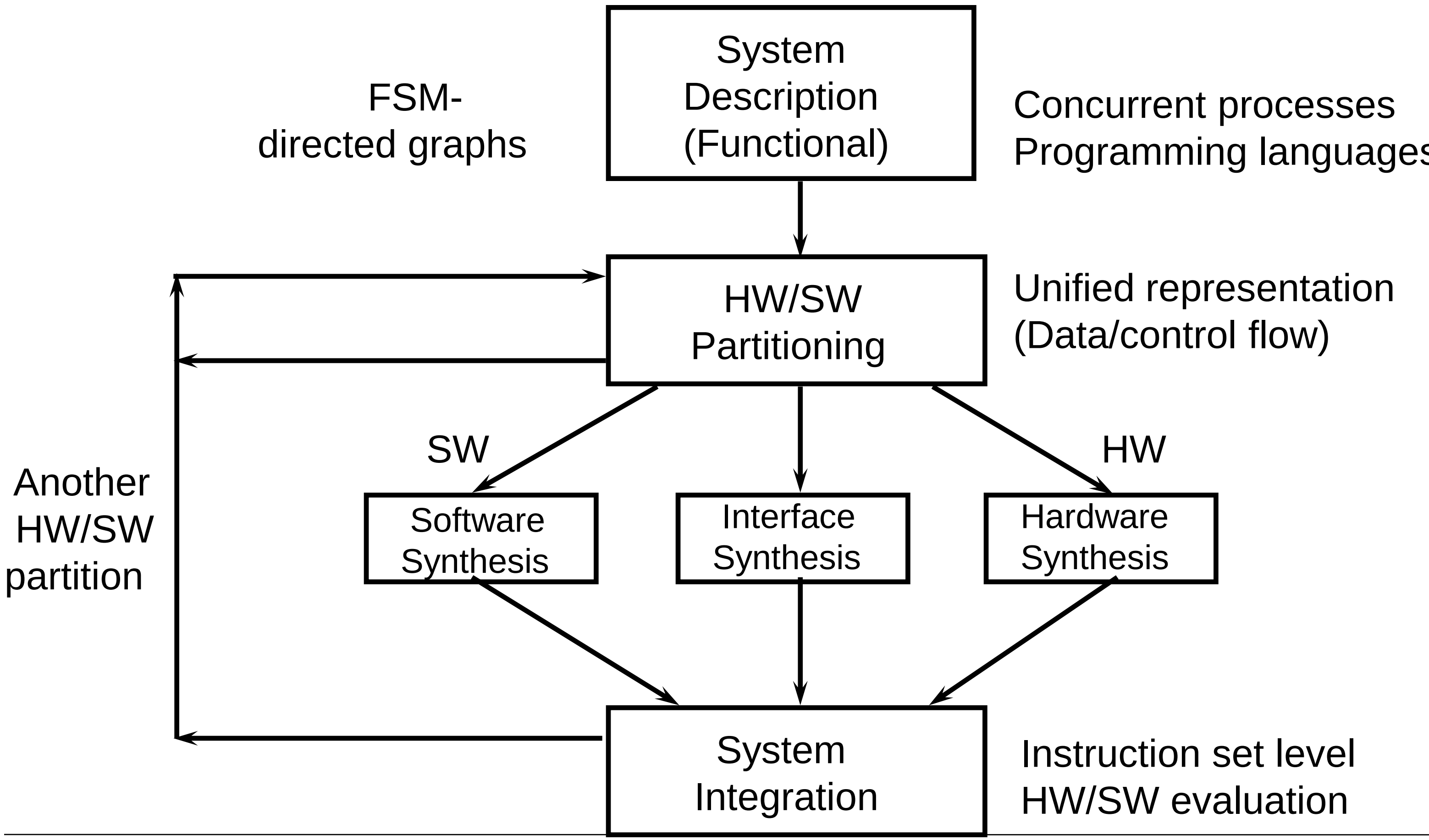
EDA (electronic design automation) *tool technology has been transferred to SW CAD systems*

Cross-fertilization Between Hardware and Software Design (cont.)

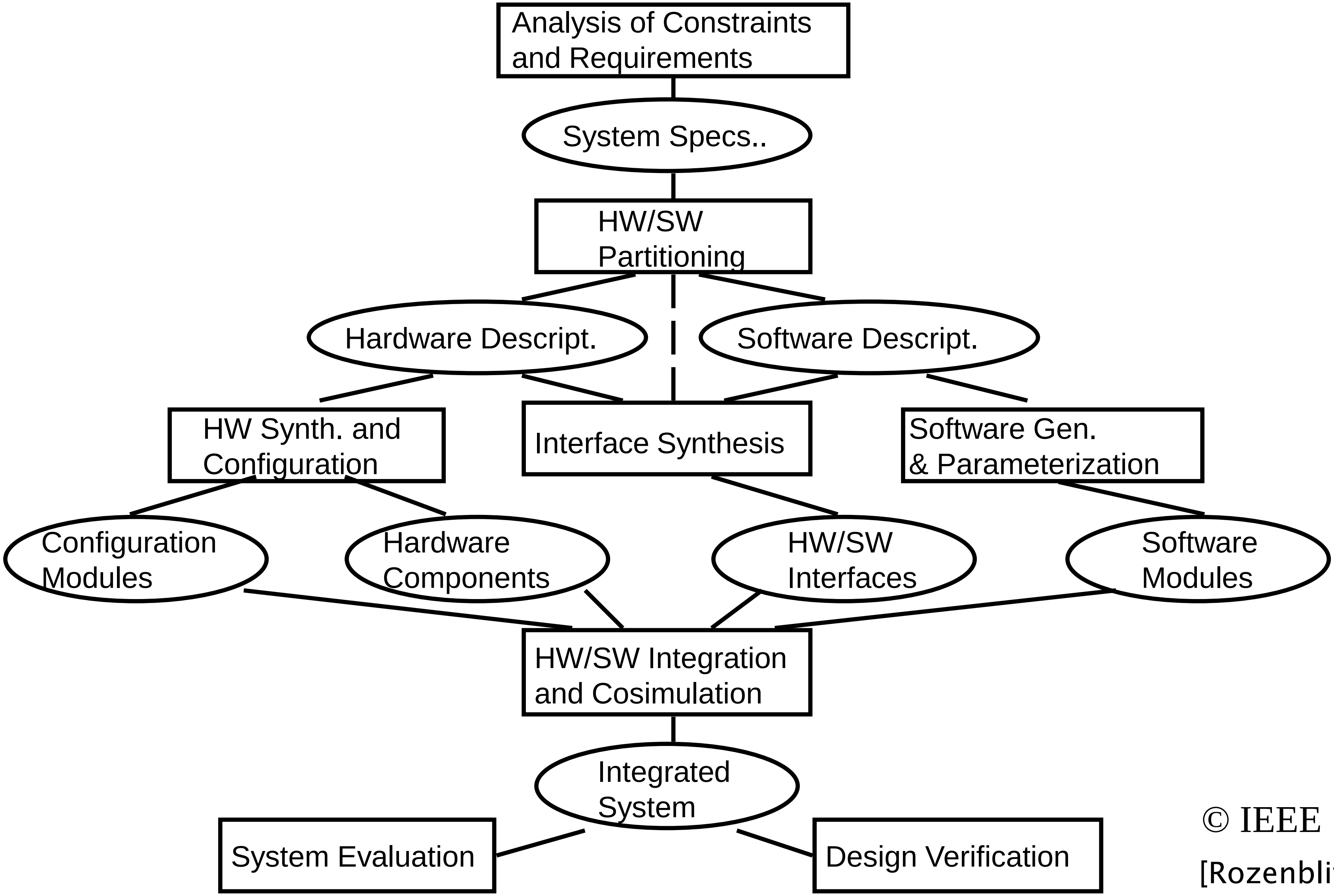
SOFTWARE ENGINEERING → VLSI DESIGN

Software technology has been transferred to EDA tools

Typical Codesign Process



Conventional Codesign Methodology



[Rozenblit94]

Codesign Features

Basic features of a codesign process:

Enables mutual influence of both HW and SW early in the design cycle

- Provides continual verification throughout the design cycle
- Separate HW/SW development paths can lead to costly modifications and schedule slippages

Enables evaluation of larger design space through tool interoperability and automation of codesign at abstract design levels

Advances in key enabling technologies (e.g., logic synthesis and formal methods) make it easier to explore design tradeoffs

State of Codesign Technology

Current use limited by:

- Lack of a standardized representation
- Lack of good validation and evaluation methods

Possible solutions:

- Extend existing hardware/software languages to the use of heterogeneous paradigms
- Extend formal verification techniques to the HW/SW domain

Issues and Problems: Integration

Errors in hardware and software design become much more costly as more commitments are made

"Hardware first" approach often compounds software cost because software must compensate for hardware inadequacies

References

[Base slide]Pao-Ann Hsiung, 熊博安 Graduate Institute of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan, R.O.C.

[Boehm73] Boehm, B.W. "Software and its Impact: A Quantitative Assessment," *Datamation*, May 1973, p. 48-59.

[Buchenrieder93] Buchenrieder, K., "Codesign and Concurrent Engineering", Hot Topics, *IEEE Computer*, R. D. Williams, ed., January, 1993, pp. 85-86

[Buck94] Buck, J., et al., "Ptolemy: a Framework for Simulating and Prototyping Heterogeneous Systems," *International Journal of Computer Simulation*, Vol. 4, April 1994, pp. 155-182.

[Chiodo92] Chiod0, M., A. Sangiovanni-Vincentelli, "Design Methods for Reactive Real-time Systems Codesign," *International Workshop on Hardware/Software Codesign*, Estes Park, Colorado, September 1992.

[Chiodo94] Chiodo, M., P. Giusto, A. Jurecska, M. Marelli, H. C. Hsieh, A. Sangiovanni-Vincentelli, L. Lavagno, "Hardware-Software Codesign of Embedded Systems," *IEEE Micro*, August, 1994, pp. 26-36; © IEEE 1994.

[Chou95] P. Chou, R. Ortega, G. Borriello, "The Chinook hardware/software Co-design System," *Proceedings ISSS*, Cannes, France, 1995, pp. 22-27.

[DeMicheli93] De Micheli, G., "*Extending CAD Tools and Techniques*", Hot Topics, *IEEE Computer*, R. D. Williams, ed., January, 1993, pp. 84

[DeMicheli94] De Micheli, G., "Computer-Aided Hardware-Software Codesign", *IEEE Micro*, August, 1994, pp. 10-16

[DeMichelli97] De Micheli, G., R. K. Gupta, "Hardware/Software Co-Design," *Proceedings of the IEEE*, Vol. 85, No. 3, March 1997, pp. 349-365.

[Ernst93] Ernst, R., J. Henkel, T. Benner, "*Hardware-Software Cosynthesis for Micro-controllers*", *IEEE Design and Test*, December, 1993, pp. 64-75

[Franke91] Franke, D.W., M.K. Purvis. "Hardware/Software Codesign: A Perspective," *Proceedings of the 13th International Conference on Software Engineering*, May 13-16, 1991, p. 344-352; © IEEE 1991

References (Cont.)

Copyright ã 1995-1999 SCRA, Hardware/Software Codesign Overview RASSP Education & Facilitation Program Module 14

Proceedings of the International Workshops / Symposium on HW/SW Codesign, 1993 ~ 2001 (ACM Press & IEEE CS Press).

Hardware Software Co-design of Embedded Systems, F. Balarin, Chiodo, et al., Kluwer Academic Publishers, May 1997.

Co-synthesis of Hardware and Software for Embedded Systems, R. Gupta, Kluwer Academic Publishers, 1995.

Special Issue of the Proceedings of the IEEE on Hardware Software Co-design edited by G. De Micheli, Vol. 85, No. 3, March 1997.

POLIS and Ptolemy tools introduction materials and manuals

[Gajski94] Gajski, D. D., F. Vahid, S. Narayan, J. Gong, *Specification and Design of Embedded Systems*, Prentice Hall, Englewood Cliffs, N J, 07632, 1994

[Gupta92] Gupta, R.K., C.N. Coelho, Jr., G.D. Micheli. "Synthesis and Simulation of Digital Systems Containing Interactive Hardware and Software Components," *29th Design Automation Conference*, June 1992, p.225-230.

[Gupta93] Gupta, R.K., G. DeMicheli, "Hardware-Software Cosynthesis for Digital Systems," *IEEE Design and Test*, September 1993, p.29-40; © IEEE 1993.

[Hermann94] Hermann, D., J. Henkel, R. Ernst, "An approach to the estimation of adapted Cost Parameters in the COSYMA System*", 3rd International Conference on Hardware/Software codesign*, Grenoble, France, September 22-24, 1994, pp. 100-107

[Hood94] Hood, W., C. Myers, "RASSP: Viewpoint from a Prime Developer," *Proceedings 1st Annual RASSP Conference*, Aug. 1994.

[IEEE] All referenced IEEE material is used with permission.

[Ismail95] T. Ismail, A. Jerraya, "Synthesis Steps and Design Models for Codesign," *IEEE Computer*, no. 2, pp. 44-52, Feb 1995.

[Kalavade93] A. Kalavade, E. Lee, "A Hardware-Software Co-design Methodology for DSP Applications," *IEEE Design and Test*, vol. 10, no. 3, pp. 16-28, Sept. 1993.

References (Cont.)

[Parker84] Parker, A.C., “Automated Synthesis of Digital Systems,” *IEEE Design and Test,*, November 1984, p. 75-81.

[RASSP94] *Proceedings of the 1st RASSP Conference*, Aug. 15-18, 1994.

[Rozenblit94] Rozenblit, J. and K. Buchenrieder (editors). <u>Codesign Computer -Aided Software/Hardware Engineering</u>, IEEE Press, Piscataway, NJ, 1994; © IEEE 1994.

[Smith86] Smith, C.U., R.R. Gross. “Technology Transfer between VLSI Design and Software Engineering: CAD Tools and Design Methodologies,” *Proceedings of the IEEE*, Vol. 74, No. 6, June 1986, p.875-885.

[Srivastava91] M. B. Srivastava, R. W. Broderson, “Rapid prototyping of Hardware and Software in a Unified Framework,” *Proceedings ICCAD*, 1991, pp. 152-155.

[Subrahmanyam93] Subrahmanyam, P. A., “Hardware-Software Codesign -- Cautious optimism for the future”, Hot Topics, *IEEE Computer*, R. D. Williams, ed., January, 1993, pp. 84

[Tanenbaum87] Tanenbaum, A.S., *Operating Systems: Design and Implementation*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1987.

[Terry90] Terry, C. “Concurrent Hardware and Software Design Benefits Embedded Systems,” *EDN*, July 1990, p. 148-154.

[Thimbleby88] Thimbleby, H. “Delaying Commitment,” *IEEE Software*, Vol. 5, No. 3, May 1988, p. 78-86.

[Thomas93] Thomas, D.E., J.K. Adams, H. Schmitt, “A Model and Methodology for Hardware-Software Codesign,” *IEEE Design and Test*, September 1993, p.6-15; © IEEE 1993.

[Turn78] Turn, R., “Hardware-Software Tradeoffs in Reliable Software Development,” *11th Annual Asilomar Conference on Circuits, Systems, and Computers,* 1978, p.282-288.

[Vahid94] Vahid, F., J. Gong, D. D. Gajski, “*A Binary Constraint Search Algorithm for Minimizing Hardware During Hardware/Software Partitioning”,* 3rd International Conference on Hardware/Software Codesign, Grenoble, France, Sepetember22-24, 1994, pp. 214-219

[Wolf94] Wolf, W.H. “Hardware-Software Codesign of Embedded Systems,” *Proceedings of the IEEE*, Vol. 82, No.7, July 1994, p.965-989.

References (Cont.)

Additional Reading:

Aylor, J.H. et al., "The Integration of Performance and Functional Modeling in VHDL" in *Performance and Fault Modeling with VHDL*, J. Schoen, ed., Prentice-Hall, Englewood Cliffs, N.J., 1992.

D'Ambrosio, J. G., X. Hu, "Configuration-level Hardware-Software Partitioning for Real-time Embedded Systems", *3rd International Conference on Hardware/Software codesign*, Grenoble, France, September 22-24, 1994, pp. 34-41

Eles, P., Z. Peng, A. Doboli, "VHDL System-Level Specification and Partitioning in a Hardware-Software Cosynthesis Environment", *3rd International Conference on Hardware/Software codesign*, Grenoble, France, September 22-24, 1994, pp. 49-55

Gupta, R.K., G. DeMicheli, "Hardware-Software Cosynthesis for Digital Systems," *IEEE Design and Test*, September 1993, p.29-40.

Richards, M., Gadient, A., Frank, G., eds. *Rapid Prototyping of Application Specific Signal Processors*, Kluwer Academic Publishers, Norwell, MA, 1997

Schultz, S.E., "An Overview of System Design," *ASIC and EDA*, January 1993, p.12-21.

Thomas, D. E, J. K. Adams, H. Schmit, "A Model and Methodology for Hardware-Software Codesign", *IEEE Design and Test*, September, 1993, pp. 6-15

Zurcher, F.W., B. Randell, "Iterative Multi-level Modeling - A Methodology for Computer System Design," *Proceedings IFIP Congress '68,* Edinburgh, Scotland, August 1968, p.867-871.

Lnu.se