EECS 298: Embedded Software Synthesis Lecture 8

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

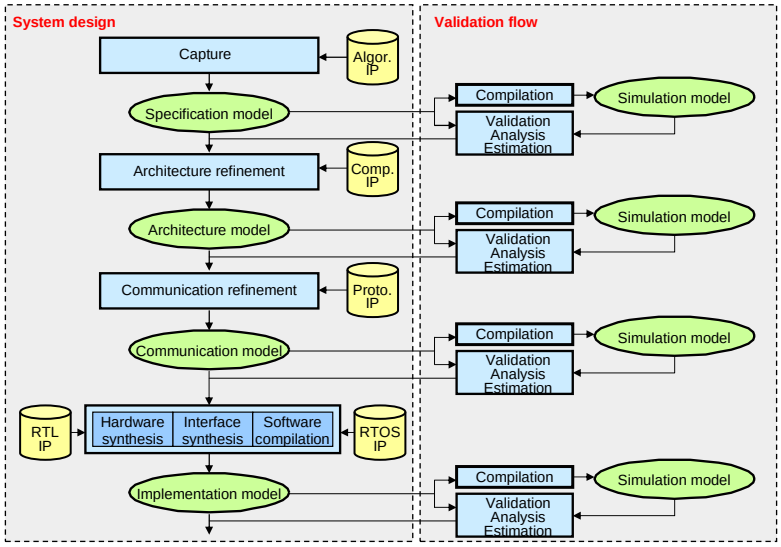
Lecture 8: Overview

- Project Status and Progress
- Software Synthesis
  - Tasks
  - Benefits
  - Embedded Talk
    - Haobo Yu, Ph.D. Candidate, UCI
    - *"Software Synthesis for System on Chip"*
- Presentation
  - Code Generation (Eun Kyong Seo)

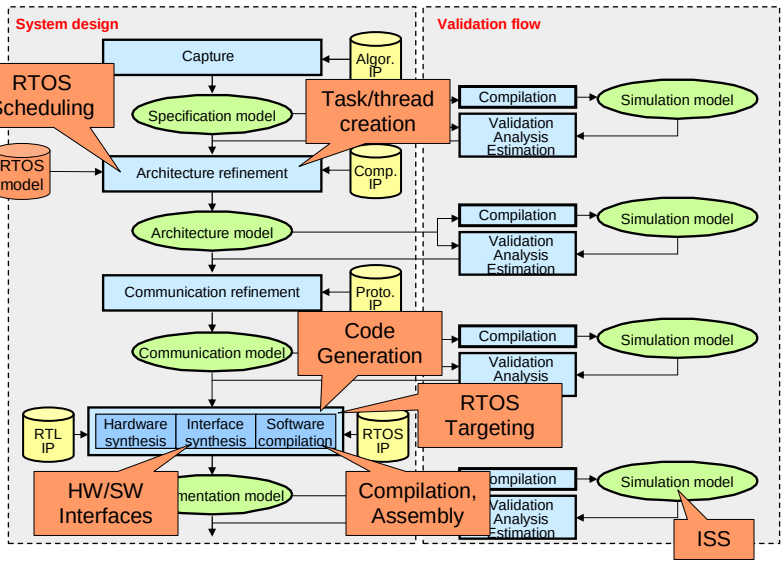
Project Status and Progress

			Schedule
•	Option 1: Hands-on Experience with Embedded Software		
	– CJC:	Mobile IP (embedded Linux) on wireless access point	TBD
	– CBH:	Port PalmOS application to WindowsCE	TBD
	– SYC+CWS:	Traffic light controller on Xilinx board	TBD
	– QKN + RL:	Temperature sensor on flash microcontroller	TBD
	– ML + HL:	Instant messenger application on mobile phone	TBD
	– KLN:	Snake game (Java) on mobile phone	TBD
	– SI:	Real-time UML/Java appl. wallet PDA, cash register PC	TBD
•	Option 2: Literature Research		
	– HEC:	RTOS survey	Week 6
	– KDS:	Target processor survey	Week 7
	– GK:	Power management for embedded applications	Week 7
	– EKS:	Code generation for embedded processors	Week 8
•	Option 3: Embedded Software Synthesis using SpecC		
	– JHB:	Reed-Solomon decoder	TBD
	– AG:	Digital camera	Week 9
	– TWH:	Tic-tac-toe game	Week 9
	– GS:	Wireless sensor node measuring motion	TBD
	– ISG:	Elevator controller	TBD
	– HCL:	Algorithm evaluation for fair packet scheduling	TBD

Software Synthesis



Software Synthesis



Software Synthesis

- RTOS Scheduling and Task Creation
  - For each SW PE, serialize the execution of behaviors
  - Option (a): Static scheduling
    - For each set of concurrent behaviors, determine fixed order of execution
  - Option (b): Dynamic scheduling by RTOS
    - Choose scheduling policy, i.e. Round-robin or priority-based
    - For each set of concurrent behaviors, determine scheduling priority
  - Result: System model with abstract RTOS scheduler

Software Synthesis

- Software Implementation
  - Code generation
    - Conversion of SLDL (i.e. SpecC) to C
  - RTOS targeting
    - Code targeting for selected processor
    - Code targeting for selected RTOS
  - Compilation, assembly, linking
    - Generation of instruction set code
  - Validation
    - Instruction Set Simulation (ISS)
  - Result:
    Clock-cycle accurate model of each SW PE
    - Output: downloadable object code

EECS298: Embedded Software Synthesis, Lecture 8 (c) 2004 R. Doemer 7

Software Synthesis

- Why? What are the benefits?
  - Abstract specification
    - free of implementation details
  - Boundary between SW and HW is open
  - SW and HW are being developed concurrently
    - No need for HW platform before SW development
  - HW/SW interfaces are automatically generated
    - No need to know low-level HW details
  - Portable
    - platform independent specification
  - Retargetable
    - Independent from RTOS
    - Allows RTOS exploration

EECS298: Embedded Software Synthesis, Lecture 8 (c) 2004 R. Doemer 8

Software Synthesis

- Embedded Talk:
  "Software Synthesis for System on Chip"
  Haobo Yu, Ph.D. Candidate, UCI