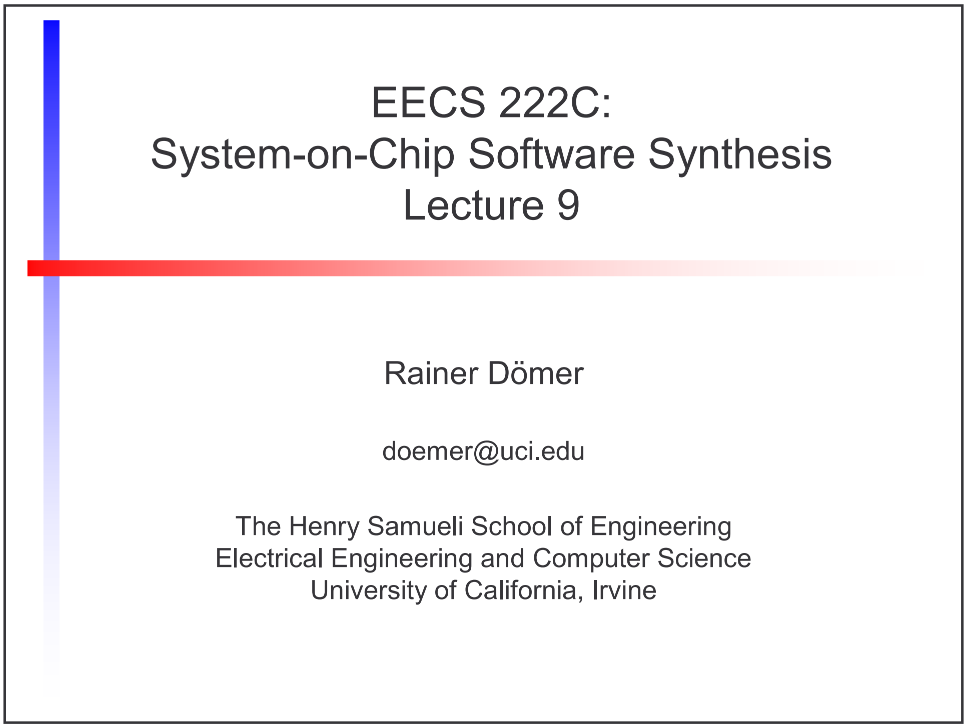
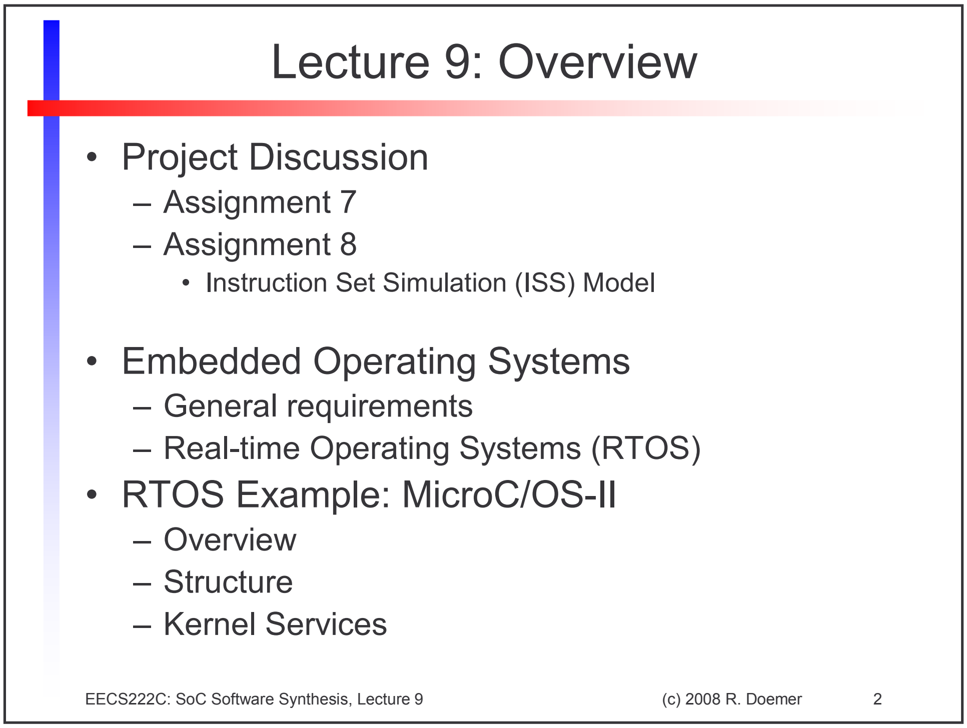


EECS 222C:
System-on-Chip Software Synthesis
Lecture 9

Rainer Dömer

doemer@uci.edu

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



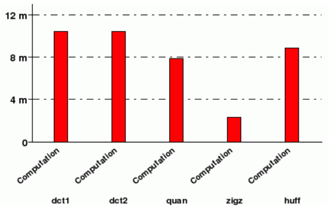
Lecture 9: Overview

- Project Discussion
  - Assignment 7
  - Assignment 8
    - Instruction Set Simulation (ISS) Model
- Embedded Operating Systems
  - General requirements
  - Real-time Operating Systems (RTOS)
- RTOS Example: MicroC/OS-II
  - Overview
  - Structure
  - Kernel Services

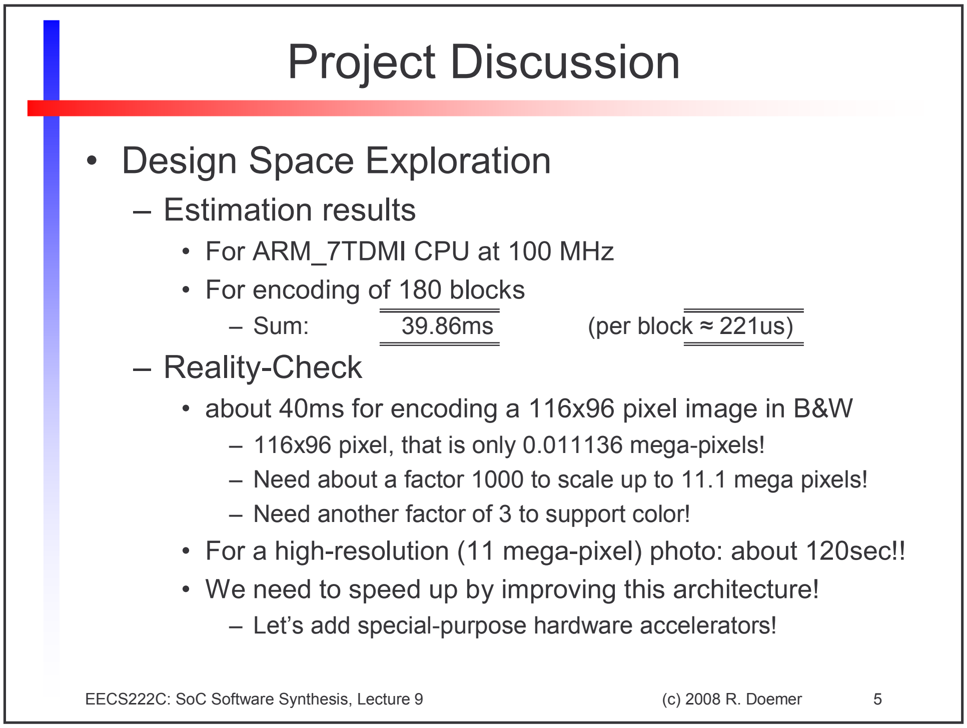
Assignment 7

- Software Synthesis and Instruction Set Simulation
  1. Follow the demo given in Lecture 8
     - Detailed instructions are provided in
     - `/home/doemer/EECS222C_F08/lecture8/Lecture8.txt`
     - on our server
     - `epsilon.eecs.uci.edu`
  2. Note the major steps in the software synthesis process
  3. Discuss any issues and/or questions on the noteboard
- Deliverable
  - none (but be prepared to perform similar software synthesis and instruction set simulation for our JPEG Encoder example in the next assignment… ;-)
- Due
  - by Friday, Nov 21, 2008, at noon

Project Discussion



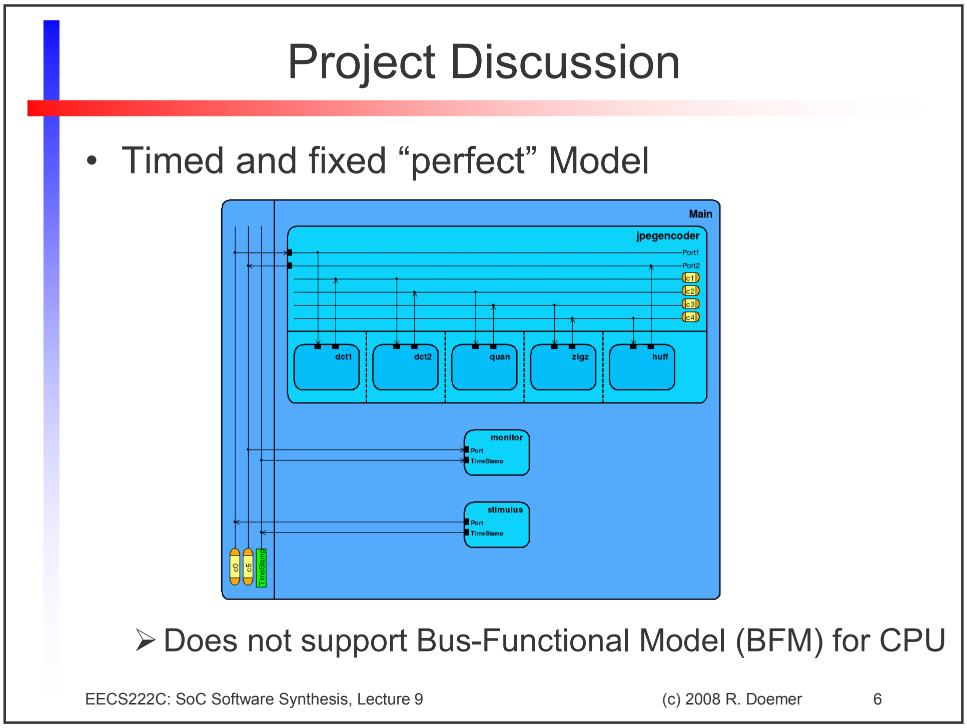
- Design Space Exploration
  - Estimation results
    - For ARM_7TDMI CPU at 100 MHz
    - For encoding of 180 blocks
      - ChenDCT1: 10.41ms (10411200ns / 180 ≈ 58us)
      - ChenDCT2: 10.41ms (10411200ns / 180 ≈ 58us)
      - Quantize: 7.84ms (7839030ns / 180 ≈ 44us)
      - Zigzag: 2.32ms (2316600ns / 180 ≈ 13us)
      - Huffman: 8.88ms (8882810ns / 180 ≈ 49us)
        - » Sum: 39.86ms (per block ≈ 221us)
  - Reality-Check!
    - 40ms for encoding the test JPEG image…
    - …is that fast or is it slow???



Project Discussion

- Design Space Exploration
  - Estimation results
    - For ARM_7TDMI CPU at 100 MHz
    - For encoding of 180 blocks
      - Sum: 39.86ms (per block ≈ 221us)
  - Reality-Check
    - about 40ms for encoding a 116x96 pixel image in B&W
      - 116x96 pixel, that is only 0.011136 mega-pixels!
      - Need about a factor 1000 to scale up to 11.1 mega pixels!
      - Need another factor of 3 to support color!
    - For a high-resolution (11 mega-pixel) photo: about 120sec!!
    - We need to speed up by improving this architecture!
      - Let's add special-purpose hardware accelerators!

EECS222C: SoC Software Synthesis, Lecture 9 (c) 2008 R. Doemer 5



Project Discussion

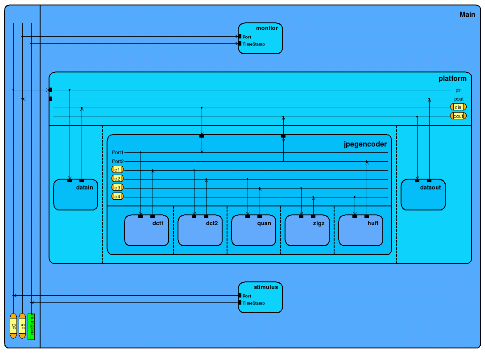
- Timed and fixed “perfect” Model

➢ Does not support Bus-Functional Model (BFM) for CPU

EECS222C: SoC Software Synthesis, Lecture 9 (c) 2008 R. Doemer 6

Project Discussion

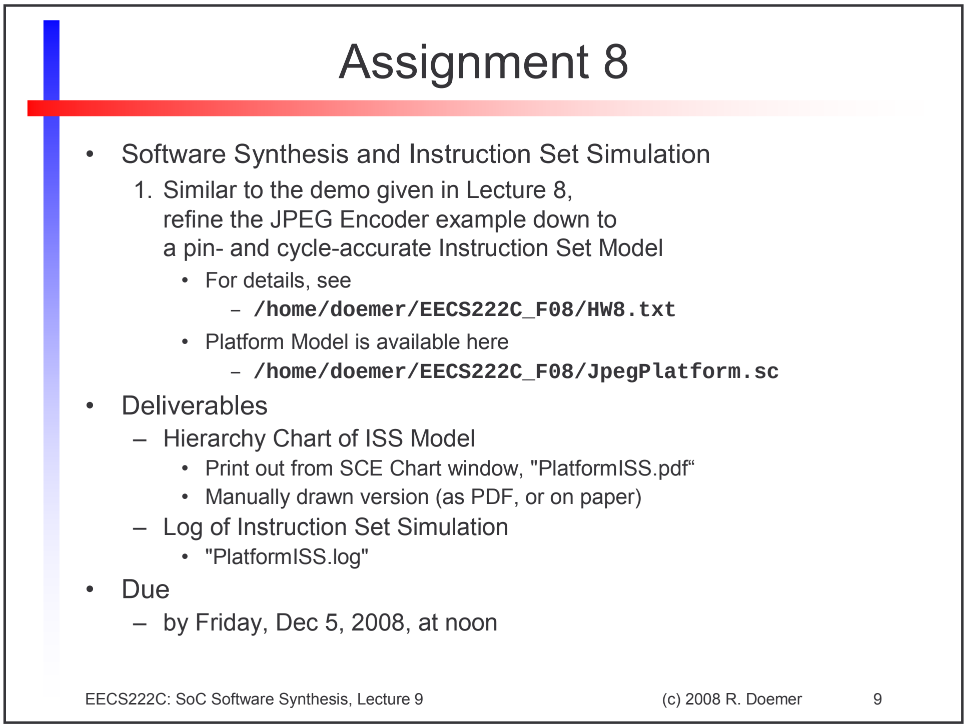
- Platform Model



  - Communication in **`jpegencoder`** can be refined to actual CPU bus
    - I/O units **`datain`** and **`dataout`** convert between
      - Abstract test bench communication (typed double-handshake)
      - BFM communication via CPU bus

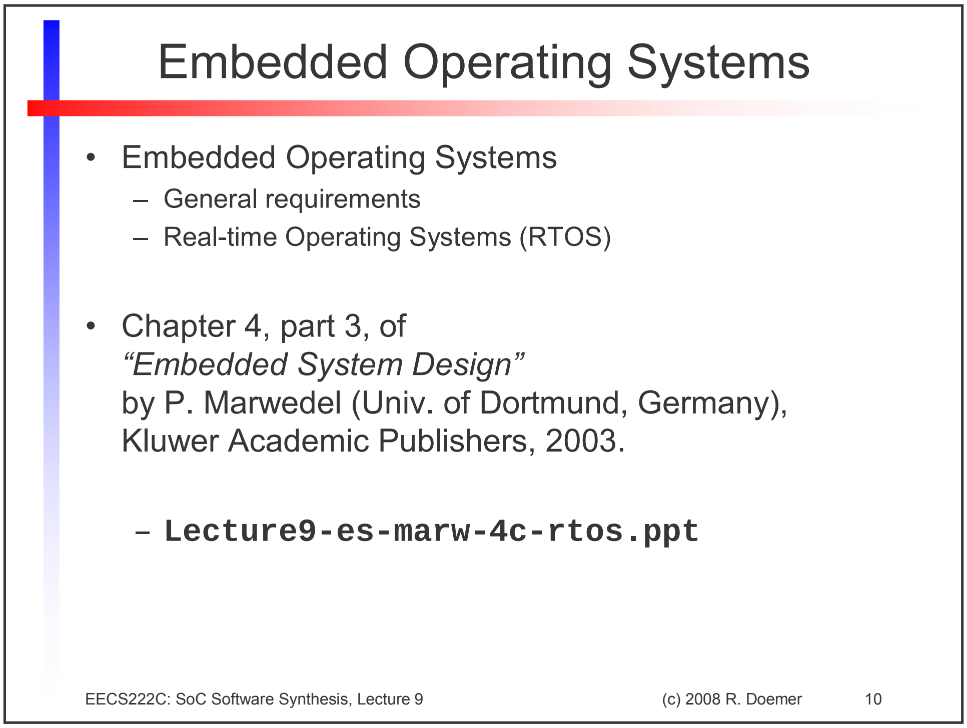
Project Discussion

- Current SCE Limitations
  - Instruction Set Simulator
    - Only available for ARM_7TDMI
    - Max. 1 system-wide instance
  - RTOS
    - Only available port for ARM_7TDMI is micro-OS II
    - Requires priority-based scheduling
      - with different priorities for each task
  - Code generator
    - CPU-internal channels limited to
      - Type-less **`c_handshake`**
      - Type-less **`c_double_handshake`**
    - CPU-external channels
      - Type-less **`c_handshake`**
      - Type-less and typed **`c_double_handshake`**
      - Type-less and typed **`c_queue`**



Assignment 8

- Software Synthesis and Instruction Set Simulation
  1. Similar to the demo given in Lecture 8,
     refine the JPEG Encoder example down to
     a pin- and cycle-accurate Instruction Set Model
     - For details, see
       - `/home/doemer/EECS222C_F08/HW8.txt`
     - Platform Model is available here
       - `/home/doemer/EECS222C_F08/JpegPlatform.sc`
- Deliverables
  - Hierarchy Chart of ISS Model
    - Print out from SCE Chart window, "PlatformISS.pdf“
    - Manually drawn version (as PDF, or on paper)
  - Log of Instruction Set Simulation
    - "PlatformISS.log"
- Due
  - by Friday, Dec 5, 2008, at noon



Embedded Operating Systems

- Embedded Operating Systems
  - General requirements
  - Real-time Operating Systems (RTOS)

- Chapter 4, part 3, of
  “Embedded System Design”
  by P. Marwedel (Univ. of Dortmund, Germany),
  Kluwer Academic Publishers, 2003.

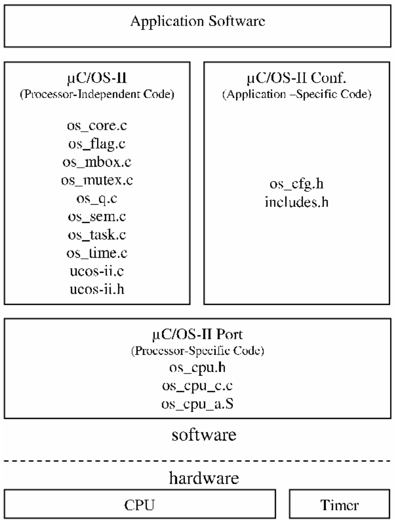
  - **`Lecture9-es-marw-4c-rtos.ppt`**

Embedded Operating Systems

- Example: MicroC/OS-II
  - Overview
    - multi-tasking real-time kernel
    - real-time support (most kernel functions deterministic)
    - task management
    - priority scheduling
    - preemption
    - ROM'able (executable from firmware)
      - memory footprint about 20 KB
    - portable (to over 40 different processor architectures, 8-64bit)
      - about 5500 lines of ANSI-C source code
      - small amount of processor-specific assembly code

Embedded Operating Systems

- Example: MicroC/OS-II
  - Structure



Application Software

µC/OS-II
(Processor-Independent Code)
os_core.c
os_flag.c
os_mbox.c
os_mutex.c
os_q.c
os_sem.c
os_task.c
os_time.c
ucos-ii.c
ucos-ii.h

µC/OS-II Conf.
(Application –Specific Code)
os_cfg.h
includes.h

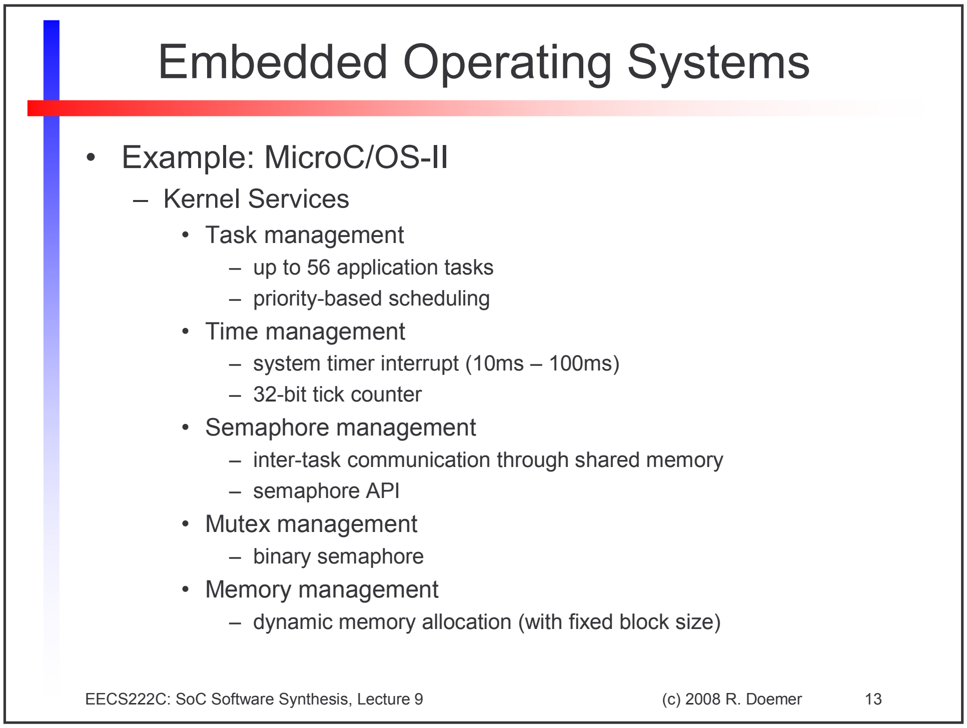
µC/OS-II Port
(Processor-Specific Code)
os_cpu.h
os_cpu_c.c
os_cpu_a.S

software

hardware

CPU

Timer



Embedded Operating Systems

- Example: MicroC/OS-II
  - Kernel Services
    - Task management
      - up to 56 application tasks
      - priority-based scheduling
    - Time management
      - system timer interrupt (10ms – 100ms)
      - 32-bit tick counter
    - Semaphore management
      - inter-task communication through shared memory
      - semaphore API
    - Mutex management
      - binary semaphore
    - Memory management
      - dynamic memory allocation (with fixed block size)

EECS222C: SoC Software Synthesis, Lecture 9 (c) 2008 R. Doemer 13