

# EECS 221: System-on-Chip 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

- Embedded Operating Systems
  - General requirements
  - Real-time Operating Systems (RTOS)
  - Middleware
- Example: MicroC/OS-II
  - Overview
  - Structure
  - Kernel Services

## Embedded Operating Systems

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

– `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

The diagram illustrates the structure of MicroC/OS-II, organized into several layers and components:

- Application Software**: The top layer, containing user applications.
- Processor-Independent Code**: A layer containing core OS code files:
  - 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
- Processor-Specific Code**: A layer containing application-specific code files:
  - os\_cfg.h
  - includes.h
- Port**: A layer containing processor-specific code files:
  - os\_cpu.h
  - os\_cpu\_c.c
  - os\_cpu\_a.S
- software**: A layer representing the compiled software.
- hardware**: A layer representing the hardware components, separated from software by a dashed line.
- CPU** and **Timer**: Specific hardware components at the bottom.

EECS221: SoC Software Synthesis, Lecture 8 (c) 2006 R. Doemer 5

## 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)

EECS221: SoC Software Synthesis, Lecture 8 (c) 2006 R. Doemer 6