

Spring 2011

CEG 751-01: Microprocessors II

Jack Jean

Wright State University - Main Campus, jack.jean@wright.edu

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Repository Citation

Jean, J. (2011). CEG 751-01: Microprocessors II. .
https://corescholar.libraries.wright.edu/cecs_syllabi/1030

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.

CEG 751: Microprocessors II
(Spring 2011, 6:05 – 7:20 Tue. Thr. at 339 RC)

Instructor: Jack S.N. JEAN

Office Hours: 3-4 PM, 5:35-6PM, T, Thr; 334 RC, 775-5106, email: jack.jean@wright.edu

Course Format: The classes will meet at RC 339 for project discussion and implementations. Lectures are given only initially to clarify project requirements and to provide background information. There is no textbook and no test.

Project: The project consists of three phases and is to be done by a team of two or three students.

Phase 1: Define an application which utilizes an MC9S12C32 microcontroller board and requires communication between the microcontroller and a PC via a USB-I2C-IO card. The communication should move at least 1K bytes. One way communication is fine. One possible application is the extension of the CEG750 logic-analyzer project. In addition, the 6812 task should be of reasonable complexity that can demonstrate the advantage of using multi-tasking implementation. Some inter-task synchronization using semaphore(s) is required. Turn in a one or two page short description to outline (1) the application, (2) the multi-tasking implementation, (3) a non-multi-tasking implementation, and (4) the tradeoff analysis.

Phase 2: Implement the hardware and software for the application. Build your own USB-I2C-IO API functions and corresponding USB device drivers so that your application software can explore USB block transfer. The resulting data transfer rate should be as high as possible. The 6812 software should be a multi-tasking implementation based on a subset of SALVO-lite RTOS (real time OS).

Phase 3: Package the application so that the 6812 software is placed in flash memory and the PC software can be automatically installed.

Project Option: A student may propose to work on a different project as long as the instructor deems appropriate.

Final Report: Each team needs to turn in a 10-page (excluding schematics and code listing) final report at the end of the quarter to summarize and comment on the project results. Schematics and code listing should be attached. There will be penalty on typos and grammatical errors. Lessons learned or problems encountered in the project must be documented in the report. The report will not be returned.

Grading: Grading will be based on project result and instructor's (subjective) evaluation. Team members do not necessarily get the same grade.