**20-EECE-5117C/6017C—Embedded Systems—Fall 2017—Lecture Schedule**

**LAST UPDATED: September 22, 2017**

**IMPORTANT!! Schedule will be UPDATED throughout the semester. Updates will usually be announced, but you should still check the schedule frequently. There will never be “updates” to dates that have already occurred.**

**1.Note that this class has 3 lecture sessions per week (MWF 9:05-10:00 am) and one lecture session per week (M 6:00-8:50 pm). You ae expected to attend all lecture and lab sessions.**

**2. Assigned reading for lecture is to be done BEFORE class. Class discussions and in-class assignments will assume that you have completed the reading.**

**3. Lecture homework to be turned in is due at the beginning of class on the due date. Late homework will have credit deducted. Any lecture homework or quizzes to be done in blackboard will not be accepted after the blackboard due date and time.**

**4. Lab reports will also have specific due dates and times. Late lab reports will have credit deducted.**

**5. Occasionally you may be asked to demonstrate part of a lab project. You will be given a scheduled time for this demonstration. If the project is a group project, all group members need to attend the demonstration and to be able to answer any questions about it.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Lecture Topics / Notes** | | **Lecture Reading / References and Assignments** | **Lab-Topics / Notes** | | **Lab Reading / References and assignments** | |
| WK 1  08/21 | **Lec 1**—Overview—Embedded Systems, IoT;  Background Survey | | Please familiarize yourself with the course bbd page and make sure you can access all the links on the syllabus  EXERCISE (team): due Wed. Aug. 30—“PROCESSOR DESIGN”—answer the questions assigned on the lecture slides for the processor we are discussing in class | NO lab this week | | 1.Download Altera’s Quartus Lite software for students to your laptop (probably version 16.1 will work best; 17.0 is available, but the Altera documentation seems to be updated only to 16.1)  2. With your team, purchase an Altera DE-10 lite board and make sure you have a solderless breadboard, wires, potentiometer, etc. (kits available from amazon for about $15)  3. You may need to purchase an extension cord to use the lab outlets for your board power supply; if so you can obtain one at Walgreen’s  We will use the software in week 2 lab and the board in week 4 lab (no lab week 3, that is a holiday) | |
| :08/23 | **Lec 2**—Example: a small embedded processor |  | | |  | |  | |
| 08/25 | **Lec 3**—Example: PIC processors, PIC 16F887;  Discussion of embedded processor exercise |  | | |  | |  | |
|  |  |  | | | | |  | |
| WK 2  08/28 | **Lec 4—**  Processor control instructions—notation;  Introduction to Verilog and Design Automation | **Reading for this week:**  \* ch. 1, sections 1,2,3  and ch. 3  \*sections 1.1, 2, and 4 of the Wikipedia article  <http://en.wikipedia.org/wiki/IEEE_floating_point>  [optional:  A.if you are interested, read the article on D/A and A/D converters in the “documents and links” section)  B.if you are interested, read the 1991 article by Goldberg on floating pt arithmetic in the “documents and links” section]  **Homework for this week:**  **Due at the beginning of class Fri Sept 8:**  **GROUP WORK: each team can submit one set of answers; homework answers must be IN ORDER of questions stapled together, and labeled with your last names as they appear on the roster**  **INDIVIDUAL WORK: there will be a quiz on this homework in class on Friday Sept. 8 and some of these problems may appear on the midterm exam**  1-4.Marwedel, ch. 1—problems 7,8,10,11  5.Marwedel, ch. 3—BRIEFLY explain the meaning of figure 3.15 (1-2 short sentences)  6.Why might it be reasonable to use scratch pad memory in an embedded system but not cache memory?  7.Do problem 8 in Marwedel, chapter 3.  8. why is 2’s c integer arithmetic more “efficient” than sign-magnitude integer arithmetic?  9-22. Do the exercises on fixed point and floating point numbers in lecture 6.  23. Why do we need to create a test bench file for simulating an Altera design? | | | **LAB 1:**  1. Using the Quartus CAD tools (design and simulation)  2. Introduction to Verilog  3. Traffic light project—introduction  Please bring your laptop to lab with the Quartus tools installed | | \*Make sure your board is ordered  HW: team work due in lab 2 (processor code including testbench & traffic light specifications to hand in, along with code demo(s))    \*Verilog processor design  \*traffic light project: --requirements:  (some questions to answer:  --what states are needed?  --how will you use the board leds to code the states?  --how will you use pushbuttons to change states?  --how long will your light remain in each state?) | |
| 08/30 | **Lec 5**—Basic data types; Sensors; Processors;  Specs and tests**;**  Altera’s NIOS II Soft Core Processor;  Additional Embedded System Components;  I/O connections |  | | |  | |  | |
| 09/01 | **Lec 6—**  Integer arithmetic;  Multiplication:  Carry-save example (added to lecture 4) and Booth example;  A/D & D/A conversions; introduction to security;  fixed pt & floating pt numbers;  minimal instruction sets |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 3  09/04 | **Holiday-Labor Day-US** |  | | |  | |  | |
| 09/06 | **Lec 7**—  Analog & Digital Input & Output:  Reading Input from Sensors; Sampling;  Analog Output | **Reading for this week:**  \*Read the Wikipedia article on RISC architectures, <https://en.wikipedia.org/wiki/Reduced_instruction_set_computing>  \*Read the tutorial on Communication Protocols by Chintapalli in the Documents & Links Section  \*Read the article on Trust in Embedded Systems by Fisher in the Doeuments & Links Section  **Homework for this week:**  **GROUP WORK: 1 assignment per team, due to bbd before class Fri. Sept. 15**  **INDIVIDUAL WORK: quiz on this material in class on Monday Sept. 18:**  Answer questions 1,2,3,5,6,7,8,11  in chapter 6 of the tutorial by Chintapalli | | |  | |  | |
| 09/08 | **Lec 8**—  Processor Efficiency; Communication Protocols; Trust & Security in Embedded Systems (Introduction) |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 4  09/11 | **Lec 9**—System design approaches;  Specifying HW and software components;  Communication | **Reading for this week:**  Marwedel: 1.4,2.1,2.2,2.3  **Homework for this week:**  **GROUP WORK: due to bbd before class Fri. Sept. 22**  Specifications in UML for the “traffic light” you have been designing and comparison to FSM specification you did last week (see last slide of Lecture 11)  **INDIVIDUAL WORK: quiz in class Monday Sept. 25:** give a (UML) use case description similar to the description your team developed for the example on slide 17 of lecture 11 and associated system tests for a given embedded system (system to be specified on the quiz) | | | **LAB 2:**  1.Implement your processor on your board and demonstrate it  2. Implement the traffic light fsm and demonstrate it  3.read and understand the slides on how to choose an efficient set of tests for a 32-bit adder | | **Lab HW:**  1.Complete the 2 assignments for next week if you do not finish them in lab  2. Submit to bbd properly organized and commented Verilog code for each project; include testbenches  (put each project in a zip file for submission)  3. Read the additional slides on developing an efficient set of tests for an adder and use the strategy outlined there to define an efficient set of tests for your processor. Submit your test set as a doc file; include your explanation for why the tests you chose are “sufficient”. Grades will be awarded for this on a competitive basis; the team(s) with the smallest set that does the most complete testing job will receive the most points; other team grades will be relative to the “best” solutions | |
| 09/13 | **Lec 10**—Waterfall model; agile model; DFT; contracts |  | | |  | |  | |
| 09/15 | **Lec 11**-Requirements & specifications; UML |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 5  09/18 | **Lec 12**-Alternate modeling approaches; state machines, etc. | **Reading for this week:**  Marwedel: 2.4—2.10;  Look at marte site:  <http://www.omg.org/omgmarte/>  **Homework for this week:**  **GROUP work due to bbd by beginning of class Friday Sept. 29:**  1.Write 2 use cases (AND TESTS) for the interior lighting system of a car:  a. lights turn on when any door is opened; they turn of after N minutes  b. lights turn on when the driver switches from any other state to park; they turn off after M minutes  c. lights can be turned on and off manually  2.Develop a sequence diagram for each of your use cases and an object diagram showing necessary pathways between objects.  3.For each object you have defined, list black box tests that should be carried out.  4.Give a state machine description for overall control of your system  5.Give a BRIEF answer to the question: what is MARTE?  **INDIVIDUAL WORK: Quiz on Monday Oct. 2 on use cases, sequence diagrams, object diagrams, system tests, black box tests, state diagrams for a system to be given to you in class** | | | LAB 3:  A. Read Sections 3.5-3.10 and Chapter 4 of the DE10-Lite User Manual.  B. make sure you know how to use the control panel and run the demonstrations described in chapter 5 of the user manual.  (You may need the DE10-Lite software available for download at:  <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=234&No=1021&PartNo=4#section>).  C. work through the QSys and NIOS II tutorials available at  <https://www.altera.com/support/training/university/materials-tutorials.html#ifup-tut-co>  Do as much of this as you have time for in lab.  (more information on NIOS II is available at:  https://www.altera.com/products/processors/overview.html) | | LAB 3 HW:  Turn in a team document describing any problems you have with the assigned demos and tutorials. If you did not finish all the examples and tutorials, list what you still need to accomplish. | |
| 09/20 | **Lec 13**—UML—control structures in sequence diagrams; class exercise |  | | |  | |  | |
| 09/22 | **Lec 14**—UML  Component modeling (hardware modeling);  Additional UML tools;  Class/module diagrams and “contracts” |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 6  09/25 | **Lec 15**—Low level design and test;  error detection and correction | **Reading for this week:**  Read the paper on digital design verification provided in the Documents and Links section of the bbd page  **Homework for this week due to bbd by class time Oct. 6**  1.Apply the basis path method to the example program on **slide 10** of lecture 16  2.Give the number of additional bits that must be added to a n-bit word for (a) single error correction (b) double error detection for values of n = 8,16, 32,43, 128,256. In EACH CASE, also give the MAXIMUM number of message bits which could be sent.  3. a. show how to set the parity bits for  m15…m3 = 10101110101  show how to detect that m12gets flipped in transmission;  describe what happens if both m12 and m13are flipped—what does applying the Hamming code tell you in this case?  Use thenotation from the lecture, which is repeated here    **No quiz Oct. 9 (reading day); midterm exam Oct. 11; this material will be on midterm** | | | **L**ab 4:  Continue working on what was assigned for lab 3;  some VGA cables are available in 810 Rhodes | | LAB 4 HW:  See the bbd lab section to turn in your reports on   1. What you accomplished last week 2. What you accomplished this week 3. Any remaining issues related to using the components or understanding Qsys | |
| 09/27 | **Lec 16**-low level design and test continued; bit-level error correcting codes;  basis path testing—in-class exercise |  | | |  | |  | |
| 09/29 | **Lec 17**—Adding additional constraints; introduction to co-design |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 7  10/02 | **Lec 18**— Review of UML constructs;  Codesign example—Pqueue | **Reading for this week:**  1.Marwedel 8.1-8.4 (low-level testing—should have been assigned last week)  2. paper by Høeg et al.  **Homework for this week:**  1.Summarize (briefly) the process given in the paper by Høeg et al. for deciding how much of the queue should be implemented in hw and how much should be implemented in sw  2.Given the sequence of priorities: 11, 92, 87, 100, 59, 99, 8, 70, 16, 54  where the priority rule is that if A < B then A has higher priority than B, assume that one item is added at time 0 and then after every 3 time units and that the highest priority items are removed at times 5, 17, and 25  a.show the software heap after each number is added or removed  b.show what the array holding the heap data looks like at time 30  c. for the hardware queue implementation, show all the register contents at times 9, 18, and 27 | | |  | |  | |
| 10/04 | **Lec 19**- codesign-pqueue example (continued) |  | | |  | |  | |
| 10/06 | Review for Midterm Exam |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 8  10/09 | **Reading Day-No Class** |  | | |  | |  | |
| 10/11 | Midterm Exam |  | | |  | |  | |
| 10/13 | **Lec 20**—  Operating systems-overview  Midsemester Survey | Marwedel: 4.1 | | |  | | **Additional reading: read the course project description posted with this lab** | |
|  |  |  | | |  | |  | |
| WK 9  10/16 | **Lec 21**- RTOS |  | | |  | |  | |
| 10/18 | **Lec 22**—RTOS continued |  | | |  | |  | |
| 10/20 | **Lec 23**--RTOS cont.; Scheduling: task communication, deadlocks |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 10  10/23 | **Lec 24**-RTOS cont.—priority inversion;  Aperiodic tasks | **Reading for this week (lectures 22-25): FIX!!!!**  Marwedel--4.1; 6.1-6.2; 5.1-5.4  Ip-Comparison of VxWorks and RTLinux (available with Lecture 24 notes)  **Homework for this week:**  1.Answer the questions on RTOS scheduling on the last slide of lecture 22  2.briefly summarize the “performance analysis” paper provided with the slides for lecture 24. In particular, explain what features the author chose as being important and how the two systems evaluated differed. Which system is preferred? What evidence did the author rely on to make this choice?  3.What are the current versions of the two operating systems described in Ip’s paper?  4.HW question on fault-tree analysis, provided with the slides for lecture 25 | | |  | |  | |
| 10/25 | **Lec 25**-System evaluation-reliability, availability, maintainability |  | | |  | |  | |
| 10/27 | **Lec 26**—System evaluation—design optimization |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 11  10/30 | **Lec 27**-Tool chains; formal methods; industry and govt. standards | **Reading for this week:**  Marwedel 5.5-5.8 and 6.1-6.4  Hills-Embedded C—Traps and Pitfalls—document provided with Lecture 29 slides  **HW for this week:**  1.Do the problem on the last slide in lecture 26 | | |  | |  | |
| 11/01 | **Lec 28**-Application mapping; scheduling revisited; heterogeneous processors |  | | |  | |  | |
| 11/03 | **Lec 29**-Programming for embedded systems—C |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 12  11/06 | **Lec30**-Performance analysis and optimization I | **Reading for this week:**  Marwedel 7.1-7.4  **HW for this week (due in class FRIDAY Nov. 18):**  problem 1, section 7.5 of Marwedel | | |  | |  | |
| 11/08 | **Lec 31**-Performance analysis and optimization II |  | | |  | |  | |
| 11/10 | **Holiday-Veterans’ Day (Armistice Day, end of WW I, the “Great War”, at the “11th hour of the 11th day of the 11th month”)** |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 13  11/13 | **Lec 32-**Introduction to information theory | **References:**  Hamming codes: <http://www.cs.utsa.edu/~wagner/laws/hamming.html>  Reed-Solomon codes:  <http://www.cs.cmu.edu/afs/cs/project/pscico-guyb/realworld/www/reedsolomon/reed_solomon_codes.html>  Huffman coding (data compression):  [www.cs.nyu.edu/~melamed/courses/102/lectures/huffman.ppt](http://www.cs.nyu.edu/~melamed/courses/102/lectures/huffman.ppt) | | |  | |  | |
| 11/15 | **Lec 33—I/O:** useful codes— error correction revisited; code compression; coding exercises;  class exercise |  | | |  | |  | |
| 11/17 | **Lec 34**—Introduction to reliability, fault tolerance, safety and risk |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 14  11/20 | **Lec 35--**Security (overview): legacy code, obfuscation, & reverse engineering; buffer overflow;  side channel attacks; moats: virtual machines | **Reading for this week:**  Slides on avoiding risk by Koopman, included with lecture 34  Source: http://users.ece.cmu.edu/~koopman/pubs/koopman11\_escsv\_handouts.pdf  **HW for this week (due in class Monday 11/28):**  1.Build the tree for Huffman encoding of the 13 most frequent letters (use table from slide 11 of lecture 33)  2.use your table from 1 to give the Huffman encoding for a. DECIDE; **b.LEARN NOTE changed word!!)**  3.use the LZW encoding algorithm on slide 26 of lec. 33 to encode BBABAABAAABB (show your work!)  4.use the LZW decoding algorithm on slide 35 of lec. 33 to decode the string you created in part 3 (show your work) | | |  | |  | |
| 11/22 | **Project work** |  | | |  | |  | |
| 11/24 | **Holiday**-Thanksgiving Weekend-US |  | | |  | |  | |
|  |  |  | | |  | |  | |
| WK 15  11/27 | **Lec 36-**Security cont. | **Reading for this week:**  Paper by Kocher; additional papers to be added | | |  | |  | |
| 11/29 | **Lec 37-**Security cont. |  | | |  | |  | |
| 12/01 | **Review for Final;**  ***Class Survey;***  **Sign up for Project Demo / Report Presentation Times** |  | | |  | |  | |
|  |  |  | | |  | |  | |
| EXAMWEEK  12/04 | **Final Exam**-8-10 am  (final exam will probably be at least part take-home) |  | | |  | |  | |