Overview

- What are embedded systems?
  - Computers masquerading as non-computers

Embedded System Challenges

- Differs from general-purpose computing
  - Real-time constraints
  - Power constraints
  - Exotic hardware
  - Concurrency
  - Control systems
  - Signal processing
  - User interface
  - Physics

The Role of Languages

- Language shapes how you solve a problem.
- Java, C, C++ and their ilk designed for general-purpose systems programming.
- Do not address timing, concurrency.
- Domain-specific languages much more concise.
- Problem must fit the language.

Syllabus

- Software languages
  - Assembly
  - C
  - C++
  - Java

- Real-time operating systems
  - Concurrency
  - Meeting deadlines

- Dataflow languages
  - Signal processing

Syllabus

- Synchronous Languages
  - Global clock

- Hardware languages
  - Discrete-event modeling

- SystemC
  - Modeling hardware in C

Copyright © 2001 Stephen A. Edwards. All rights reserved
Goal of the Class

- Breadth
  - Knowledge of many different languages
  - Languages embody design methodologies
  - Broader knowledge, bigger “bag of tricks”

- Depth
  - Big design project
  - Gives you in-depth experience with one of the languages

How to Listen to a Lecture

- Ask questions

- Trick: Presenters do a better job when they think someone is listening

- I’m from Berkeley
  Every VW bus there sports this bumper sticker:

  QUESTION AUTHORITY

Required Text

- Languages for Digital Embedded Systems

- Available at Papyrus, 114th and Broadway
- Textbooks are downstairs
- Bookstore may run out: more can be ordered

Class Website

www.cs.columbia.edu/~sedwards/classes/2001/w4995-02

- Contains
  - Lecture slides
  - More project ideas
  - Pointers elsewhere
  - PDF/PS files
  - Detailed syllabus

Shortcut from www.cs.columbia.edu/~sedwards/

Class Structure

- Four homework assignments
  - Collaboration permitted, but work must be your own

- Two exams
  - One covering first half of class
  - One covering second half

- One big project
  - Project proposal due in two weeks
  - Literature review
  - Presentation of literature review
  - Presentation of final project
  - Final write-up

The Project

- Goal is to produce a workshop-caliber paper
  - You don’t have to submit it
  - But aim for that level

- Final writeup will consist of
  - Introduction
  - Literature survey
  - Technical details
  - Experimental results
  - Conclusions

- Literature survey due at midterm time
Project Ideas

- “Use the languages”
  - Compare the simulation performance of Verilog and System C
  - Compare the performance of an RTOS and Linux
  - Model a wristwatch in different languages
- “Analyze or implement the languages”
  - Verilog Hierarchy browser
  - Implement Kahn Process Networks
  - A Java-to-C translator
  - Compiled event-driven simulator for Esterel

- More ideas on the class web site

Project Proposal

- One-paragraph description of what you plan to do
- Due soon: September 26

- Use the web site for more ideas
- “Related Classes” lists classes at other institutions with additional project ideas
- Visit during office hours to discuss ideas

Collaboration

- You may collaborate on homework, but whatever you turn in must be your own
- Project teams should be two or three people

Late Policy

- No credit for late assignments unless you’ve made prior arrangements with me
- Homework is due at the beginning of class

One-minute Feedback

- Spend a minute at the end of each class writing a sentence or two.
- Examples of desired feedback:
  - “I really didn’t understand nondeterminism.”
  - “You spent too much time talking about structural Verilog.”
  - “I found the part about Ritchie’s hatred of Pascal really interesting.”

- Won’t be graded, but sign your name.