CSE477 – Hardware/Software Systems Design

- Welcome to CSE 477
  - Instructor: Carl Ebeling
  - Hardware Lab Manager: Chris Morgan
- Some basics
  - what is a system?
  - what is digital system design?
- Objectives of this class
  - designing real systems
  - combining hardware and software
  - e.g. projects: graphics display, user interfaces, integrated systems
- Class administration and logistics

What is a system (in our case, mostly digital)?

- A collection of components
  - work together to perform a function
  - judiciously chosen to meet some constraints
    - cost, size, power consumption, safety
  - communicates with its environment
    - human interaction
    - communication with other systems over wired or wireless networks
- One person's system is another's component
  - no universal categories of scope/size
  - subsystems need to be abstracted
- How is it documented?
  - interface specification
    - Use a component without knowing about internal design
  - functionality is often implicit in the interface spec

What is digital system design?

- Encompasses all computing systems
  - combination of hardware and software components
  - partitioning design into appropriate components is key
- Many technologies and components to choose from
  - programmable components (e.g., PLDs and FPGAs)
  - processors
  - memories
  - interfaces to analog world (e.g., A/D, D/A, special transducers)
  - input/output devices (e.g., buttons, pressure sensors, etc.)
  - communication links to environment (wired and wireless)
- The Art: Designing a good solution to a problem
  - choosing/defining the right components
  - meeting performance, cost, power, usability, safety constraints

Trends in digital system design

- Forces
  - cost (cheaper), size (smaller), weight (lighter), power (lower)
  - time-to-market (shorter)
  - upgradeability (in-the-field)
  - recyclability (reusable parts)
  - ubiquity (anywhere, everywhere, and highly task-specific)
  - standardization of interfaces (leverage)
- Effects
  - increased use of high-level languages: C, Verilog
  - high-level specifications: formal interface descriptions
  - automatic synthesis tools (hardware and software compilers)
  - programmable hardware (quick to prototype, reconfigurable)

Examples of embedded systems

Programmable hardware

- (Re)configurable hardware (e.g., PLDs, FPGAs)
  - high-performance interfaces
    - graphics controller
    - communications links
  - compute-intensive tasks
    - signal processing
    - graphics processing
- Microprocessors and microcontrollers
  - “low-performance” system component
    - microcontrollers are fast enough for most things
  - allows complex system implementation
    - user interfaces
    - co-ordination of multiple devices
  - integration of surrounding logic onto processor chip
    - timers, memories, configurable I/O ports

Systems-on-a-Chip

- processor core
- custom logic optimized to specific application
  - e.g. Viterbi decoder, MPEG2 decoder
- task-specific sensors and actuators
  - (e.g., MEMS)
- application specific instruction sets
  - (e.g., DSP processors)
- reconfigurable logic (FPGA components)

CSE 477

- Capstone design course
  - ties together curriculum with an intensive design experience
- For computer engineering
  - programming, data structures, operating systems
  - electronics, logic design, computer design
  - communication skills (oral and written), documentation of designs
  - group effort, interaction with users
- Project experience must have most of these elements
  - connecting thread through the discipline
  - invaluable opportunity to add to student portfolio
  - just what employers want to hear about
  - independently motivated experiences grad schools like to hear about

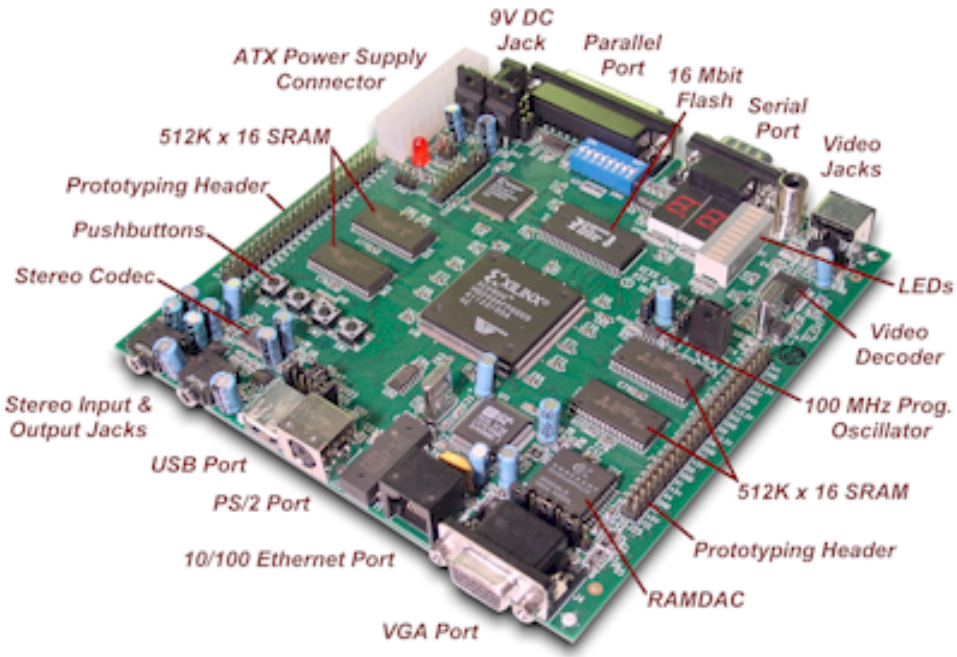
Course rationale

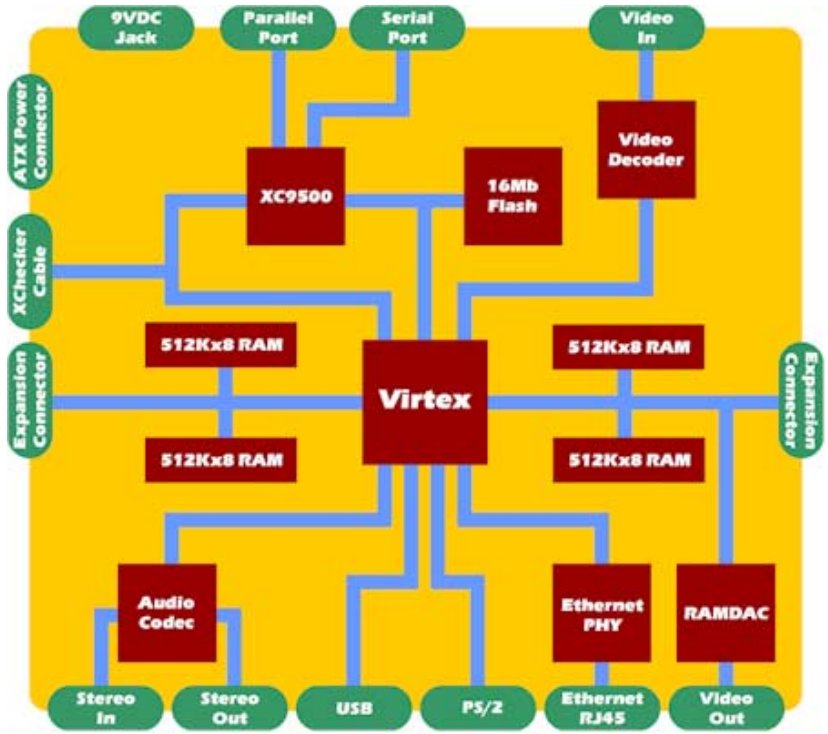
- Assignments and exams
  - learn/apply concepts presented in lecture
  - create infrastructure for possible use in projects
  - opportunity to evaluate individual creativity and understanding
  - gain familiarity with laboratory equipment and software
- Embedded system design project
  - wide variety of possibilities in a chosen domain
    - your chance to be creative
  - design reviews of other projects (learn from others' experience)
  - must be possible to complete in 10 **short** weeks
  - presentation/documentation
    - in-class presentations
    - web-based documentation

Project scope

- Project group: four students
- Project: combination of software and hardware
  - software: 8051 microcontroller, . . .
  - hardware: Xilinx FPGA, memory, . . .
  - interfaces: accelerometer, IR, wireless, display, audio codec, . . .
- Example project domain
  - graphics processor/interface to VGA monitor
  - hardware provides processing power
  - microcontroller provides interface/system integration
  - interesting user interfaces
    - Palm Pilot
    - remote interface
    - accelerometer-based interface

Project Platform – XESS XSV Board





Microcontroller

- Synthesizable 8051 Core
  - SOC IP component
  - Very common in ASICs
- We will interface hardware components to the 8051
  - function call to a filter
  - read/write a frame buffer
  - process ethernet packets
- Partitioning problem between HW and SW is key
  - Interface must be clean and well-defined
- We will simulate system using Xilinx Foundation

Project Organization

- Our focus is the project
- Define, specify, design, build and test a product
  - Concept to prototype in 10 weeks
- We will organize as an Advanced Development group
  - Marketing
  - Architecture
  - Engineering
  - Sales
- Labs and lectures are to support the project
  - Students will help drive

Group Dynamics

- Every class starts with a project meeting
  - Status reports
  - Planning
  - Designing
  - Design reviews
- VP/Engineering rotates
  - Sets agenda for meetings
  - Leads the meetings
  - A scribe will take and publish action items

The Process

- Product Definition (Marketing) [Week 1/2]
  - Marketing plan, marketing requirements
- Product Architecture (System Architect) [Week 3]
  - Block diagram, component functionality, high-level interfaces
- Detailed Design Specification (Engineering) [Week 4]
  - Component specs: interface + functionality
- Detailed Design (Engineering) [Week 5-8]
  - Verilog, schematics, test fixtures
- System Integration (Engineering) [Week 9/10]

- Product Demonstration and Documentation (Sales) [Week 11]

Example Project Ideas

- Real-time image processing, e.g. shrink/zoom/warp/sharpen/...
- Camera-based user interface
  - Gesture, laser guided, . . .
- Multi-modal user interfaces: camera, accelerometer, range-finder, audio, speech-recognition, ...
- Video games?
  - “pong” (with hand-motion interface)
  - fly-through with hand-sensing interface
  - animations
- Simple graphics card
  - Shaded polygon drawing, texture mapping
- Motion capture
- Audio signal-processing, e.g. equalizer
- Use your imagination

Course schedule

- First half
  - lectures
  - laboratory assignments
  - midterm exam
  - definition and specification of product
- Second half
  - detailed design and implementation of product
  - design reviews in the form of presentations
  - documentation on web
- Project meetings
  - Every class

Background (prerequisite) material

- Logic design
  - combinational logic
  - sequential logic
  - control/data-path
  - Verilog/simulation/synthesis
- Computer architecture
  - assembly language programming
  - computer organization
  - memory hierarchy
  - interrupt mechanisms
- Programming skills
  - facility with programming in C
  - software engineering skills
    - modularization, interface specifications

Refreshers

- Courses
  - CSE341 – Programming Languages
  - CSE370 – Introduction to Logic Design
  - CSE378 – Machine Organization / Assembly Language Programming
  - CSE467 – Advanced Digital Design
- Find your textbooks and notes from these courses
  - review chapters and lecture notes as topics come up
  - review written assignments and any projects

Goals for CSE477

- Lots of fun doing projects
  - cool project
  - amaze your friends and family (and future employers)
- Lots of learning in the process
  - you don't really understand it until you do it
  - great way to end your undergraduate career
  - killer interview material
- Produce some great demos
  - wow your friends and family