CS/EE 5710/6710 Project Ideas

The plan for the class project is that every team will design a digital system of some sort and implement it as a VLSI chip to the point where it could be sent off for fabrication. If teams want to fabricate and test their chip they can but that's not required. What is required is that each project be finished to the point where it could be built. So, the obvious question is – what makes a good project? The best projects are those that the team members are actually interested in. That is, rather than build a project because it's a requirement for the class, think of something that is interesting to you and build that! If you're stuck for interesting ideas, this document will propose some ideas to get you thinking.

One generic idea that's always popular is to build some sort of processor. A processor makes a great project because it's very clear what the processor is supposed to do: it executes instructions according to some instruction set architecture (ISA). The next question is then – what should the instruction set be? That depends on what application you would like your processor to be useful for. Before deciding on a processor ISA, think about an application domain and think about the types of instructions you'd need to support that application. Also think about the type of I/O you'd need to have to make a good processor.

Of course, processors are fun, but there are a gazillion other things you could do that are more specific than a processor. Chips that are built for specific applications are called ASICs for Application Specific Integrated Circuits. These special-purpose chips are everywhere that digital control is called for but a processor is too dramatic. Automotive controllers, cell phone controllers, washing machine controllers, digital camera controllers, etc. are just a very few of the types of things that usually involve ASICs. ASICs can be just about anything that includes a controller (i.e. a finite state machine) and a data path to achieve some specialized computation.

In 2001, everyone in the class built 16bit processors starting with the same instruction set (based on the CR16 from National Semiconductor). In 2002 the projects were: bit-serial FIR filter, digital match filter, 128 channel signal encoder for biomedical stuff, a DES encrypter/decrypter, a Christmas light and sound controller, a floating point unit, a UART, a digital metronome, a 16-bit CORDIC processor, and four different types of 8-bit microprocessors.

Here are some ideas from the Michigan folks who run a very similar VLSI class. They generally have everyone build a processor but have each group tailor their processor to a specific application and add functionality to the processor to support that functionality:

Projects

Treadmill Controller
Temperature/Pressure Controller
Implantable Cardiac Defibrillator Controller
Home Lighting Controller
Image Processing Controller for Boundary Recognition
FAX Machine Controller
MIDI Processor for Musical Instruments
Encoder for Data File Size Compression
Memory Tester
Position Sensing Micro controller for Automobiles
Alarm System Controller
Advanced Cruise Control System
Automobile Thermo Controller
Music Key Shifter
Vector Processing
Microwave Oven Controller
cochlear Implant
Fingerprint Detection Controller

Additions to Baseline

Extra Instructions not in Baseline
Special Instructions not in ISA
On-Chip Memory
Design-For-Test
Interrupts
Peripheral Timers/Counters
simple Serial I/O
Multiplier or MAC - a lot of work
A/D, D/A - a lot of work

Airbag Controller
Vending Machine Controllers
Real-Time Image Processor
Caller ID Routing Controller
DSP for Hand-Printed Character Recognition
Heating and Cooling Micro controller
Adaptive Equalization Filter for Cellular Communications
High Precision Motor Controller
Elevator Control System
Automatic Timing System for Swimming Races
Telephone Management System
Consumer Electrocardiograph
Automatic Brake System (ABS) Controller
FM Stereo Transmitter