Experiences Teaching an FPGA-based Embedded Systems Class

Stephen A. Edwards
Department of Computer Science, Columbia University
www.cs.columbia.edu/~sedwards
sedwards@cs.columbia.edu
The Boards

XESS XSB–300E

Spartan IIE Starter
Course Structure

- Length: 14-week semester
- Goal: Hardware/software codesign in C and VHDL
- Tools: Xilinx ISE and EDK (Microblaze soft processor, OPB)
- Labs: teach the tools and how to code VHDL
- Projects: Design-your-own, groups of four
### 2004 Lab Assignments

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count in decimal on 7-segment LEDs</td>
<td>C</td>
</tr>
<tr>
<td>Display “Hello world” using framebuffer</td>
<td>C</td>
</tr>
<tr>
<td>TV typewriter</td>
<td>C</td>
</tr>
<tr>
<td>Count in hex on 7-segment LEDs</td>
<td>VHDL</td>
</tr>
<tr>
<td>Make framebuffer display characters</td>
<td>VHDL</td>
</tr>
<tr>
<td>TV typewriter using character display</td>
<td>C &amp; VHDL</td>
</tr>
</tbody>
</table>
## 2005 Lab Assignments

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count in decimal on 7-segment LEDs</td>
<td>C</td>
</tr>
<tr>
<td>Terminal emulator (video controller supplied)</td>
<td>C</td>
</tr>
<tr>
<td>Reverse-engineer some VHDL</td>
<td>drawings</td>
</tr>
<tr>
<td>Sum the contents of a memory</td>
<td>VHDL</td>
</tr>
<tr>
<td>Complex multiplier as OPB peripheral</td>
<td>VHDL</td>
</tr>
<tr>
<td>SRAM controller for OPB</td>
<td>C &amp; VHDL</td>
</tr>
</tbody>
</table>
Disciplined Hardware Design
First: Block Diagrams

Char. RAM 2.5K

Font RAM 1.5K

Controller

Dout
Din
Addr

VSYNC
HSYNC

Load/Shift
BLANK

Video

Shift Register
Second: Timing Diagram

- **Clk**: Timing signal with periodic intervals.
- **CharAddr**: Address signal changing between $i-1$, $i$, and $i+1$.
- **LoadChar**: Load character signal.
- **CharData**: Character data signal changing between $i-1$, $i$, and $i+1$.
- **FontLoad**: Font load signal.
- **PixelData**: Pixel data signal changing between $i-1$, $i$, and $i+1$.
- **Load/Shift**: Load/shift signal.
- **Bit**: Bit signal with values from 0 to 7.
MemCycleFSM : process(OPB_Clk, OPB_Rst)
begin
    if OPB_Rst = '1' then
        MemCycle1 <= '0';
        MemCycle2 <= '0';
    elsif OPB_Clk'event and OPB_Clk = '1' then
        MemCycle2 <= MemCycle1;
        MemCycle1 <= ChipSelect;
    end if;
end process MemCycleFSM;

VGA_xferAck <= MemCycle2; -- OPB output

WE <= RamSelect when ChipSelect = '1' and RNW = '0' and OPB_Rst = '0'
else "00000000";

RST <= not RamSelect when ChipSelect = '1' and RNW = '1' and OPB_Rst = '0'
else "11111111";
Nifty Projects
Project: Scorched Earth

Based on 1990s DOS game. Custom video hardware (terrain + sprites + text). Very popular.
Project: MAYD

Project: JAYcam

Internet video camera. NTSC video in, digitized, packetized, sent over UDP, displayed by Java program.
Project: Nortsam

2048-point real-time FFT on 48 KHz stereo audio samples.
Project: Muddrover

Line-following Mindstorms robot with video vision.
Project: TAMF

Real-time video effects generator. Dynamically distorted still image.
Evaluation
### Columbia Course Evaluations

<table>
<thead>
<tr>
<th>Question</th>
<th>2004</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount Learned</td>
<td>3.72</td>
<td>4.04</td>
</tr>
<tr>
<td>Appropriateness of Workload</td>
<td>3.33</td>
<td>3.64</td>
</tr>
<tr>
<td>Overall Quality</td>
<td>3.74</td>
<td>3.89</td>
</tr>
</tbody>
</table>

Numbers are averages, with 0=poor and 5=excellent.
Selected Comments from 2005

“Tough class but learned a great deal. Recommended.”

“I’d like to see a lecture that goes into more detail about the way that the various files definitions and programs are used to create the hardware. We end up learning it in pieces but a more detailed overview would be useful since the tools are a key component of understanding this class.”

“The lectures didn’t seem to serve as much help for the assignments and project.”
Conclusions

Definitely a work-in-progress

Steep learning curve on tools. Should they be the focus of the lectures?

Students accustomed to quick recompile & run times; leads to “jungle programming.”

One digital design course not enough preparation

Software vastly easier to create than hardware

Should we make it easier for them?
The Class Website

All slides, lab assignments, lab files, etc.

http://www1.cs.columbia.edu/~sedwards/