Abstract

The use of CAE tools in digital logic and computer circuit design is growing in importance in two separate but closely related areas: first, by allowing the design team to bring new products to market faster, and second, to provide the tools needed to cope with the complexity and density of the integrated circuits used in the products. This paper describes our experiences in integrating commercial quality CAE tools from Viewlogic, Xilinx and Altera into our computer engineering courses. These tools include schematic entry and digital analysis in our Logic Circuits course, and the use of VHDL Simulation and Synthesis, and place and route tools for PLD and FPGAs in the VLSI and microprocessor courses.

Introduction

Until a few years ago our computer engineering sequence of courses at WPI was rather traditional in nature. This consisted of the following undergraduate courses:

- EE3801: Logic Circuits
- EE3803: Introduction to Microprocessor Systems
- EE3815: Introduction to VLSI
- EE4801: Microprocessor System Design.

The undergraduate courses have lecture and lab components with credit equivalent to 3 semester hours. The lab component requires attendance in a lab once a week for at least 3 hours.

Although CAE tools have been integrated into all four courses this paper concentrates on the changes made to the Logic Circuits and Introduction to VLSI Design courses (now called Digital System Design with VHDL).

The lab exercises for the Logic Circuits course consisted of simple combinational and sequential logic circuits. Students used paper and pencil to draw their circuits which they constructed in the lab using TTL 7400 series integrated circuits placed into prototyping boards and connected together and to switches and LEDs with wire. Although we tried to encourage students to carefully analyze their circuits prior to building them we found that they would try and build something as quickly as possible and when it didn’t work only then they would start to actually think about the operation of the circuit.

In some cases the circuit would fail to function correctly due to faulty components or wires not correctly inserted into the board or the wiring pattern not matching the drawn schematic. We found that rather than the lab exercise being used to complement the theoretical material taught in the class, the time was spent debugging either the circuit or the prototyping board.

A similar case existed in the Introduction to VLSI design class. Although CAD tools were used in this class they were rather primitive in nature and consisted more of a drafting tool to aid placing and connecting individual transistors on a die.

Schematic Entry and Digital Analysis

Approximately four years ago we obtained a major donation of CAE tools from Viewlogic Systems, Inc. as part of their university support program. The first donation consisted of the Powerview and Workview Plus tools running on UNIX workstations and PCs respectively. These tools have since been upgraded so that we now run the Workview Office tool suite exclusively on PCs running Windows 95.

To support the Logic Circuit class we use the following three main tools from Workview Office:

- ViewDraw
- ViewSim
- ViewTrace

ViewDraw is a schematic editor used to draw schematics and symbols.

ViewSim is a gate level simulator to simulate digital designs and back-annotate simulation values to ViewDraw designs.
**ViewTrace** is a waveform analyzer to display simulation results in the form of waveforms for analysis.

By integrating these tools into the lab exercises students can now use ViewDraw to quickly draw and modify their designs. More importantly however they perform a functional simulation on their design by using ViewSim and ViewTrace to verify that the circuit operates correctly. This is achieved by applying input signals to the circuit being designed and verifying that the outputs are correct. With the Viewlogic tools, design errors are relatively easy to find and correct.

Only after the circuit is shown to operate correctly (by providing suitable ViewSim and ViewTrace plots) are students allowed into the lab to actually build their circuit. Assuming there are no wiring errors or component failures the circuit should work first time! If the circuit does fail to work then students now have a firm base on which to start debugging their wiring.

We also quickly realized that the time now spent in the lab manually building a previously correctly simulated circuit was not very productive. Since we still feel that it is important that students do get their hands ‘dirty’ actually building circuits and monitoring the waveforms using test gear (logic probes, oscilloscopes) we have not entirely eliminated the hands-on component to the lab exercises. However we now spend more time using the schematic entry and digital analysis tools to allow students to design more complex combinational and sequential logic circuits. In the lab they then usually only build a part of the design. For example a complete four-bit ripple carry or look ahead adder would be designed but only a two-bit version would be built. For state machine design a traffic light controller with 12 or more states may be designed and simulated but only a sub-section would be built in the lab.

Another advantage of this arrangement is that the lab times are no longer fixed around a rigid three hour period each week. The Viewlogic software is installed on all the department PCs in a variety of labs so students may design and simulate their circuits any time. Also, now the student edition of Workview Office[1] is available students may use these tools on their own PCs.

**VHDL**

In the Logic Circuits revision described above the theoretical material has not changed much. The main changes were to the laboratory component. However for the Introduction to VLSI Design course (now renamed to Digital System Design with VHDL) we have made major revisions to the course content as well as to the labs. The main change has been to use VHDL for the design, simulation, and testing of VLSI devices. Also, instead of targeting custom integrated circuits the course focuses on using FPGAs from Xilinx.

The course concentrates on VHDL for synthesis so all the VHDL code examples have been written to be both simulatable and synthesizable. With this approach the subset of VHDL that is synthesizable is covered first. The course also introduces the concept of test benches to show that VHDL can be used to test, by simulation, a previously written synthesizable VHDL design. At this point additional VHDL statements and features (such as the assert statement and timing related statements) are covered.

As the design examples are introduced the corresponding synthesized hardware is shown in the form of a schematic (using generic gates or CLBs when a Xilinx FPGA is the target). This helps to show the equivalence between a VHDL code fragment and the resulting logic. This process is useful to ensure that the VHDL has been written correctly and also synthesized as expected by the tools. For example a piece of VHDL code describing a combinational circuit should not result in any latches or flip-flops being produced. Also if a four-bit shift resister is being described then only four flip-flops should be used in the synthesized design.

The laboratory exercises for this course start with tutorial exercises to show students how to use the Viewlogic and Xilinx tools. The design of a simple traffic light controller is used to show the complete design process from text entry of the VHDL description to the place and route of the synthesized design into an FPGA. We use Viewlogic’s Speedwave tool to carry out a functional simulation on the design and then use ViewSynthesis to synthesize the design into an XC4000 FPGA. Students then download this into a Xilinx development board with an XC4003 so students can see the traffic light controller working.

Subsequent laboratory exercises require students to write their own VHDL descriptions of different designs starting with combinational logic type circuits (design of an ALU) to state machine designs (vending machine).

**Conclusions**

We feel we have made good progress in integrating commercial quality CAE tools into most of our computer engineering courses. As a result we believe that our students are able to graduate with not only a good theoretical background but that they are also familiar with the types of tools being used by industry today.

Some problems with this approach are due to the learning process involved in introducing students to a wide variety of complex software, frustrating bugs in the
tools, and the cost (both financial and faculty time) associated with purchasing high-performance systems to run the tools and to continually update the lab exercises and tutorials as the products change.