

Using Synthesis, Simulation, and Hardware Emulation to Prototype a Pipelined RISC Computer System

James O. Hamblen
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250

Abstract

This paper describes a VHDL based rapid prototyping approach to simulate, synthesize, and implement a prototype computer system using commercial CAD tools, a meta assembler, a retargetable C compiler, and FPGAs in a hardware emulator. This methodology is utilized in a senior design laboratory sequence of two required courses for computer engineering students at Georgia Tech.

1. Introduction

CmpE 4500 and 4510, Computer Engineering Design I and II, are required five hour design courses taken in sequence by computer engineering seniors at the Georgia Institute of Technology. Students are already familiar with digital design, FPGAs, VHDL modeling, simulation, and synthesis, assembly language, C programming, and computer architecture from earlier required coursework[1,2]. Students work together in teams of two to four on the six-month computer design project. Students are assigned the task of developing hardware and software for a pipelined RISC processor of their own design. This required two quarter course sequence is structured to enable students to apply all they have learned towards designing, documenting, and building a complete computing system. The sequence was developed with active feedback from industry, and is quite aggressive in its goals.

2. Rapid Prototyping Methodology

The rapid prototyping methodology used in the design process in this laboratory is shown in fig. 1. VHDL synthesis [3] and a 30,000-gate hardware emulation device with automatic partitioning software are used to develop a working prototype on thirty-four large FPGAs. A meta assembler is configured by students to

produce machine language test programs. Lcc, a retargetable C compiler [4] with a code generator generator, lburg, is used to develop a cross compiler for the student processor design.

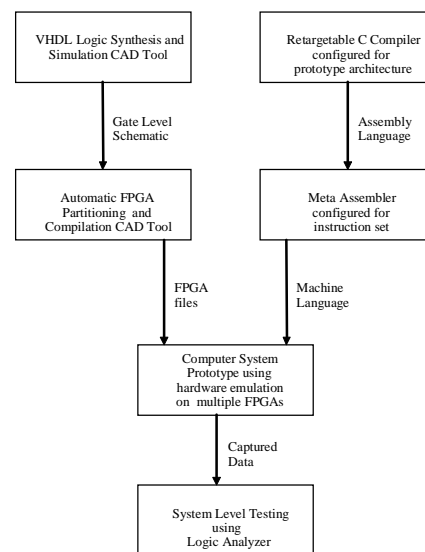


Figure 1. Rapid Prototyping Methodology

After VHDL synthesis, the resulting schematic or netlist is then implemented on the hardware emulation device using commercial software, ZyCAD's Concept Silicon and Xilinx's FPGA tools. This software automatically splits the design, a gate level netlist at this point, into multiple logic array chips. The hardware emulation system, a ZyCAD Paradigm RP, contains sixteen Xilinx 4010 10,000 gate FPGAs and eighteen Xilinx 3090s used to automatically interconnect the 4010 chips. In theory over 200,000 gates are available for a prototype design; however, due to routing and pin out constraints designs of up to 30,000 gates are typical. On large emulation systems, it is not uncommon to obtain only ten to fifteen percent utilization of the total FPGA gate resources. The automatic interconnection

using FPGAs eliminates the need for any manual wiring on the prototype system. This automatic interconnection feature makes this approach ideal for classroom projects. Automatic generation of the data files to produce the prototype systems requires two to three hours of workstation time for each design. Designs can be downloaded to the hardware emulator in few minutes. Prototype digital designs of 10,000 to 30,000 gates are possible with clock rates of several megahertz. This type of technology is widely used in industry to prototype ASICs, Application Specific Integrated Circuits, and recent microprocessor designs such as Intel's Pentium and AMD's K5 processor chips.

3. CmpE 4500/10 Course Sequence

In CmpE 4500, the first course in the sequence, students designed a synthesizable VHDL model of a pipelined RISC processor. Their VHDL design is then automatically synthesized into gate level logic and simulated using a popular commercial CAD tool, Synopsys. This tool has predefined VHDL based implementations of ALUs, multiplexers, and registers making the design process easier. Hardware for add, subtract and multiply operations can be automatically inferred from the symbols "+", "-", and "*" appearing in VHDL code. The synthesis process requires several hours of workstation time for each group.

In the first two weeks of the course, students divide into groups ranging in size from two to four and design their instruction set. Then using a meta assembler, students develop an assembler for their new machine. A benchmark program is hand-compiled from C into assembly language for use in the first course of the two-course sequence. At the same time as the assembler and benchmark programs are being developed, students design the data path architecture of a processor for their instruction set. Coding several benchmark programs at this point exposes any major deficiencies in the instruction set. Short machine language programs, assembled using the meta assembler, are run in the VHDL simulation to verify correct operation of the processor at the instruction level. After the simulation was successful, the design is synthesized. The netlist or schematic is then input to the automatic partitioning software

that is used to divide the design between multiple FPGAs in the hardware emulator.

In CmpE 4510, the second course in the sequence, students run machine language programs on the prototype machine and verify correct operation using an HP 16500 logic analysis system and by using programmed I/O to an attached data terminal. The terminal is interfaced to the emulator by synthesizing a UART in VHDL. In parallel with this task, work starts on retargeting the C compiler for their machine. Lcc, a public domain retargetable C compiler available for UNIX workstations and PCs, is used for this step. Assembly language files generated by the compiler are then input to the meta assembler to produce machine language code. Machine language files are then used to run several benchmark C programs on the prototype machine. A final test of several C benchmark programs is run to evaluate designs.

4. Conclusions

With the proper CAD tools and equipment it is now possible for groups of undergraduate students to design, simulate, and develop working prototypes of complex computer systems as part of their laboratory coursework. A rapid prototyping design laboratory demonstrates and unifies many of the ideas taught in numerous undergraduate classes and serves as an ideal capstone design course for computer engineering students.

References

- [1] J. Hamblen, "Using VHDL based modeling, synthesis, and simulation in an introductory computer architecture laboratory", *International Journal of Electrical Engineering Education*, vol. 33, no. 3, pp. 251-260, July 1996.
- [2] J. Hamblen, H. Owen, S. Yallamanchili and B. Dao, "Using Rapid Prototyping in Computer Architecture Design Laboratories", *IEEE Technical Committee on Computer Architecture Newsletter*, pp. 44-52, June 1996.
- [3] P. Kurup, and T. Abbasi, *Logic Synthesis Using Synopsys*, Boston, Mass: Kulwer Academic Publishers, 1996.
- [4] C. Fraser, C and D. Hanson, *A Retargetable C Compiler: Design and Implementation*, Redwood City, CA: Benjamin/Cummings Publishing, 1995.
<http://www.cs.princeton.edu/software/lcc>