Facilitating Interconnect-Based VLSI Design

Ramon Mangaser and Kenneth Rose
ECSE Department, Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Abstract

Since interconnect is becoming a limiting constraint for microelectronics technology, VLSI design curricula and supporting CAD tools require significant change. We describe the introduction of Rensselaer’s Interconnect Performance Estimator (RIPE) into a VLSI design class.

Introduction

The remarkable progress in microelectronics has been largely the result of shrinking feature sizes for MOSFET devices. As we enter the era of deep submicron feature sizes, interconnects are a major obstacle to improved performance because the delays associated with interconnect do not scale with reduced feature sizes. This dominance of interconnects rather than devices or logic gates in design decisions will have a profound influence on VLSI Design, obsoleting current approaches.

In an environment where physical design increasingly constrains what is possible, an early analysis of the constraints which interconnects impose on performance is equally important for system design and process technology. We are developing a CAD tool, the Rensselaer Interconnect Performance Estimator (RIPE) which estimates the wirability, performance, and power dissipation of processor chips on the basis of inputs reflecting device technology, interconnect materials and dimensions, and chip architecture. Our research is part of Rensselaer's new Center for Advanced Interconnect Science and Technology (CAIST) which is funded primarily by the Semiconductor Research Corporation (SRC).

We are introducing RIPE at Rensselaer as part of our course in Advanced VLSI Design for Spring 1997. In the following sections we describe RIPE, the issues involved in introducing it to the classroom, and our future plans for research and education in this area.

Performance Estimation

Performance estimation is used at a very early stage of design and technology development to determine the costs and benefits of particular decisions about system architecture, circuit design, or materials technology.

Programs like RIPE allow system designers and process technologists to examine relevant tradeoffs in detail. Such programs attempt to derive accurate estimates from minimum information about a particular technology or architecture. An early example of such a program has been described by Bakoglu [1]. RIPE 1.0 essentially implements the Bakoglu model. Recently, Sai-Halasz at IBM Yorktown Heights extensively revised the Bakoglu model to model performance trends in high-end processors [2]. RIPE 2.0 essentially implements the Sai-Halasz model. We have used RIPE to model the limits of multilevel interconnect technology [3].

Inputs to our current program, RIPE 2.5, include descriptions of the device technology, system, and interconnect. Outputs include estimates of wirability, clock frequency, and power dissipation. Key parameters for device technology include minimum feature size and power supply voltage; for interconnect we need estimates of the number of wiring levels, wiring pitch, and resistance and capacitance per unit length. The important global system parameters are total chip area, the number of transistors used for logic, the amount of on-chip cache SRAM in kilobytes, and the number of I/O signal pins. In addition, RIPE’s user has to know or guess something about the structure of the critical path that defines the cycle time of the microprocessor. This involves the logic depth of the critical path and the average fanout of the logic gates. To provide a good estimate of chip power dissipation, we need to know the average duty cycle (activity factor) for logic gates and clock circuitry.

Note that we do not know much about the chips whose performance we are estimating. In particular, we do not know netlists, wiring distributions, or even chip partitions except as rather crude approximations. We have to guess
activity factors and logic depths for the critical paths, since this information is not typically supplied by chip manufacturers. We do allow the user to select a wiring strategy which allows longer wires to have larger cross sections ('fat' wires). With all of these uncertainties, how good are RIPE's predictions? To judge this we have used available information about existing microprocessors to benchmark RIPE. Table 1 compares RIPE's simulations for power dissipation and clock frequency with actual values for PowerPC 620 and Pentium processors. The predictions are quite good, considering the degree of abstraction inherent in RIPE's model.

<table>
<thead>
<tr>
<th></th>
<th>PowerPC 620</th>
<th>Pentium</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Simulated</td>
<td>Actual</td>
</tr>
<tr>
<td></td>
<td>Simulated</td>
<td>Actual</td>
</tr>
<tr>
<td>Power dissipation (W)</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Clock frequency (MHz)</td>
<td>160</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>120</td>
</tr>
</tbody>
</table>

Table 1 - Comparison of RIPE Results with Actual Values

Introducing RIPE to the Classroom

RIPE provides a perspective on VLSI design which is very useful to the VLSI designer. Because it comprehensively considers the significant factors in architecture and technology, it allows the student designer to simulate the sensitivity of processor performance to technology and architecture decisions in the broadest sense. In a sense, RIPE simulation provides boundaries for possible chip designs in the same way that thermodynamics provides boundaries for possible engine and refrigerator designs.

However, like any simulation program, the student designer needs to understand the nature of the models in RIPE if they are to be used effectively. Analogies would be blind SPICE simulation of MOSFET circuits, without any understanding of the factors in the MOSFET model or the simulation of logic circuits with unit delay times and no hazard bands. Thus, we were particularly concerned to introduce RIPE so that student designers can improve their design expertise from a basic level to a more sophisticated level, depending on the degree to which default values are changed in the program.

In particular, this has meant developing a detailed manual for RIPE which begins with a simple model of technology and architecture factors but allows the user to explore particular aspects of design in greater depth. Since RIPE 2.5 is being placed online on the World Wide Web for users to explore, this manual is being developed in hypertext form, so that more complete explanations of particular aspects of RIPE's model are available by clicking on key words. The student is introduced to RIPE through a sequence of design questions which can be answered using RIPE.

Our approach to introducing RIPE into the classroom is consistent with our long-standing approach to VLSI design [4]. The CAD tools we teach and use reflect a blend of influences. In particular, we favor the use of tools which do not oversimplify design issues and for which we have source code. For us, the availability of source code means that tool maintenance and extension as well as tool use can become part of the curriculum. Typically, once the tools we develop become reasonably stable, introducing them into our course work allows us to beta-test tools on a representative group of novice users.

Future Plans

Down the road we envision RIPE as providing a front end for more detailed physical design tools. However, this will require a more complete design at the block and bus level. Block and bus information will allow RIPE to be extended to include the partitioning of circuit modules, floorplanning, placement, and global routing.

Acknowledgments

We would like to thank Bibiche Geuskens for developing RIPE, and John Reinhart for developing a World Wide Web site for RIPE. Funding for the development of RIPE has been provided by IBM and the SRC.

References