Curricular Integration for Next Generation in Microsystem Design Education

Rajesh K. Gupta
Information and Computer Science
University of California, Irvine, California 92697.
(rgupta@ics.uci.edu)

Abstract—Rapid advances in microelectronic technologies increasingly require microsystem designers to be also system architects. At the same time, architecting systems for submicron technologies can no longer be done in isolation from technology limitations and capabilities. To train future engineers, an effective integration of design technology and architecture is needed at a time when additional demands are being placed due to expanded offering in software/web-centric courses. In this paper, we discuss how this integration can be achieved by teaching microsystem organization and architecture that brings together existing courses in VLSI, CAD and Computer Architectures into a coherent theme.

I. INTRODUCTION

A typical personal in North America comes into contact with over eighty computing devices in the course of a day [3]. Each of these devices represents a system design challenge that would have been, only less than a decade ago, responded to by a major design team working on a single system design for several man-years. Instead, these designs must now be done by much smaller teams of hardware engineers (often a single engineer must carry out all the design tasks) in a matter of weeks. What makes this possible today is sophistication in design methods which have a strong basis in sound analytical principles and tools. In particular, the computer-aided design (CAD) for electronic circuits and systems has seen tremendous growth both in the development of theory and its application to practical IC design problems.

This growth in VLSI design has lead to a full circle from the 1987 CAVE prediction that “ICs will be designed by system architects [4].” Indeed, now we are looking at early signs that complete on-chip systems are being architected by VLSI designers [2]. As this trend continues, CAD techniques have now become a key enabling technology for a host of existing and new technology areas. In particular, emerging technology areas such as personal and portable communication systems provide new and exciting areas of growth in design and design automation.

To train future engineers for these growth areas, an effective integration of CAD and computer architecture curricula is urgently needed. Present day curricula in most computer engineering programs does not reflect this transition: computer architectures are still taught with most preliminary concepts about the implementation technology; and teaching in VLSI design, rarely if ever, considers application analysis and development. The underlying assumption being that the two problem domains can be isolated such that architectural design can be done in a technology-independent manner. This is no longer true and a situation is likely to get worse in coming generations of silicon processing where submicron technology trends (such as interconnect dominance) will influence system design at all levels.

II. TEACHING MICROSYSTEMS

Effective teaching of microsystem design requires an integrated presentation in the exploration of system architectures using computer-aided design tools. An important objective of the education in Computer Science and Engineering is that the students develop a knowledge of the design process, and have an experience in taking a design from concept, to specification, to a working prototype and finally be able to verify the system functionality [1]. Micro-system realization requires a good understanding of advances in three areas: component hardware, design automation and system architecture. While the areas of component hardware and computer aided design have been well integrated, the same is not true of computer architecture.

Historically teaching in computer architecture and computer-aided design have followed different evolutionary paths. The focus of CAD education has traditionally been on problems related to design, simulation and testing of micro-electronic devices and circuits. On the other hand, Computer architecture education has traditionally been concerned with macro-level organization issues at the hardware end and lately the attention has been shifting towards its integration with compiler technology and systems software. As a result, the CAD curriculum has not been integrated into mainstream computer architecture teaching. But there are early signs that this integration is beginning to happen. For example, current computer architecture courses are experimenting with ways to incorporate reprogrammable hardware technologies (such as FPGA) in building experimental sys-
architecture and CAD track subject material is presented when we evaluate architectures for use in systems for specific application areas.

The focus of the design technology is to present fundamental concepts, techniques and tools for the CAD at higher levels of abstractions. The emphasis must be on the identification of design problems and their mathematical treatment in the context of basic problems of coloring, covering and satisfiability. The course can use (and review) basic concepts from graph theory and algorithms. To challenge the student individual/team projects can be suggested (e.g., building a FPGA-based system using existing CAD tools), or design automation (e.g., building a CAD tool), or both components. These projects should be based on specific lectures which provide sufficient starting point for the interested student.

Thus, a single EDA course material will be sufficiently well contained for a student to be well-prepared to learn about in new developments in both design and design automation. At the same time, this course provides a necessary first step for the students who aspire to be at the frontier of design technology either as a researcher or practitioners.

IV. SUMMARY

Rapid advances on microsystem technologies have lead to fundamental changes in how these systems are put together and the division of tasks in engineering teams. System integration is often driven by system architecture, and system architecture is in-turn driven by technological realities. To prepare engineering graduates to meet incoming educational challenges, we need a curriculum that presents CAD concepts in a vertically integrated manner rather than being a collection of individual problems in different application areas and blends it effectively with concepts in computer architecture.

ACKNOWLEDGMENTS

The author would like to acknowledge support from National Science Foundation CAREER award number MIP 95-01 615 and from NSF EEC 89-43166.

REFERENCES