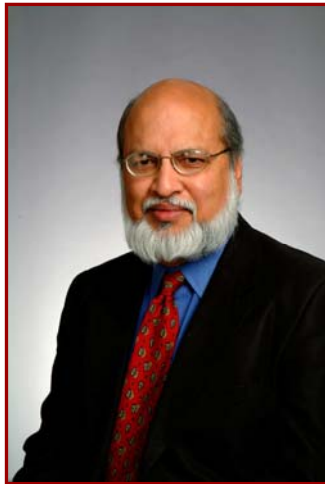


THURSDAY
OCTOBER 9, 2003

Scaife Hall Auditorium
Room 125

4:00 PM
Refreshments—3:30 PM



ARVIND

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Arvind is the Johnson Professor of Computer Science and Engineering at the Massachusetts Institute of Technology. As the Founder and President of Sandburst, a fabless semiconductor company, Arvind led the Company from its inception in June 2000 until his return to MIT in August 2002. His work at MIT on high-level specification and description of architectures and protocols using Term Rewriting Systems (TRSs), encompassing hardware synthesis as well as verification, laid the foundations for Sandburst and more recently Bluespec Inc. Previously, he contributed to the development of dynamic dataflow architectures, and together with Dr. R. S. Nikhil published the book "Implicit Parallel Programming in pH". Arvind is an IEEE Fellow and was awarded the Charles Babbage Outstanding Scientist Award in 1994. He has received the Distinguished Alumni Awards from the Indian Institute of Technology, Kanpur, and the University of Minnesota.

For more information:

<http://amp.ece.cmu.edu/ECESeminar>

James C. Hoe, ECE Seminar Host
jhoe@ece.cmu.edu

BLUESPEC: WHY CHIP DESIGN CAN'T BE LEFT TO EE'S

A 5M-gate ASIC is common place in 180nm technology today. We may see 50M to 100M-gate ASICs within a decade as technology improves to sub 90nm. (Fully custom-designed microprocessors are, of course, much denser and faster than ASICs but also require dramatically more design resources). Numerous problems related to process and design need to be solved before such large chips will become commonplace. Some of these problems, e.g., leaky transistors, porous oxide, controlling multiple V_t 's are clearly in the domain of EE's but computer scientists are much better equipped to solve the new problems related to the design-in-the-large. Large designs have to be conceived and executed in terms of a hierarchy of blocks. The hierarchy cannot be constructed in an ad hoc manner but should use some method of composition systematically.

Bluespec is a language/methodology that promotes correctness-by-construction. Its underlying execution model is based on atomic actions on state elements (flip-flops, registers, . . .), i.e., any legal behavior is explainable in a terms of a sequence of atomic actions on the state. Bluespec has facilities for expressing highly parameterized modules ("generic classes" in the language sense) and an expressive language to compose modules. The expressivity of the language has no limits because its semantics are orthogonal to hardware execution semantics—the source program is turned into a flat interconnection of modules by "static elaboration" during the compile phase.

In this talk I will present Bluespec via examples and show some of the designs done so far.

Bluespec is a joint work of people at MIT and Sandburst Corporation. 