UNIVERSITY *of* WISCONSIN
LA CROSSE

Distinguished Lecture Series
## in Computer Science

Monday, Oct. 4, 2010

Charles Thacker

Chuck Thacker has spent 40 years in several industrial research labs. He received the B.A. in physics from U.C. Berkeley in 1967. He joined Xerox PARC in 1970, where he was responsible for the hardware of a number of innovative systems, including the Alto, the first networked personal computer. In 1983, he joined the DEC Systems Research Center, where he was responsible for a number of networking and computing systems, including the AN1 and AN2 networks and the Firefly multiprocessor.

He joined Microsoft in 1997 to help establish the company's Cambridge, England laboratory. After returning to the U.S. in 1999, he joined the newly-formed Tablet PC group and managed the design of the first prototypes of this new device. He has also worked on low-cost computing devices for elementary education, and is currently working in Microsoft Research, where he leads a Computer Architecture group. He is the principal designer of the BEE3, and FPGA-based platform for research in architecture and algorithm acceleration, and has recently developed Beehive, a many-core computer system implemented on a single FPGA.

Thacker has published extensively and holds a number of U.S. patents in computer systems and networking. In 1984, he was awarded (B. Lampson and R. Taylor) the ACM's Software Systems Award for the development of the Alto. He is a distinguished alumnus of the Computer Science Department of the University of California, and holds an honorary doctorate from the Swiss Federal Institute of Technology (ETH). He is a member of the IEEE, a fellow of the ACM, a member of the American Academy of Arts and Sciences, and a member of the National Academy of Engineering, which in 2004 awarded him, (with A. Kay, B Lampson, and R. Taylor) the Charles Stark Draper prize. In 2007, the IEEE awarded him the John Von Neumann medal, and in 2010 he was awarded the A.M. Turing award by the ACM.
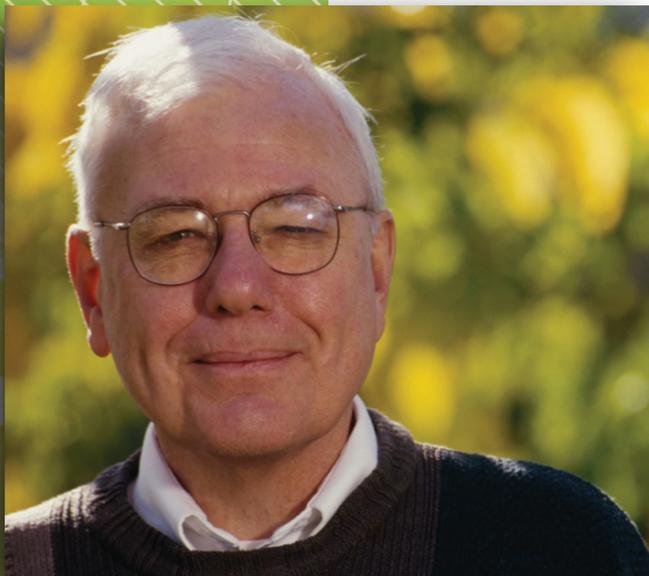
Schedule of Events

10:30 a.m. **Registration**
Cleary Alumni & Friends Center | UW-L Campus

11 a.m. **Symposium**
RARE: Rethinking Architectural Research and Education

By the late '80s, the cost of chip fabrication had increased to the point that it was no longer feasible for university researchers to do architectural experimentation on real systems. Groups could no longer do the sort of experiments that led to the establishment of companies such as Sun and MIPs. Simulation replaced implementation as the experimental vehicle of choice, and papers in the field became much more incremental as researchers focused on improvements to existing techniques, rather than the exploration of new ideas at scale.

The current limits on processor performance improvement provide a strong motivation to rethink the systems that we build and study. Fortunately, the development of better design tools and methodologies, coupled with the rapid progress of field-programmable hardware, may provide a way to change the way that architectural research and education are done.

In our laboratory, we have developed Beehive, a full-system implementation of a many-core processor, as well as its memory, peripherals and a supporting tool chain for software development. Beehive is simple enough that it can be rapidly understood and modified by individuals with little hardware experience. It enables full-system experimentation at the hardware-software boundary, using inexpensive development boards and tools provided by Xilinx.

Early experiences with Beehive, including experiences with its use as the basis for a short course at MIT in January (*http://projects.csail.mit.edu/beehive/*), and as a base for a hardware transactional memory system will be discussed.

4:30 p.m. **Registration**
Cleary Alumni & Friends Center UW-L Campus

5 p.m. **Keynote**
Improving the future by examining the past

During the last 50 years, the technology underlying computer systems has improved dramatically. As technology has evolved, designers have made a series of choices in the way it was applied in computers. In some cases, decisions that were made in the 20th century make less sense in the 21st. Conversely, paths not taken might now be more attractive given the state of technology today, particularly in light of the limits the field is facing, such as the increasing gap between processor speed and storage access times and the difficulty of cooling today's computers.

Thacker will discuss some of these choices and suggest some possible changes that might make computing better in the 21st century.