

SFGate.com

Intel, Microsoft fund university chip research

Tom Abate, Chronicle Staff Writer

Wednesday, March 19, 2008

After more than two decades of boosting chip performance by using two hardware tricks that work hand-in-glove, Intel Corp. has realized that one trick makes computers too hot, so it has been forced to redesign its microprocessors in a way that requires the invention of a whole new approach to PC software.

On Tuesday, Intel, joined by Microsoft Corp., said it will invest \$20 million over the next five years to fund software research at UC Berkeley and the University of Illinois at Urbana-Champaign to program a way around this unexpected dead-end in chip design.

In a nutshell, it was only a few years ago that Intel realized its approach to chip design - making a single microprocessor smaller and faster - would turn out processors too hot to handle, said UC Professor Kurt Keutzer, who is part of the local team.

Intel is already at work on a new approach to chip design that would put a slew of mini-processors, or cores, on a single sliver of silicon, the idea being that each could run slower and cooler but together they would still get the job done.

But before these new multicore chips can do anything useful for consumers or businesses, Intel and Microsoft must develop a new approach to software, called parallel processing. That is the part of the puzzle Intel and Microsoft hope the computer whizzes at Berkeley and Illinois can solve.

"Right now, it's a chicken-and-egg problem," said UC Berkeley Professor David Patterson, another scientist on the project. "This is a new way of processing that we don't know how to do yet."

Potential applications for multicore computers include faster and more realistic video processing, better speech recognition and quicker searching of databases.

There's another dimension to this dilemma that has the potential to disrupt the chip-making industry. Other firms, like Nvidia and IBM, also are building multicore processors, and if they solve the software-and-hardware problem first, that could tilt the balance of power in personal computing away from the industry's reigning co-rulers, Intel and Microsoft.

"This is a really new time in the history of computing, truly a paradigm shift," said David Bader, a professor of computing at the Georgia Institute of Technology who is an expert in parallel processing but is not directly associated with the Intel-Microsoft project.

"When there is a paradigm shift, there is a risk to established companies," agreed Marc Snir, the University of Illinois computer scientist who will lead efforts there to make sure that the "Wintel"

duo doesn't miss the paradigm shift.

But Intel - whose motto used to be "Only the paranoid survive" - is already feeling the heat from its Santa Clara neighbor, Nvidia, which is testing multicore chips with a new way to program them called Cuda.

"When we hear Intel say 'multicore' we say 'Welcome to the party,' " said Nvidia processor expert Ujesh Desai.

During a teleconference Tuesday, the parties involved in the research project explained how the intellectual fruits of this university-industry partnership would be shared. Computer scientists at both universities will be free to publish any and all findings, while Microsoft and Intel will be get nonexclusive licenses to any ideas they want, and be able to negotiate for more-exclusive terms on technologies they deem particularly useful.

Patterson, the UC scientist, said the research deal represents a great win for Berkeley, which was chosen over 25 institutions, including Stanford, to lead the project.

Patterson likened Berkeley's involvement in the parallel-processing software to its leadership in past technology leaps, like its pivotal role in the development of the Unix operating system and other technologies that "led to multibillion-dollar industries."

E-mail Tom Abate at tabate@sfchronicle.com.

<http://sfgate.com/cgi-bin/article.cgi?f=/c/a/2008/03/19/BUP7VM629.DTL>

This article appeared on page **C - 1** of the San Francisco Chronicle