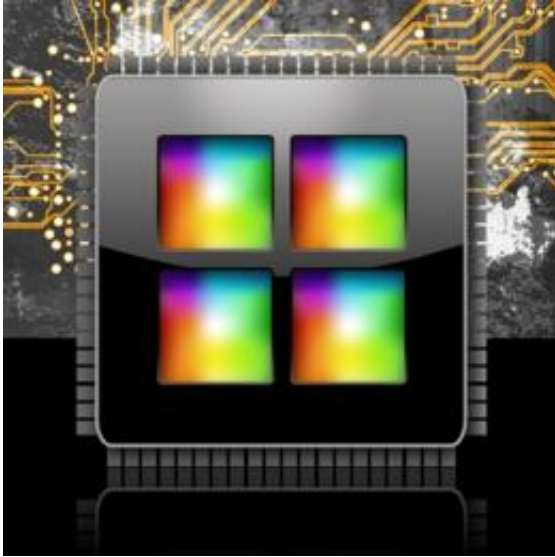


Designing the hardware

February 23 2011, by Larry Hardesty



Graphic: Christine Daniloff

Computer chips' clocks have stopped getting faster. To maintain the regular doubling of computer power that we now take for granted, chip makers have been giving chips more "cores," or processing units. But how to distribute computations across multiple cores is a hard problem, and this series of articles examines the different levels at which MIT researchers are tackling it, from hardware design up to the development of new programming languages.

With the multicore chips in today's personal computers, which might have four or six or even eight cores, splitting computational tasks hasn't proved a huge problem. If the chip is running four programs — say, a

word processor, an e-mail program, a Web browser and a media player — the operating system can assign each its own core. But in future chips, with hundreds or even thousands of cores, a single program will be split among multiple cores, which drastically complicates things. The cores will have to exchange data much more often; but in today’s chips, the connections between cores are much slower than the connections within cores. Cores executing a single program may also have to modify the same chunk of data, but the performance of the program could be radically different depending on which of them gets to it first. At MIT, a host of researchers are exploring how to reinvent chip architecture from the ground up, to ensure that adding more cores makes chips perform better, not worse.

In August 2010, the U.S. Department of Defense’s Defense Advanced Research Projects Agency announced that it was dividing almost \$80 million among four research teams as part of a “ubiquitous high-performance computing” initiative. Three of those teams are led by commercial chip manufacturers. The fourth, which includes researchers from Mercury Computer, Freescale, the University of Maryland and Lockheed Martin, is led by MIT’s Computer Science and Artificial Intelligence Lab and will concentrate on the development of multicore systems.

The MIT project, called Angstrom, involves 19 MIT researchers (so far) and is headed by Anant Agarwal, a professor in the Department of Electrical Engineering and Computer Science. In 2004, Agarwal cofounded the chip company Tiler to commercialize research he’d done at MIT, and today, Tiler’s 64-core processor is the state of the art for multicore technology.

One way to improve communication between cores, which the Angstrom project is investigating, is optical communication — using light instead of electricity to move data. Though prototype chips with optical-

communications systems have been built in the lab, they rely on exotic materials that are difficult to integrate into existing chip-manufacturing processes. Two of the Angstrom researchers are investigating optical-communications schemes that use more practical materials.

In early 2010, an MIT research group led by Lionel Kimerling, the Thomas Lord Professor of Materials Science and Engineering, demonstrated the first germanium laser. Germanium is already used in many commercial chips simply to improve the speed of electrical circuits, but it has much better optical properties than silicon. Another Angstrom member, Vladimir Stojanović of the Microsystems Technology Laboratory, is collaborating with several chip manufacturers to build prototype chips with polysilicon waveguides. Waveguides are ridges on the surface of a chip that can direct optical signals; polysilicon is a type of silicon that consists of tiny, distinct crystals of silicon clumped together. Typically used in the transistor element called the gate, polysilicon has been part of the standard chip-manufacturing process for decades.

Other Angstrom researchers, however, are working on improving electrical connections between cores. In today's multicore chips, adjacent cores typically have two high-capacity connections between them, which carry data in opposite directions, like the lanes of a two-lane highway. But in future chips, cores' bandwidth requirements could fluctuate wildly. A core performing a calculation that requires information updates from dozens of other cores would need much more receiving capacity than sending. But once it completes its calculation, it might have to broadcast the results, so its requirements would invert. Srinivasa Devadas, a professor in the Computer Science and Artificial Intelligence Lab, is researching chip designs in which cores are connected by eight or maybe 16 lower-capacity connections, each of which can carry data in either direction. As the bandwidth requirements of the chip change, so can the number of connections carrying data in

each direction. Devadas has demonstrated that small circuits connected to the cores can calculate the allotment of bandwidth and switch the direction of the connections in a single clock cycle.

In theory, a computer chip has two main components: a processor and a memory circuit. The processor retrieves data from the memory, performs an operation on it, then returns it to memory. But in practice, chips have for decades featured an additional, smaller memory circuit called a cache, which is closer to the processor, can be accessed much more rapidly than main memory, and stores frequently used data. The processor might perform dozens or hundreds of operations on a chunk of data in the cache before relinquishing it to memory.

In multicore chips, however, multiple cores may have cached copies of the same data. If one of the cores modifies its copy, all the other copies have to be updated. There are two general approaches to maintaining “cache coherence”: one is to keep a table of all the cached copies of the data, which has a cost in the computation time required to look up or modify entries in the table; the other is to simply broadcast any data updates to all the cores, which has a cost in bandwidth and, consequently, energy consumption. But Li-Shiuan Peh, an Angstrom researcher who joined the MIT faculty in 2009, is advocating yet a third approach. She has developed a system in which each core has its own “router,” which, like the routers in the Internet, knows only where to forward the data it receives. Peh’s simulations show that a network of routers is more computationally and energy efficient than either of the standard alternatives.

Whether the Angstrom project settles on electrical or optical connections remains to be seen. But Agarwal says that future multicore chips could well use both: Electrical connections would move data between individual cores; but optical connections would provide chip-wide broadcasts.

Not all the MIT faculty researching multicore architectures are affiliated with the Angstrom project, however. Jack Dennis is technically an emeritus professor, but together with colleagues at the University of Delaware and Rice University, he's received National Science Foundation funding to research a radically different multicore architecture. In Dennis' system, a computer's memory is divided into chunks of uniform size, each of which can store data but can also point to as many as 16 other chunks. If a data structure — say, a frame of video — is too large to fit in a single chunk, the system creates additional chunks to share the burden and links them to existing chunks.

Dennis' data chunks have three unusual properties. First, they are abstractions: Several chunks storing a single data structure might be found in a core's cache, but if the cache fills up, other chunks might be recruited from main memory or even from a flash drive. The system doesn't care how the chunks are instantiated. Second, and perhaps most counterintuitively, once a chunk has been created, it may never be altered. If a core performs an operation on data stored in a chunk, it must create a new chunk to store the results. This solves the problems of multiple cores trying to modify the same location in memory and of cache coherence. Once a chunk is no longer in use by any core, it's simply released for general use. Another chunk, storing the result of a computation, could take its place in the network of links. Finally, because any operation of any core could result in the creation or deletion of chunks, the allocation of the chunks is performed by circuits hard-wired into the chip, not by the computer's operating system. "As far as I know, nothing like this is going on anywhere else," Dennis says.

Whatever the architectural challenges posed by multicore computing, however, they're only the tip of the iceberg. The next installment in this series will begin to look at MIT research on software.

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology

Citation: Designing the hardware (2011, February 23) retrieved 14 August 2024 from <https://phys.org/news/2011-02-hardware.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.