
BOOK REVIEWS

The Connection Machine. W. Daniel Hillis. Cambridge, Massachusetts: The MIT Press, 1985.

The Connection Machine introduces a new type of parallel computer which may lead to radically new ways of thinking about computing, particularly in Artificial Intelligence. The author, Daniel Hillis, is the designer of the machine and the founder of Thinking Machines Corporation, a company committed to building "Connection Machines." Hillis' book describes the Connection Machine and the issues surrounding its design.

The Connection Machine is like an active memory made up of thousands, potentially millions, of small, simple, processors working simultaneously, each with its own tiny memory. This memory is controlled by a conventional, sequential, host computer. Programs running on the host can dynamically configure the connections between the processors in the Connection Machine, creating what are called *active data structures*. The idea behind this approach is to allow the machine to be reconfigured in a way which best suits the problem at hand; the requirements being that there are enough processors available and that their connections may be specified by software. Hillis provides examples illustrating how this concurrency can be applied to image processing, VLSI simulation, and semantic network knowledge representations.

Hillis devotes a chapter to illustrating how a Connection Machine could be programmed to take advantage of its power. He presents Connection Machine Lisp (CmLisp) which extends Common Lisp by adding operations to exploit the massive parallelism of the Connection Machine. The language maintains the serial control constructs of traditional Lisp but adds the ability to perform parallel operations on composite data structures, built from a basic data structure called a **xector**. A xector is similar to a set in which each member of the set is stored in its own processor/memory element. This distributed nature of xectors allows operations to be performed on all elements of a xector simultaneously. Since CmLisp's operations mirror the Connection Machine's hardware, the author's emphasis is on the language and its relationship to the machine. He does not attempt to address broader issues, or problems, in the design, implementation, or use of other parallel languages.

When the author deals with practical and theoretical design considerations such as the ideal processor/memory cell size and the physical communications topology he does so with respect to the trade-offs between cost and functionality and the constraints imposed by current technology. This discussion analyses many of the important aspects of parallel architecture design and in so doing, provides a good introduction to these issues. Even the readers with little background in conventional computer architecture

can read this material and gain insight into some of the issues surrounding the design of parallel machines. The treatment is not detailed enough to be used as a text on architecture design but it is illuminating and interesting to read.

Once the reader has been introduced to the basics of Connection Machine architecture, the author presents a description of a prototype called CM-1; a machine with 64K cells each with 4K bits of memory. Hillis discusses the custom VLSI chip, details of the simple processor cells, and the routers which are wired in a Boolean n-cube pattern to handle the communications between processing cells. At times the Connection Machine appears so different from current computers that it seems more akin to science fiction than to high technology. The description of the prototype helps dispense this aura of science fiction.

A machine with such radically different hardware provides the opportunity to implement radically different data structures. Hillis carefully describes the Connection Machine's active data structures which are special purpose computing machines dynamically designed to optimize operations on a given piece of data. Hillis describes sets, trees, butterflies, strings, arrays, and graphs, gives examples of their uses, shows their correspondence to CmLisp's xector notation, and ties their structure and implementation to the underlying structure of the Connection Machine. In addition, Hillis carefully examines issues of computational complexity surrounding the use of these various representations.

The book concludes with a philosophic look at the science of computation and its relation to modern physics. What follows this philosophic conclusion is a full and pragmatic annotated bibliography. This multi-disciplined bibliography spans the fields of artificial intelligence, parallel architectures, programming languages, psychology, philosophy, and mathematics.

The book, like its bibliography, has something to offer nearly everyone in the computer science community, even those with no background in computer architecture. In particular it will interest researchers and practitioners in artificial intelligence because it describes a new technology which may alter their approaches and improve their chances for success. Thus, the book provides a glimpse of what the next generation of computers will be like and it whets one's appetite for these exciting new machines.

Michael J. Black
GTE Government Systems
100 Ferguson Dr. 5G10
P.O. Box 7188
Mountain View, CA 94039