A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology

James R. Heath, Philip J. Kuekes, Gregory S. Snider, and R. Stanley Williams

Published: *Science, 12 Jun 1998, vol. 280, no. 5370, pp. 1716-1721.*

Teramac was the largest configurable computing system of its time. The full machine comprised two large computer cabinets housing 1,728 FPGA devices mounted on 16 large circuit boards. In early publications Teramac was distinguished for its capacity (1M gates), its compilation speed, and performance on several applications. Applications could be compiled onto the 1,728 FPGAs in approximately one hour using an HP workstation. The resulting applications demonstrated performance that exceeded CPUs by an order of magnitude or more.

Teramac broke new ground in several areas. At the time it was constructed, placing and routing a single FPGA device often took several hours. In contrast, Teramac was able to place and route a single FPGA device in seconds because the Teramac FPGA device architecture was optimized for fast place and route. Moreover, during the 1-hour or less compile time, Teramac also automatically partitioned a single application description into hundreds of circuits that were then mapped to the hundreds of individual Teramac FPGAs. At that time, a designer would manually partition designs because vendors did not provide automated partitioning software.

Perhaps the most compelling part of the Teramac story is that it was constructed with defective components. One-half of the multi-chip modules, 75% of the FPGAs and an unknown number of ribbon cables and printed circuit boards contained defects, yet applications compiled quickly and worked correctly. The key to Teramac's default tolerance was the combination of software that found and avoided defects and a rich routing architecture that tolerated defects.

I saw the effectiveness first hand. When HP kindly delivered a small, one-board, Teramac for use in my classroom at BYU, no one was told about the defect-tolerance scheme (because patents were being filed), and we successfully used the machine with undergraduates and graduate students, blissfully unaware of the large number of defects contained in our machine.

The Science article is noteworthy because it provides a general context for Teramac and suggests that Teramac shows how to build computing machines that function correctly in the presence of large numbers of defects---a necessity if we hope to manufacture useful computing machines with trillions of components using chemical syntheses that have statistical yield. The Science article provides details of the Teramac implementation not published elsewhere.

Endorsement by: Brad Hutchings, Professor, Brigham Young University

DOI: http://doi.org/10.1126/science.280.5370.1716