

High-Performance Numerical Computation using Field Programmable Gate Arrays (FPGAs)

Siddhartha Krishnamurthy*

Configurable Computing Laboratory, Bradley Department of and Electrical and Computer Engineering, Virginia Tech

Science and engineering research often requires the use of high-performance computational tools for simulations, modeling, and calculations. However, there is a caveat in the supercomputing industry: as scientists and engineers have realized the benefits of supercomputing, the demand for greater computing power has grown significantly. The need to execute inherently parallel applications of growing complexity has emerged in many fields ranging from the biosciences to mechanics. On standard PCs, executing inherently parallel operations is done in the same manner as executing sequential operations: one computation at a time. Conventional ways of shortening the computation time include acquiring faster computers, leasing time on supercomputers, and building clusters of computers to gain parallel processing speed up. These techniques can be expensive and frequently impractical. A more pragmatic and economical way of reducing the computation time is desired.

Field Programmable Gate Arrays (FPGAs) are increasingly being looked at as a solution. These reconfigurable processors can have dedicated circuits built into them to execute several operations concurrently. This research investigated how an FPGA can be used to solve equations for two computationally intensive and inherently parallel applications: Finite Difference Time Domain (FDTD) simulations for electromagnetism and the modeling of biochemical reaction networks. Furthermore, an investigation was performed into whether or not an FPGA implementation for these areas could practically and inexpensively improve computation speed over conventional implementations such as PCs and clusters.

The FDTD method requires the electromagnetic fields of thousands or millions of Yee Cells to be computed before a single iteration is completed. Usually, thousands of such iterations need to be run. Significant work on FDTD simulations using FPGAs has only been done in the past two years within a few research groups.

Understanding a biochemical reaction network involves finding chemical concentrations and constants of different processes in the network over time. The reaction network can be modeled using differential equations. A correct model is achieved when the differential equations emulate and predict empirical results. Different values for constants and concentrations need to be tried to arrive at the correct model. A substantial amount of computing power is needed to carry out this task.

The design of FPGA circuits for FDTD and biochemical reaction networks share common themes. Both areas involve the need to solve multiple instances of the same equations simultaneously with only the values of constants being varied in the instances. Solving these equations involves iterative methods. A single FPGA must be able to solve as many equations as possible in parallel to make it faster and competitive towards the conventional implementations. A combination of parallelism, pipelining, and reusing circuitry for sequential operations was used to come up with efficient circuits. A Xilinx[®] XC2V4000 FPGA with a 100MHz clock was used for building circuits in both areas of investigation. JHDL was used as the hardware description language to create circuits for the FPGA. IEEE standard thirty-two bit floating point numbers were used.

A circuit to solve a system of equations for Yee Cells of a two-dimensional FDTD simulation was designed, but it was not run on the FPGA. Using the techniques mentioned for designing efficient circuits, it was found that a single floating point adder circuit was sufficient to simulate a Yee Cell. Control/Counter and constant multiplier circuits were incorporated into the design. Unlike the floating point adder, they did not introduce clock cycle delays or use significant space on the FPGA. Using the delay time and the amount of space consumed by an adder circuit, the computation time and the number of circuits that could be implemented on a single FPGA was determined. The XC2V4000 FPGA could simulate up to 65 circuits or Yee Cells simultaneously and had enough RAM for 512 iterations. The runtime for this FPGA implementation was computed to be 0.184ms. It was faster than that of a 2.8GHz Pentium[®] 4 PC, which was approximately 5ms. However, the FPGA implementation as designed was impractical because thousands of Yee Cells and thousands of iterations could not be simulated. It was

* I would like to acknowledge my sponsors, Dr. Mark T. Jones of the Configurable Computing Laboratory and the Bradley Department of Electrical and Computer Engineering and Dr. Roland W. Lawrence of the Electromagnetics Research Branch of NASA Langley Research Center, for giving me the opportunity to perform research in reconfigurable computing. Their guidance, support, and advice proved invaluable in performing my work, and my research would not have been possible without them.

found that if such a simulation could be implemented on an FPGA, the computation speed increase would be orders of magnitude compared to a PC implementation. For this reason, methods of improving the FDTD method with FPGAs were proposed. These techniques include ways of making the circuit architecture and algorithm faster and more compact, using more advanced hardware, and developing programs to use FDTD more easily with FPGAs.

A circuit was designed that solved a system of seven first order nonlinear differential equations that modeled the biochemical reaction network underlying adenosine 3',5'-cyclic monophosphate (cAMP) oscillations in fields of chemotactic *Dictyostelium discoideum* cells. Before the system was designed, different numerical method techniques were explored to learn each of their benefits and how differential equation solvers could be programmed. The Forward Euler method was chosen. A circuit that solved a linear first order differential equation was designed and run on the FPGA. From this circuit, the circuit to model the biochemical reaction network was designed and is completing development. The computational speed increases and the practicality of this implementation are expected to be similar to the FDTD implementation.