

I. CURRENT TECHNOLOGY FALLS SHORT FOR SCIENTIFIC COMPUTING

Bio-Chemical Algorithms for performing molecular modeling of compounds use mathematical models of biological processes to increase screening sensitivity. These techniques promise to reduce the cost of computer assisted drug discovery by US\$133M and shorten the development cycle by 4 years. Accelerating the introduction of a drug also extends its patented market window increasing revenue by US\$8.5M per day. It is generally agreed that the pharmaceutical industry is more R&D intensive than any other technology sector. In 2001, about US\$61bn was invested in pharmaceutical R&D by companies worldwide. It is expected to reach US\$73bn by 2006. In one decade, between 1990 and 2000, the R&D investment increased by 121%. This expenditure has been consistently larger than the entire NIH budget. In the US alone, this was estimated to be in excess of US\$30bn [1].

Super computers with PC or workstation based clusters with million dollar contracts are being employed to respond to the computation demand in the fields of geophysics, biosciences, signal processing, computational fluid dynamics, and image and video processing. However cluster based approach suffers from diminishing return on investment as clusters grow larger and they suffer from the headroom problem, meaning that by the end of the design cycle you have a parallel system composed of previous generation's processors competing against current faster processors. Current methods to increase processing throughput depend mainly on creating larger clusters. This approach incurs high electrical and air conditioning costs, requires a large administration staff, takes up ever-increasing amounts of space, and ultimately provides a diminishing return on investment as clusters grow larger. On top of that, very few pharmaceutical R&D industries can afford to take this route. Thus, there is a requirement for a technology to design affordable high-performance computing resources.

Responding to the computation demand with general purpose processors limits ability to efficiently exploit potential parallel executions in highly concurrent types of algorithms. This is because of their generic architecture with complex hardware mechanisms such as out order issue, branch prediction etc. Limited amount of parallelism that each processor can exploit has made it necessary to use cluster based approach for scientific computing.

Special purpose programmable processors exploit the redundancies involved in these applications through the use of multiple application specific execution units (accelerator units) thus adding the flavors of VLIW, SIMD onto the processing core. These processors do not employ complex control mechanisms compared to general purpose processors however they are still far away from taking advantage of inherently parallel nature of the scientific applications. Execution units are activated on a need basis leaving valuable hardware resources idle.

Biochemical algorithms in the field of molecular modeling and computational molecular biology employ components such as molecular mechanics, solvation methods, comparison and analysis of protein structures etc. Each component provides a generic force field, and large portions of the code can be used for implementation of new force fields. Our analysis has shown that common computation modules and iterative processing are the two important characteristics of these fields. Additionally these algorithms have complex control data flow graph representations providing plenty of opportunity to extract parallelism and reconfigurability.

FPGAs have revolutionized the digital systems business after their introduction, almost 20 years ago. Today, state-of-the-art FPGAs allow accommodating digital systems with more than 10 million equivalent gates in its reconfiguration fabric alone. FPGAs have traditionally been used for hardware prototyping or as glue logic, but today's gates are clocked at rates that make the prototype nearly as fast as the end product and much more flexible. In sheer density, FPGAs are outpacing Moore's Law. Thus, they have the capability, especially aggregated on specially designed printed circuit boards, to become self-contained, high-end supercomputers. Moreover, their flexibility raises the possibility of meta-architecture; "morphing" hardware configurations with software as needed to improve efficiency, robustness, security and capability on-the-fly. With such a system the applications designer can design algorithms to maximize the use of silicon, something not possible with today's type-T (high ratio of computation to communication) and type-C (high communication bandwidth to global memory and between processors) machines. A third order of

programmability offers the designer the capacity to dynamically change the make-up and organization of the entire compute substrate - computational elements, communications topology and memory - to optimize the system for the problem at hand.

Over the past several years, key computational biology algorithms such as the Smith-Waterman and Hidden Markov computations have been implemented on FPGAs, and have enabled many computational analyses that were previously impractical. However, the complexity of programming the FPGAs and the inability to scale a single task across multiple logic boards with multiple gigabytes of memory have severely hampered the wide-spread use of this technology. FPGAs suffer from the drawback of being application agnostic, hence incur penalties of loss of clock cycles in redundant reconfigurations, generic routing, and poor memory architectures., which impact speed , power and area.

II. A NOVEL APPROACH TO SCIENTIFIC COMPUTING

A. Overview

High-throughput technologies have led to an exponential growth in the amount of data generated in the fields of geophysics, biosciences, signal processing, computational fluid dynamics, and image and video processing over the past several years. This data explosion is forcing scientists to search for innovative computational designs to meet the growing demands. The complexity, variety of techniques and tools, and the high computation, storage and I/O bandwidths associated with these applications pose several challenges, particularly from the points of scalability, resource utilization (in terms of area and energy) and real-time implementation. Current technologies fall short of providing low cost and flexible solutions in responding to processing demand. These drawbacks have lead is into exploration of the reconfigurable architecture design space [2,3,4,5,6,7,8].Configurable computers based on FPGAs are capable of accelerating suitable applications by several orders of magnitude when compared to traditional processor based architectures. There is evidence that reconfigurable systems can deliver 10X to 100X or greater improvement in computational efficiency for many computationally intensive problems by tailoring hardware allocations to match the needs of applications. This significant speed advantage is due to the highly parallel nature of FPGA (Field Programmable Gate Array) hardware. Fortunately, many problems in those application domains are inherently parallel, and benefit from concurrent computing models. However, Lookup table (LUT) based FPGAs suffer from the drawbacks of being application agnostic and hence incur penalties of loss of clock cycles in redundant reconfigurations. Due to generic routing and poor memory architectures routing resources constitute up to 70% of the total chip area which also impact speed, power and silicon area. Therefore, there is a need for custom configurable logic and interconnect design tailored to the computation characteristics of the target applications. Research methodology extracts control data flow graph of each application and pure data dependent sequence of basic blocks extracts common patterns and maps them onto configurable processing elements. The connectivity between the processing elements is then used to define the routing architecture. Applications are then mapped on the processing elements based on the architecture constraints; placed and routed using existing placement and routing algorithms. This methodology (figure1-1) proposes to provide optimum interconnection pathways, by allocating just enough switching and wiring resources by profiling the computational characteristics of the application. Only necessary and sufficient programmable circuitry are synthesized for each task and every gate processes useful information in each clock cycle, therefore increased circuit-packing density can be achieved. That way resulting special purpose chips can make ultimate use of available circuitry to run a specific algorithm. This research work proposes such needed methodology and proves to allocate just enough switching and wiring resources by using computation profiling as opposed to generic routing in FPGAs, exploiting inherent parallelism at function level instead of program level and providing flexible and scalable system with reusable libraries of functions. Tools implemented as part of the methodology will permit seamless algorithm design at a high level of abstraction and execution at a high level of efficiency in hardware, synthesizing layers of parallel execution structures all the way to the gate level.

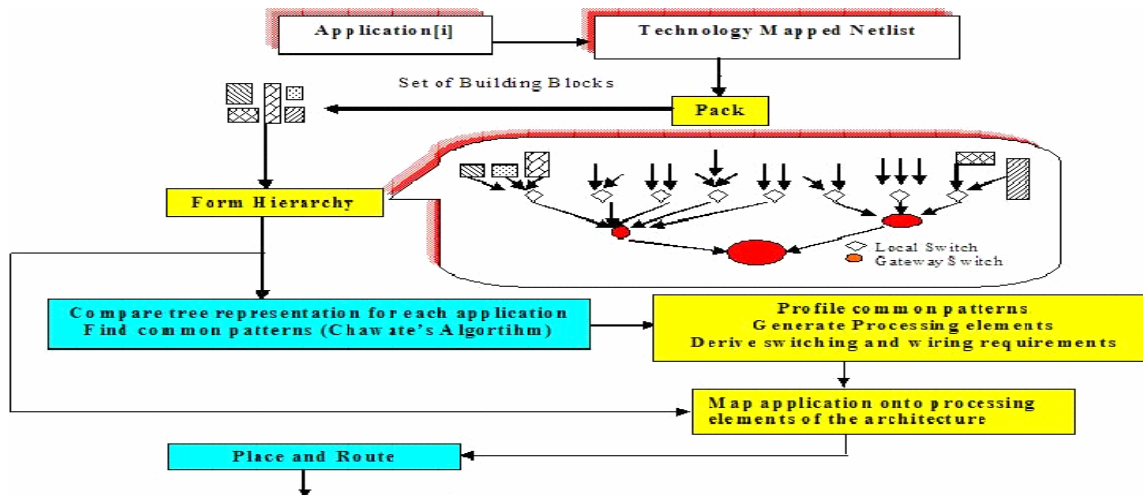


Figure-1 Methodology to Derive Application Specific Architectures

B. Motivation

1) Need for Software Platforms

- While advances in reconfigurable hardware technology is making programmable devices more attractive for many compute intensive applications, reconfigurable supercomputing for High-End Computing holds the promise of achieving breakthroughs through the development of reconfigurable computing software.
- Historically, programming FPGAs has been an extremely laborious and time-consuming process, requiring a low-level design language such as VHDL or Verilog. Alterations to algorithms implemented in an FPGA-such as changes in trace length, different sized output volumes, or different sized do-loops-required painstaking reprogramming by this same process. a completely new FPGA chip design would be required for each new data set, an effort requiring months of work. Therefore software platforms are needed for:
 - seamless algorithm design at a high level of abstraction
 - execution at a high level of efficiency in hardware,
 - synthesizing layers of parallel execution structures all the way to the gate level without the need for unwieldy hardware assembly languages such as VHDL.
- These tools should support both algorithm level programming and highly efficient mapping of algorithms into hardware. They should synthesize only the necessary and sufficient FPGA circuitry for each task and reconfigure tens of millions of gates to perform multiple tasks at the same time. As virtually every gate performs useful information transformative work in each clock cycle, increased circuit-packing density can be achieved. These improvements will fully exploit the inherent parallelism of FGPAs and leverage it in a way that more than compensates for slower FPGA clock speeds.

2) New Architectural Approach

- Current reconfigurable systems already integrate CPUs and reprogrammable devices. Beyond this, entire high-end systems of different types, e.g. vector and clustered systems, may be integrated. High bandwidth/low latency interconnects and switching between logic, memory and storage on both localized and distributed levels are possible to meet the demands of increased distance and complexity in future high-end supercomputers for scientific computing. In that sense there is a need to design application specific reconfigurable systems tailored to the computation characteristics of the target application. This can be achieved by employing set of software tools that carries out the following tasks:
 - profile and extract computation patterns of algorithms belonging to a target application domain in high level language
 - map computation patterns onto reconfigurable logic

- profile connectivity between reconfigurable logic units
- define an architecture that suits to the interconnection characteristics of the reconfigurable logic units
- map the target algorithm onto application specific architecture
 - if reconfigurable hardware has already been programmed then reconfigure only necessary logic
- That way resulting special purpose chips can make ultimate use of available circuitry to run a specific algorithm. This research work proposes such needed methodology and proves to
 - Allocate just enough swathing and wiring resources through the use of computation profiling as opposed to generic routing in FPGAS.
 - Exploit inherent parallelism at function level instead of program level,
 - Provide flexible and scalable system
 - Reusable libraries of functions can be easily modified and extended, similar to the way high level defined are modified and extended.
 - The core algorithms can be implemented to take advantage of the inherent parallelisms in FPGA technology, and can be converted from C or FORTRAN to library modules.

3) *Analysis of Target Applications*

- There is also a need to analyze a wide range of compute intensive applications with the perspective of the computational characteristics on configurable logic [7,9,10]. Researchers usually pick one or two applications from scientific computing and map them onto configurable processors to test their architecture. From that point of view there is a need for a more in depth analysis of a wide range of target applications and provide generic libraries of these applications in hardware description language.

III. AGENDA

- We have already made ahead start to derive such application specific architectures tailored to the computation characteristics of the target application [8]. For that purpose, we completed a through analysis of compute intensive routines from a wide range of applications in GNU Scientific Library, Biochemical Algorithms Library, Computational Fluid Dynamics Library, SPEC2000 and EEMBC 2.0 benchmark suites. This is the first in depth computation analysis of such variety of applications. Control data flow graph of each application has been generated. Common computation patterns among several CDFGs have been extracted. Routines selected from target applications have been implemented in Hardware Description Language (HDL), netlists have been generated using Xilinx and Synopsys tools. Numerical results of this study will be a much needed reference point in the academia. Intermediate results obtained through Xilinx Floorplanner on Virtex-II have proven to show that application specific reconfigurable computing has the potential to respond to the computation demand of such highly parallel applications in molecular mechanics, biochemical algorithms and computational fluid dynamics.
- Our priority is to develop a much needed fully automated software platform to help researchers design new architectures for high performance scientific computing. Our goal is contribute to the process of translating new knowledge into products with commercial value for public benefit under the umbrella of University of Arizona.

REFERENCES

- [1] Faiz Kermani, "Research and Development in the pharmaceutical Industry – options for success," a strategic report from Urch. 2004,
http://www.urchpublishing.com/publications/research_dev_exec_summary.pdf
- [2] A. Akoglu, A. Dasu, A. Sudarsanam, M. Srinivasan, and S. Panchanathan, "Pattern Recognition Tool to Detect Reconfigurable Patterns in MPEG4 Video Processing", Workshop on Parallel and Distributed Computing in Image Processing, Video Processing, and Multimedia (PDIVM'2002), April 2002, Fort Lauderdale, FL

- [3] A. Akoglu, A. Dasu and S. Panchanathan, "A Framework for Design of Heterogeneous Hierarchical Routing Architecture of a Dynamically Reconfigurable Application Specific Media Processor", High Performance Computing (HiPC), 10th International Conference, December 2003, Hyderabad, India
- [4] A. Akoglu, A. Dasu and S. Panchanathan, "Design of Fast and Efficient Hybrid-FPGAs for Numerically Intensive Applications", 7th annual Military and Aerospace Programmable Logic Devices (MAPLD) International Conference, September 2004, Washington, D.C.
- [5] A. Akoglu, A. Dasu and S. Panchanathan, "Application specific Hybrid-FPGA Design", IS&T/SPIE17th Symposium, Electronic Imaging Science and Technology, January 2005, San Jose, CA
- [6] A. Akoglu and S. Panchanathan, "Application Specific Reconfigurable Architecture Design Methodology", International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA'04), June 2005, Las Vegas, NV
- [7] A. Dasu, A. Akoglu, and S. Panchanathan, "Reconfigurable Processing" *U.S Provisional Patent Application* filed on February 5, 2003.
- [8] A. Akoglu, "Application specific reconfigurable architecture design methodology", A Dissertation Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy, Arizona State University, July 2005