

# Research Projects by Graduate Students

## Supervised by: Shawki Areibi

Research projects are focused in the areas of **VLSI CAD, Combinatorial Optimization, Advanced Heuristic Search Techniques** and **Embedded Systems (Hardware/Software Co-design)** and **Reconfigurable Architectures**.

### **A Clustering Utility-Based Approach for ASIC Design**

By Matt Thompson, June 2000

This research presents two new approaches for dealing with the high complexity of ASIC design. A novel utility-based search technique is applied to iterative improvement in the standard-cell placement problem. Utility theory is used to guide a deterministic (greedy) search heuristic in finding a local minimum quickly by ranking moves based on an estimate of their proximity to an optimal location. Moves are then chosen that are statistically more likely to improve than if the moves were chosen randomly, greatly increasing the rate of convergence. Then a new hierarchical clustering heuristic is presented which clusters a standard-cell circuit by greedily collapsing net hyperedges by size, but not permitting very large clusters from forming. The clustering heuristic demonstrates excellent characteristics for reducing the execution time of standard-cell placement while achieving better results compared to non-clustered circuit placement and placement using other edge-based clustering methods.

### **Area/Congestion-Driven Placement for VLSI Circuit Layout**

By Zhen Yang, July 2003

A VLSI chip can today contain millions of transistors and is expected to contain more than 1 billion transistors in the next decade. In order to handle this rapid growth in integration technology, the design procedure is therefore divided into a sequence of design steps. Circuit layout is the design step in which a physical realization of a circuit is obtained from its functional description. Circuit placement is one of the key sub-problems of the physical design automation which involves finding the best position for all elements of the circuit while minimizing the total estimated interconnecting wire length. In this research, several global placement algorithms are constructed and compared. Both flat and hierarchical approaches are implemented to find the effectiveness of these approaches. Experiments conducted indicate that the Attractor-Repeller Placer (ARP) method produces the best results and a hierarchical approach can reduce the computation time of ARP by almost 85%.

# A Reconfigurable Hardware Implementation of Genetic Algorithms for VLSI CAD Design

By Gurwant Koonar, August 2003

In recent years there has been a great interest in accelerating time consuming algorithms that solve large combinatorial optimization problems. The advent of high density field programmable gate arrays in combination with efficient synthesis tools have enabled the production of custom machines for such difficult problems. Genetic Algorithms (GAs) are robust techniques based on natural selection that can be used to solve a wide range of problems, including circuit partitioning. Although, a GA can provide very good solutions for such problems the amount of computations and iterations required for this method is enormous. As a result, software implementations of GA can become extremely slow for large circuit partitioning problems. In this research project, an architecture for implementing GAs on a Field Programmable Gate Array (FPGA) is presented. The architecture employs a combination of pipelining and parallelization to achieve substantial speedups. The GA accelerator proposed in this paper achieves more than 100× improvement in processing speed over its counterpart software implementation.

# A Reconfigurable Computing Architecture for Implementing Artificial Neural Networks on FPGA

By Kristian Nichols, December 2003

Artificial Neural Networks (ANNs), and the backpropagation algorithm in particular, is a form of artificial intelligence that has traditionally suffered from slow training and lack of clear methodology to determine network topology before training starts. Past researchers have used reconfigurable computing as one means of accelerating ANN testing. The goal of this research was to learn how recent improvements in the tools and methodologies used in reconfigurable computing have helped advanced the field, and thus, strengthened its applicability towards accelerating ANNs. A new FPGA-based ANN architecture, called RTR-MANN, was created to demonstrate the performance enhancements gained from using current-generation tools and methodologies. RTR-MANN was shown to have an order of magnitude more scalability and *functional density* compared to older-generation FPGA-based ANN architectures. In addition, use of a new system design methodology (via High-level Language) led to a more intuitive verification / validation phase, which was an order of magnitude faster compared traditional HDL simulators.

# A Fast Heuristic Technique for FPGA Placement based on Multilevel Clustering

By Du Peng, December 2003

Field-Programmable Gate Arrays (FPGAs) are semiconductor chips that can realize most digital circuits on site by specifying programmable logic and their interconnections. The use of FPGAs has grown almost exponentially because they dramatically reduce design turn-around time and start-up cost for electronic products compared with traditional Application-Specific Integrated Circuits (ASICs). A set of Computer-Aided Design (CAD) tools is required to *compile* hardware description into bitstream files that are used to configure the target FPGA to implement the desired circuits. Currently, the compile time, which is dominated by *placement* and *routing* time can easily take hours or even days to complete for large (8-million gate) FPGAs. With 40-million gate FPGAs on the horizon, these prohibitively long compile times may nullify the time-to-market advantage of FPGAs. This research presents two novel placement heuristics that significantly reduce the amount of computation time required to achieve high-quality placements, compared with VPR, which is considered to be a state-of-the-art FPGA placement and routing tool.

# Low-Power Multi-Threshold CMOS Circuits Optimization and CAD Tool Design

By Wenxin Wang, May 2004

Over the last two decades, low-power design has become a concern in digital VLSI design, especially for portable and high performance systems. As technology scales into the Deep Sub-Micron (DSM) regime, standby subthreshold leakage power increases exponentially with the reduction of the threshold voltage. Therefore, effective leakage minimization techniques are becoming a necessity. Multi-Threshold CMOS (MTCMOS) has emerged as an effective circuit-level technique that attains a high performance, while standby subthreshold leakage is minimized by cutting off the power of the inactive blocks by sleep transistors. This research presents several techniques to solve the leakage power problem in the form of Genetic Algorithms, Set-Covering and Set Partitioning. In addition, an automatic MTCMOS design environment is devised and integrated into the Canadian Microelectronics Corporation (CMC) digital ASIC design flow.

# Sequential/Parallel Global Routing Algorithms for VLSI Standard Cells

By Hao Sun, May 2004

Global routing is an important and time consuming step in VLSI physical design automation. In order to effectively solve this problem, three standard cell global routers are presented in the form of (i) a Sequential Heuristic Global Router (SHGR) (ii) a Hierarchical Heuristic Global Router (HHGR) and (iii) an Integer Linear Programming (ILP) based Global Router (ILP-GR). The objective of SHGR is to find the minimum cost path for each net by enumerating a set of possible 2-bend routes. It achieves 16% improvement over wirelength obtained by placement based on Half Perimeter Wire Length (HPWL) method. HHGR is a pre-processing technique that provides good routing estimations in reasonable time. It can predict the routability of circuits and reduces CPU time on average by 82.5% compared with SHGR. ILP-GR formulates the global routing problem as two Integer Linear Programming (ILP) models. These ILP models are further relaxed to Linear Programming (LP) models to reduce computation time. ILP-GR overcomes the net-ordering problem and provides optimal solutions which can be used as upper bounds for other global routers. It achieves on average an improvement of 6.7% over wirelength solutions produced by the SHGR technique.

# A HARDWARE/SOFTWARE CO-DESIGN APPROACH FOR FACE RECOGNITION BY ARTIFICIAL NEURAL NETWORKS

By Shaw Li, August 2004

Artificial Neural Networks (ANNs), and the multi-layer perceptrons trained using an error *backpropagation* algorithm (MLP-BP) in particular, have proven to be an effective method today in many applications. However, this technique has suffered from slow training and lack of clear methodology to determine the network topology before training starts. The speedup to this algorithm is desired so that reasonable experimentation with various network topologies and on-line working are possible. Although Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs) can achieve speedup over a general processor, the flexibility is a tradeoff with speed. To balance them, an embedded computing system consisting both a processor with dedicated hardware on an FPGA chip is proposed. Results obtained show that this system achieves 1.69 speedup (Amdahl's Law) over the system which consists of only a processor on an FPGA chip. At the same time, the flexibility is preserved to some extent.

# **A MEMETIC ALGORITHM IMPLEMENTATION ON A FPGA FOR VLSI CIRCUIT PARTITIONING**

**By Stephen Coe, August 2004**

During the last decade, the complexity and size of circuits have been rapidly increasing, placing a stressing demand on industry for faster and more efficient CAD tools for VLSI design. One major problem is the computational requirements for optimizing the place and route operations of a VLSI Circuit. Thus, this research investigates the feasibility of using Reconfigurable Computing platforms to improve the performance of CAD optimization algorithms for the VLSI circuit partition problem. The proposed Reconfigurable Computing Genetic Algorithm architecture achieved a 5x speedup over conventional software implementation while maintaining 85% solution quality. Furthermore, a Reconfigurable computing based Memetic Algorithm improved upon this solution while using a fraction of the execution time required by the conventional software based approaches. This research also investigates the tradeoff of developing Reconfigurable computing solutions using a high-level language (Handel-C) vs a low-level language (VHDL). Implementing a Local Search algorithm in VHDL produced speedups of nearly twice that of the Handel-C implementation while requiring five times more development time. This speedup is a result of optimizing the VHDL architecture to target the specific FPGA hardware.

# **CONSTRUCTIVE/ITERATIVE BASED TECHNIQUE FOR FPGA PLACEMENT**

**By Xiaojun Bao, August 2004**

Today the logic capacity of Field-Programmable Gate Arrays (FPGAs) has increased dramatically (up to 10-million gates) that prohibitively long compile times may adversely affect instant manufacturability of FPGAs and become intolerable to users seeking very high speed compile. This research presents several heuristic techniques and investigates the effectiveness and efficiency of heuristics and meta-heuristics for FPGA placement. In constructive based heuristics, Cluster Seed Search (CSS) is developed to improve averagely random initial solutions by 19%; GRASP and a Partitioning based method are also implemented and achieve 25% and 44% improvement respectively. In iterative based heuristics, an enhanced local search technique is implemented in two forms: Simple Local Search (SLS) and Immediate Neighbourhood Local Search (INLS), which both achieve 50% improvement quickly. A Tabu Search (TS) technique and a Genetic Algorithm approach are also implemented to further enhance solution quality. Results obtained indicate that both Tabu Search and Genetic Algorithms can enhance solution for FPGA placement and produce on average 74% and 20% improvement in reasonable time.

# MOBILE ROBOTS PATH PLANNING OPTIMIZATION IN STATIC AND DYNAMIC ENVIRONMENTS

By Ahmed Elshamli, August 2004

Path planning for mobile robots is a complex problem. The solution should not only guarantee a collision-free path with minimum traveling distance, but also provide a smooth and clear path. In this research, a Genetic Algorithm Planner (GAP) is proposed for solving the path planning problem in static and dynamic mobile robot environments. The GAP is based on a variable-length representation. A generic fitness function is used to combine the objectives of the problem. Different evolutionary operators are applied some are random-based, and others use problem-specific domain knowledge. Various techniques are investigated to ensure that the GAP is appropriate for dynamic environments. To further increase the efficiency of the GAP, an Island-based GA (IGA) is developed on a ring topology and Message Passing Interface (MPI) library is utilized to implement the IGA. A new Local Search (LS) is also developed in this research and different approaches are examined for combining the LS algorithm with the GAP to obtain superior solutions.

# SEQUENTIAL/PARALLEL HEURISTIC ALGORITHMS FOR VLSI STANDARD CELL PLACEMENT

By Guangfa Lu, September 2004

With advanced sub-micron technologies, the exponentially increasing number of transistors on a VLSI chip causes placement within physical design automation to become more and more important and consequently extremely complicated and time consuming. This research addresses the placement for VLSI standard cell designs. A number of heuristic optimization techniques for placement are studied and implemented, in particular, local search, Tabu Search, Simulated Annealing and Genetic Algorithm. The Tabu Search reduces wire length on average by 52.4% while Simulated Annealing yields a 61% improvement on average. Furthermore, two parallel island-based GA models are implemented on a loosely-coupled parallel computing architecture to pursue better performance. With the synchronous model, an average speedup of 6.2 was achieved by seven processors. On the other hand, the asynchronous model achieved a 7.6 speedup. The former obtained the speedups while maintaining equal or better quality of solutions than a serial GA. In addition to the above developed algorithms, preprocessing and postprocessing procedures were analyzed and developed to further enhance solution quality.