

## **400-MHz Frequency Counter: A Case Study in Semi-Synchronous Design**

Bernie New and Peter Alfke

Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124, USA, 1-408-559-7778  
bernie.new@xilinx.com, peter.alfke@xilinx.com

This poster describes the implementation of a 400-MHz frequency counter in an XC4002XL FPGA. In addition to speed, other objectives were low power and efficient resource utilization. These objectives were met using a semi-synchronous design technique where pairs of flip-flops operate as synchronous state machines that are cascaded asynchronously. XC4000XL CLBs each contain two flip-flops that share a common clock input. This common clock permits the pair of flip-flops to operate synchronously in spite of clock routing on local interconnect. A fully asynchronous design would waste half of the flip-flops since there would be no individual clock access. The outcome of the project was a full-featured frequency counter that operates at 400 MHz, consumes only 130 mW at the maximum input frequency, and occupies 56 CLBs, less than 90% of an XC4002XL.

## **Architecture Considerations for Mixed Signals FPGAs**

Luigi Carro

Departamento de Engenharia Eletrica - Universidade Federal do Rio Grande do Sul, Porto Alegre - RGS - Brasil  
carro@iee.ufrgs.br

This work studies some architectural characteristics of mixed signals FPGAs. The effect of programmability through the use of switches is analyzed, and it is shown that the result is a transfer function modification when real characteristics of the switches are assumed. Moreover, the paper proposes the use of externally linear, internally non-linear analog circuits, since this procedure could eliminate the error introduced by the switches. Using this approach, analog area is greatly reduced, and circuits can be built on top of completely digital technologies. Some experimental results in the analog and digital domain support the proposed approach to mixed circuits reprogrammability, being the basis for a mixed signal FPGA.

## **ATLANTIS - A Hybrid Approach Combining the Power of FPGA and RISC Processors Based on CompactPCI**

K.Kornmesser, T.Kuberka, A.Kugel, R.Manner, S.Ruhl, M.Sessler, H.Simmmler, H.Singpiel

University of Mannheim, Mannheim, Germany, email:holger.singpiel@ti.uni-mannheim.de  
[http://www-mp.informatik.uni-mannheim.de/groups/mass\\_par\\_1/parallelproc.html](http://www-mp.informatik.uni-mannheim.de/groups/mass_par_1/parallelproc.html)

ATLANTIS is the result of 5 years experience with large stand-alone and smaller PCI based FPGA processors. It realizes a hybrid system with a close coupling of RISC and FPGAs. Current applications are pattern recognition in high energy physics (HEP), image processing and n-body calculation. CompactPCI provides the basic communication mechanism. Dedicated FPGA boards for computing and I/O plus a private backplane for up to 1 GB/s data rate support flexibility and scalability. FPGAs with over 200k gates and 400 I/O pins are used. The I/O board with 2 FPGAs is configurable via mezzanines allowing up to 8 channels of 100MB/s. The computing board with 4 FPGAs has a fixed architecture but a flexible memory system by using submodules. For HEP e.g. a total of 40MB SRAM with 4.5 GB/s bandwidth is used. The high-performance hardware is complemented by CHDL, a FPGA design tool with special support for hybrid systems.

# **A Computational Intelligence Based Coarse-Grained Reconfigurable Element**

C. Hart Poskar, Peter J. Czezowski, and Robert D. McLeod

Department of Electrical & Computer Engineering  
University of Manitoba, Winnipeg, Manitoba, R3T 5V6 Canada  
{hpos | czezow | mcleod}@ee.umanitoba.ca

Computational intelligence techniques, such as neural networks and fuzzy systems, are increasingly being employed in real-world applications. Accordingly, much design effort is going into developing customizable modules to meet required hardware/software specifications. We have prototyped a fuzzy neuron as a coarse-grained reconfigurable element to be used in such a module. This module contains multiple reconfigurable fuzzy neurons, together with built-in memory, interface and fine grained reconfigurable logic to implement a fuzzy neural network in the fashion of a system-on-a-chip. The result is a dynamically reconfigurable computational intelligence based control/decision making system which features a parallel structure and in-situ learning.

## **Design issues in the development of a JAVA-processor for small embedded applications**

Hagen Ploog, Tino Rachui, Dirk Timmermann

Department of Electrical Engineering and Information Technology  
University of Rostock, Germany  
hp@e-technik.uni-rostock.de

This poster presents some design issues in the development of a JAVA-processor according SUN92s JavaCard 2.0 API for use in small embedded applications which could be realized with FPGAs. We employed this API because threads and garbage collection are not defined within this specification which leads to small area requirements. As our current solution is microcode-based we demonstrate that the footprint of the Java-processor can be reduced when using loosely coupled state machines (a microcode-sequencer and three slave state machines). Each slave state machine can HALT the microcode-sequencer while itself is still running. Furthermore we discuss some architecture details on implementing the stack on such systems as Java machine implementations are stack-based computer architectures.

## **Dynamically Programmable Cache Evaluation and Virtualization**

Mouna Nakkar, David G. Bently, John Harding, David Schwartz, Paul Franzon, and Thomas Conte

North Carolina State University

Dynamically Programmable Cache (DPC) is a novel architecture for embedded processors which offers high memory bandwidth and fast data accessibility. DPC processors merge reconfigurable arrays with data cache blocks at various cache levels to create multi-level reconfigurable machines. This will provide high memory bandwidth for FPGA cells and higher computation capacity per memory access. In addition, DPC machines implement a multi-context switching (Virtualization) concept. Virtualized DPC machines have two advantages: 1) they allow implementation of large subroutines with fewer FPGA cells, 2) and they can execute several operations in parallel resulting in faster execution time. The speedup improvements for the DPC machine are shown to be 5X faster than an Altera FLEX10K FPGA chip and 2X faster than a Sun Ultra1 SPARC station for three different algorithms (convolution, motion estimation, and run-length coding).

# **Efficient Support of Hardware Debugging through FPGA Physical Design Partitioning**

John Lach, William H. Mangione-Smith, Miodrag Potkonjak  
56-125B Engineering IV University of California Los Angeles, CA 90095  
Phone: (310) 794-1630 Fax: (310) 825-7928  
jlach@icsl.ucla.edu, billms@icsl.ucla.edu, miodrag@cs.ucla.edu

Emulation based on FPGA technology, for functional verification, is becoming a widespread practice for digital integrated circuit designers. One drawback to this debugging and testing technique is the lengthy time spent in the back-end computer-aided design (CAD) tools for each design iteration. Even for small and localized debugging changes large portions of the design are typically re-placed-and-routed. We have developed a more fine-grained approach that allows the CAD tools to re-place-and-route only the portions of the design affected by the debugging changes. This goal is achieved by partitioning the design at the physical level into independent components. Design changes are localized to the affected components, allowing more limited re-placement-and-routing. The result is a shorter time between emulation and debugging iterations, and thus a shorter time-to-market for the design. Experiments on two large designs and six smaller MCNC benchmarks quantify the reduced back-end CAD tool time.

## **Exploiting Early Partial Reconfiguration of Run-Time Reconfigurable FPGAs in Embedded Systems Design**

Byoungil Jeong, Sungjoo Yoo, Kiyoung Choi  
School of Electrical Engineering  
Seoul National University  
{bijeong,ysj,kchoi}@poppy.snu.ac.kr

With run-time reconfigurable FPGAs, we can perform partial reconfiguration, which allows reconfiguration of a part of an FPGA while the other part is executing some functional computation. The partial reconfiguration of a function can be performed earlier than the time when the function is really needed. Such early partial reconfiguration can hide the reconfiguration time overhead more effectively. In this paper we incorporate the technique of early partial reconfiguration of FPGA into hardware- software partitioning for FPGA-based embedded systems design. We model the problem as an integer linear programming. In the model, we consider overlapping functional computation and partial reconfiguration. Experimental results show that the proposed method achieves 38% performance gain and 64% hardware cost reduction on the average over the conventional static reconfiguration method. It also achieves 13% performance gain and 9% hardware cost reduction on the average over the lazy partial reconfiguration method.

## **Extra-Dimensional Island-Style FPGAs**

Herman Schmit  
Carnegie Mellon University  
herman@galant.ece.cmu.edu

This paper proposes modifications to standard island-style FPGAs that provide interconnect capable of scaling at the same rate as typical netlists, unlike traditionally tiled FPGAs. The proposal uses a logical third and fourth dimensions to create increasing wire density for increasing logic capacity. The additional dimensions are mapped to standard two-dimensional silicon. This innovation will increase the longevity of a given cell architecture, and reduce the cost of hardware, CAD tool and Intellectual Property (IP) redesign. In addition, extra-dimensional FPGA architectures provide a conceptual unification of standard FPGAs and time-multiplexed FPGAs, opening up new possibilities for prototyping of extremely large FPGAs and for IP transfer.

## **FPGA based computer vision camera**

A. Lecerf, F. Vachon, D. Ouellet, M. Arias-Estrada

Universite Laval

A computer vision camera prototype for real-time applications has been developed. The camera integrates a CMOS image sensor, an FPGA based coprocessing card, and an embedded PC for communication and control tasks. The system is targeted to computer vision tasks where low level processing and feature extraction can be implemented in the FPGA device. The FPGA coprocessing card integrates a medium size FPGA from Xilinx (XC4025E) with two memory banks, an ISA interface, and an image sensor interface. The camera can be accessed for architecture programming, data transfer, and control through an Ethernet link from a remote computer. The architecture of a classical multi-scale edge detection algorithm based on a Laplacian of Gaussian convolution has been developed to show the capabilities of the system. The camera can be used for hardware/software codesign, research on new computer vision architectures or educational purposes.

## **FPGA design experiences using the CSELT VIP(TM) Library**

E. Filippi, A. Montanaro, M. Paolini, M. Turolla

CSELT- Centro Studi e Laboratori Telecomunicazioni

Via Guglielmo Reiss Romoli 274-10148 Torino, Italy

{Enrica.Filippi, Achille.Montanaro, Maurizio.Paolini, Maura.Turolla}@cselt.it

<http://www.cselt.it/products/viplibrary/index.htm>

We describe the results of our design experiences using a fast prototyping methodology based on two key factors: FPGA technology and the CSELT Very High Level Intellectual Property (VIP) library. This library is composed of a set of high level modules, written in synthesizable RT-level VHDL and implementing a set of functions mappable both on ICs and FPGAs in different application areas. Modules are parametric in terms of both functionality and data width. The library includes blocks for Telecom, mainly in the ATM (Asynchronous Transfer Mode), and Multimedia areas, and general purpose modules. An ATM buffer/interface unit, a sorter for ATM traffic management, a shared buffer manager and a reconfigurable serial link interface have been implemented and tested on industrial applications both for internal and external customers. The use of the VIP(TM) library and FPGA devices proved to be very effective in terms of design time, with highly satisfactory performance.

## **FPGA-Targeted Development System for Embedded Applications**

V. Sklyarov, J. Fonseca, R. Monteiro, A. Oliveira, A. Melo, N. Lau, K. Kondratjuk, I. Skliarova, P. Neves, A. Ferrari

Department of Electronics and Telecommunications, Aveiro University (Portugal)

skl@inesca.pt, {jaf, ricardo, arnaldo, andreaia, lau, const, iouliia, neves}@ua.pt, ferrari@inesca.pt

This paper considers approaches to the design and implementation of embedded systems using XC6200 FPGAs. The methods that are introduced enable the synthesis of circuits that are modifiable and extensible, and that provide a virtual function capability. The accepted behavioral specification supports modularity and hierarchy. The developed design tools allow translating this specification into dynamically modifiable control circuits. A method based on reconfigurable cores for rapid design of reconfigurable virtual datapath was suggested. A stand-alone board using one XC6216 FPGA was designed and two other solutions, currently under development, were discussed. They can be used as virtual embedded controllers. An integrated design environment (IDELS) has been developed to provide specification, synthesis, simulation, testing, debugging, and implementation of the circuits in hardware. The software has been developed using Visual C++ and allows access to both stand-alone and built-in PC boards.

# **Hardware/Software Partitioning between Microprocessor and Reconfigurable Hardware**

M. Anand, Sanjiv Kapoor and M. Balakrishnan.

Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi 110016.

This paper presents a theoretical study of the partitioning problem for a co-design environment for speeding up compute intensive applications. The environment has a uniprocessor host with a reconfigurable target platform comprising of FPGAs and a library of functions pre-synthesized for hardware or software implementation. The partitioning problem is thus reduced to choosing between hardware and software implementations for all such function occurrences. The optimal way of performing partitioning in this co-design environment is presented for two cases. In the static case we allow for one time configuration of the FPGAs at the start of execution. A greedy strategy is shown to be optimal in this case. The dynamic case allows for run-time reconfiguration of the FPGAs determined at compile time. Here, for straight line programs, we show that the problem of partitioning reduces to the problem of finding a min cost flow of value  $M$  (No. of FPGAs) in a network. For programs with branches, we describe an efficient dynamic programming solution. Some implementation results are also presented.

## **High-performance Low-cost Implementation of Two-dimensional DCT Processor on FPGA**

L. Naviner, J-L. Danger, C. Laurent

Ecole Nationale Supérieure des Telecommunications, 46, rue Barrault, 75637, Paris CEDEX 13 – France  
{lirida.naviner, danger, laurent}@enst.fr, <http://www.com.enst.fr>

This paper presents a high-performance low-cost implementation for two-dimensional discrete cosine transform processor. This processor is an ENST contribution for the AC078 ATLANTIC project and is a part of a complete MPEG2 encoder prototype. The little number of pieces imposes a FPGA approach for implementation, but accuracy requirements impose errors to be up to 16 times inferior to the ones in IEEE specification. So, an efficient architecture based on distributed arithmetic is used to reduce the hardware amount and enhance the speed performance. Complete use of the logic cell's is obtained with various architectural optimisations. These optimisations include pseudo multiplexing, special encoding and resource sharing for multiplications, additions and accumulations of partial inner products. 11-bits pixels in input are processed, generating 14-bits coefficients in output. System is built just on a Flex10K40 circuit of Altera, works at 36 MHz, and guarantees real-time processing for 18 MHz input pixel rate.

## **High-Performance 2-D FPGA DCTs using Polynomial Transforms**

Dr. Chris Dick

Xilinx Inc., 2100 Logic Drive, San Jose, CA 95124, [chrisd@xilinx.com](mailto:chrisd@xilinx.com)

This presentation investigates two options for the field programmable gate array (FPGA) implementation of a very high-performance 2-D discrete cosine transform (DCT) processor for real-time applications. The first architecture exploits the transform separability and uses a row-column decomposition. The row and column processors are realized using distributed arithmetic (DA) techniques. The second approach uses a naturally 2-D method based on polynomial transforms. The paper provides an overview of the DCT calculation using DA methods and describes the FPGA implementation. A tutorial overview of a computationally efficient method for computing 2-D DCTs using polynomial transforms is presented. A detailed analysis of the datapath for this approach using an 8 x 8 data-set is given. Comparisons are made that show the polynomial transform approach to require 67% of the logic resources of a DA processor for equal throughputs. The polynomial transform approach is also shown to scale better with increasing block size than the DA approach.

# High Speed Calculation of Cyclic Redundancy Codes

John McCluskey

Lucent Technologies, JMcCluskey@lucent.com , <http://www.lucent.ca/fpga>

This paper describes a high speed VHDL implementation of a polynomial division algorithm suitable for FPGA implementation. Examples & benchmarks will be shown for an ITU-T I.363 compliant CRC-32 implementation with data throughput in excess of 3 Gbits/sec bits, based on processing 32 bit words. The VHDL code is parameterized, accepting a static polynomial of arbitrary length, and an input data word of arbitrary width. A parity matrix is calculated during elaboration of the VHDL that specifies the XOR gate size and connections required for implementation of the polynomial divider circuit. URL: <http://www.mlink.net/~jqm/crc.pdf>

## Hierarchical Placement Directives for Parametric IP Blocks

James Hwang, Cameron Patterson, and Sujoy Mitra

Xilinx, San Jose, CA

Today's FPGAs are no longer used simply for glue logic. They possess sufficient gate capacity and performance to implement intellectual property (IP) blocks and other complex systems consisting of data paths, control logic, I/O, and memories. However, high performance circuits often require hand-crafted layout and hierarchical placement constraints. Determining such constraints for complex floorplans and parametric circuits is quite difficult and error-prone. In this paper we describe a new approach to hierarchical layout designed specifically for parametric IP blocks. Layouts are specified with composable templates that define geometric constraints in abstract coordinates, with a back-end that computes actual device locations "on-demand." These templates are particularly well-suited to structured circuits, and facilitate the composition and reuse of hand-crafted modules. They can often hide extraneous technology-specific architectural details from the specification, which simplifies both design and maintenance. We have implemented the placement directives and framework, supporting VHDL flows as well as Java-based module generators.

## Implementing an Artificial CPG using Fine-Grain FPGAs

Zhijun Yang and Felipe M.G. Franca

Universidade Federal do Rio de Janeiro, COPPE - Programa de Engenharia de Sistemas e Computacao, CEP: 21945-970, Caixa Postal 68511, Rio de Janeiro, RJ, Brazil

{felipe, yang}@cos.ufrj.br, [www.cos.ufrj.br/~felipe](http://www.cos.ufrj.br/~felipe), [www.cos.ufrj.br/~yang](http://www.cos.ufrj.br/~yang)

A novel, model-independent, approach for the retrieval of coupled neural oscillations observed in biological Central Pattern Generators (CPGs) during the control of walking is presented and its implementation as pseudo auto-oscillatory digital circuits in fine-grain FPGAs illustrated. Based on Scheduling by Multiple Edge Reversal (SMER), a very simple and powerful distributed synchronizer, various biological building blocks can be configured for the production of complicated rhythmic patterns and a methodology is provided for the organization, optimization and simulation of the target artificial CPGs. The whole procedure of mimicking the neurolocomotor mechanism of an hexapodal insect, from model construction to coordinated movement simulation with FPGAs, indicates that our methodology can be general enough to deal with multiple legged animals provided that their locomotive CPG architecture can be described topologically.

# **A Method for Implementing Fractal Image Compression on Reconfigurable Architecture**

Akihiro Matsuura, Hidehisa Nagano, Akira Nagoya

NTT Communication Science Laboratories, 2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, JAPAN  
{matsuura, nagano, nagoya}@cslab.kecl.ntt.co.jp

In this paper, we focus on the acceleration of fractal image compression and present a method for implementing its encoding part on reconfigurable hardware such as FPGAs. In the encoding part, distance computations between image blocks form the main time-consuming task. By employing reconfigurable computing and constructing processing element (PE) networks where each PE is devoted to a specified image block, each multiplication in the distance computations can be regarded as a multiplication of a single constant. This implementation allows us to use shifters and adders instead of variable-by-variable multipliers. The effect is analyzed for two number representations, 2's complement and the CSD format. Using benchmark images, we show that the number of additions and subtractions is reduced down to 40-50% compared with the original multipliers-based algorithm. We also show that this reduction enables us to reduce the size of PEs on Altera FPGAs to a similar degree.

## **Module Generation of High Performance FPGA-Based Multipliers**

Kun-Ming Ho\* and Allen C.-H. Wu\*\*

\*Avant! Corporation, 46871 Bayside Parkway, Fremont, CA 94538, kunming@avantcorp.com

\*\*Department of Computer Science, Tsing Hua University, Hsinchu, Taiwan, 300, Republic of China,  
chunghaw@cs.nthu.edu.tw

In this paper, we present a module generator MG\_fpga for high-performance FPGA-based multipliers. MG\_fpga is able to generate array multipliers with arbitrary bit-widths and pipelined stages. By considering path delays, routability, shape constraints during the placement stage, MG\_fpga generates high-speed multiplier modules with various shapes. Experimental results demonstrate that our multiplier generator can produce multiplier modules on an average of 20%-48% faster than that generated using a conventional module generation method.

## **Partitioning Large Designs by Filling FPGA Devices with Hierarchy Blocks**

Helena Krupnova, Gabriele Saucier

CSI/Institut National Polytechnique de Grenoble, 46, Avenue Felix Viallet 38031 Grenoble cedex, FRANCE  
{bogushev, saucier}@imag.fr

The paper addresses the problem of design partitioning into multiple FPGA devices. Most published work in FPGA partitioning was dedicated to developing the automatic partitioning methods. But the industrial experience shows that the designers were never satisfied by the full automatic partitioning results. As the design size grows, automatic partitioning takes longer CPU times and produces poor results. The present paper proposes an algorithm which may be integrated into the mixed manual/automatic partitioning framework. The hierarchy nodes of the design are selected one by one and assigned to a defined set of FPGA devices. The automatic partitioning algorithm is called to split a big node among the subset of devices taking into account previous assignments. Experimental results show that the proposed approach works well for big industrial circuits and gives better results than full automatic partitioning.

# **Practical applications of recursive VHDL components in FPGA synthesis**

John McCluskey

Lucent Technologies, JMcCluskey@lucent.com , <http://www.lucent.ca/fpga>

This paper explores and exposes practical applications of recursive structures for synthesizing high performance circuit designs in Field Programmable Gate Arrays. Recent improvements in synthesis tool compliance to the VHDL-93 standard now permit designers to write a VHDL component that actually instantiates itself. This capability can be used to create scalable circuits that would be difficult to express in a flattened topology. Examples of circuits that can benefit from this approach are wide AND, OR, XOR gates, multiplexers, demultiplexers, multipliers, ROMs, Walsh function generators, and other circuits with M-ary tree topologies. The implementation of a recursive circuit also lends itself well to FPGA's, since proper sizing of the recursive components leads to circuits easily pipelined by the addition of registers at the outputs of the recursively instantiated component. URL: <http://www.mlink.net/~jqm/recurse.pdf>

## **Prototyping board and development environment for rapid prototyping of real time and regular digital signal processing applications**

Philippe Soulard

Universite de Bretagne Occidentale

In this paper, we present a board and a programming environment dedicated to real time prototyping of digital signal processing applications. The board is based on Field Programmable Gates Arrays and the environment is based on Java and VHDL. The board is not dedicated to an application thanks to FIFO memories, and the FPGAs can be partially and dynamically programmable or not. The environment uses a specification in Java, but some methods can be written in VHDL in order to be synthesized on FPGAs. This allows an exploration in a solution space where specifications are described in an appropriate language. The parts that are never critical are only described in Java. The parts that can be sometimes critical are described in Java and VHDL, and we verify that they are equivalent. The parts that are allways critical are only described in VHDL.

## **Run-Time Parameterizable Cores**

Steve Guccione and Delon Levi

Xilinx

As FPGAs have increased in density, the demand for predefined intellectual property has risen. Rather than re-invent commonly used circuitry, libraries of standard parts have become available from a variety of sources. Currently, all of these offerings are based on the standard ASIC design flow and are used to produce fixed designs. This paper discusses Run-Time Parameterizable or RPT Cores which are an extension of the traditional static core model. Written in the Java (tm) programming language, RTP Cores are created at run-time and may be used to dynamically modify existing circuitry. In addition to providing support for run-time reconfigurable computing, RTP Cores permit run-time parameterization of designs. This adds flexibility and portablilty unavailable in existing design environments.



# Self-checking logic design for LUT-based FPGAs

P. K. Lala\*, A. L. Burrell\*\*

\*Department of Electrical Engineering, University of South Florida, 4202 E. Fowler Avenue, Tampa, FL 33620-5350,  
lala@eng.usf.edu

\*\*IBM Corp., Research Triangle Park, NC 27709

A technique for designing self-checking logic for implementation in Xilinx FPGAs is proposed. Self-checking circuits can detect permanent and transient faults during normal operation. The technique uses two types of cells; a functional cell and a checker cell. The functional cell is composed of one CLB. If a fault occurs within a CLB, it produces identical outputs. The checker cell produces complementary outputs when it receives sets of complementary inputs. Each checker cell receives outputs of two separate intermediate functional cells. Therefore if a fault occurs within an intermediate functional cell or is propagated by it, the checker cell connected to it produces identical outputs. Thus, this technique allows on-line detection of faults within the combinational functional block of a CLB, and on the interconnect lines connecting the functional blocks. Other faults in the CLBs e.g. those in the muxes and flip-flops may not necessarily be detected.

## Special Arithmetic Operations on FPGAs

Matti Tommiska

Helsinki University of Technology, Laboratory of Signal Processing and Computer Technology, Phone +358 9 451 2477 or +358 40 541 0981, Fax +358 9 460 224, Matti.Tommiska@hut.fi  
<http://wooster.hut.fi/~matti/> (In Finnish), [http://wooster.hut.fi/~matti/index\\_en.html](http://wooster.hut.fi/~matti/index_en.html) (In English)

The implementation of the exponent function, the logarithm function and the square root function on Altera's FLEX10K devices is described. The implementation of special arithmetic functions differs in many ways from traditional design tasks, since the designers must be well versed in both mathematics and the characteristic features of the targeted devices. The mathematical algorithms of the three arithmetic functions are described, and their applicability to FPGA-based implementation is demonstrated. The use of the internal memory blocks as a lookup table greatly reduces the required area in iterative algorithms, and this is utilized in the implementation of the exponent and logarithm functions. It is shown, that the implementation of the exponent and logarithm functions have many features in common, and that together with the square root functions, all three can be efficiently implemented without the need to resort to multiplications or divisions.

## Throughput Optimization with Design Space Exploration during Partitioning for Multi-FPGA Architectures

Vinoo Srinivasan, Ranga Vemuri

{vsriniva,ranga}@ececs.uc.edu

We have developed a heuristic for maximizing the throughput of a task graph that is pre-partitioned for a multi-FPGA reconfigurable architecture. The nodes in the graph represent behavioral task segments and the edges denote data dependencies. Given, for each task, is a set of implementation options corresponding to the various area-time trade-off points in its design space. A fast and efficient heuristic selects an implementation for each task such that a near-optimal execution time for the graph can be achieved, while satisfying the resource constraints imposed by the FPGAs. Also part of our approach is a area estimation heuristic that accounts for efficient sharing of resources between tasks that are mutually time exclusive. We compare our approach with a genetic algorithm, a general-purpose combinational search technique, that solves the same problem. The results show the effectiveness of our methodology.

# **Towards Adaptable Hierarchical Placement for FPGAs**

Florent de Dinechin\*, Wayne Luk, Steve McKeever\*\*

\*INRIA, France, fdedinec@ens-lyon.fr

\*\*Imperial College, London, UK, wl@doc.ic.ac.uk, swm2@doc.ic.ac.uk

This research addresses layout issues in FPGA synthesis. We describe a framework for exploiting placement information which reduces compilation time while improving design quality. The framework is general enough to allow designs with generic placement information to adapt to a range of FPGAs with different granularities and routing resources. The key novelty of our approach is the use of placement constraints expressed as polynomial expressions. This method enables constraints to be described and manipulated in a generic way, independent of the size of a circuit. The approach supports various static checks and provides informative user feedbacks on how designs can be improved. A hierarchical resolution engine has been developed to automate the solution of the placement constraint expressions. Prototype implementations of this approach are presented for Xilinx 6200 and Xilinx 4000 devices. Their effectiveness is illustrated by a bit-serial complex multiplier.

# **Unified Access to Heterogeneous Module Generators**

Andreas Koch

UC Berkeley (ICSI), 1947 Center St., #600, Berkeley, CA 94704-1198

akoch@icsi.berkeley.edu

To counter coarse logic-block granularity and limited routing resources, high-performance design flows for FPGAs often rely on module generators to implement fast sub-circuits. However, the very flexibility of current generator systems makes their automatic use by synthesis and floorplanning steps difficult. We present FLAME, the Flexible API for Module-based Environments, as a solution to these problems. FLAME defines a common model for expressing generator capabilities and module characteristics to module consumers (such as synthesis or floorplanning tools), textual and binary representations for FLAME data, and an API for exchanging FLAME expressions between programs. The API gives consumers access to the module information via a query/reply scheme supporting incremental design refinement. This dynamic approach avoids the explicit enumeration of all design alternatives in static library files, which becomes infeasible for flexible generators. FLAME-compliant module generators and consumers enable the efficient vendor-independent combination of generator-based IP and traditional design tools.

# **Universal Switch Blocks for Three-Dimensional FPGA Design**

Guang-Ming Wu, Michael Shyu, and Yao-Wen Chang

Department of Computer and Information Science, National Chiao Tung University, Hsinchu 300, Taiwan  
{gmwu, michael, ywchang}@cis.nctu.edu.tw, <http://www.cis.nctu.edu.tw/~ywchang>

In this paper, we consider the switch-block design problem for three-dimensional FPGAs. A three-dimensional switch block  $M$  with  $W$  terminals on each face is said to be universal if every set of nets satisfying the dimension constraint (i.e., the number of nets on each face of  $M$  is at most  $W$ ) is simultaneously routable through  $M$ . In this paper, we present a class of universal switch blocks for three-dimensional FPGAs. Each of our switch blocks has  $15W$  switches and switch-block flexibility 5 (i.e.,  $F_s = 5$ ). We prove that no switch block with less than  $15W$  switches can be universal. We also compare our switch blocks with others of the topology associated with those used in the Xilinx XC4000 FPGAs. Experimental results demonstrate that our universal switch blocks improve routability at the chip level. Further, the decomposition property of a universal switch block provides a key insight into its layout implementation with a smaller silicon area.

# **Why a CAD-verified FPGA makes routing so simple and fast!**

**-- A result of Co-designing FPGAs and CAD algorithms --**

Takahiro Murooka, Atsushi Takahara and Toshiaki Miyazaki

NTT Optical Network Systems Laboratories  
{murooka, taka, miyazaki}@exa.onlab.ntt.co.jp

Using a routing algorithm as an example, we show, how CAD algorithms can be simplified by the CAD-verified FPGA. The FPGA is designed with emphasis on its routing resource architecture, especially its switch module structure. This makes the routing algorithm simple; it is composed of a detail-routing strategy without global one. The experimental results indicate, the tool routes given netlists almost one-hundred-times faster than a conventional one. The results also have better quality. In the poster session, we'll explain the property of our CAD-verified FPGA and report, through comparison with a conventional routing method, how the routing strategy and algorithms are simplified using the property of the target FPGA in detail.

## **THE X-MatchLITE FPGA-BASED DATA COMPRESSOR.**

Jose Luis Nunez, Claudia Feregrino, Stephen Bateman\*, Simon Jones

Electronic Systems Design Group, Loughborough University, Loughborough, Leicestershire. LE11 3TU. England.  
<http://www.lboro.ac.uk/departments/el/research/sys/>, {J.L.Nunez-yanez,C.Feregrino-uribe,S.R.Jones}@lboro.ac.uk

\*Vice President of Engineering, GateField Corporation, 47100 Bayside Parkway, Fremont, CA  
sbateman@gatefield.com

This paper introduces a hardware amenable algorithm for lossless data compression and a highly integrable architecture which enables Gbit/s compression using contemporary ASIC technology. An FPGA prototype of the architecture is presented. A comparison between this prototype and the full version of the system is made together with the details of the engineering decisions needed to successfully realise an ASIC compressor in FPGA technology. The use of a GateField GF250F100 ProASIC has proven to be well suited to a design like the X-Match compressor yielding a high utilisation ratio. The reason is that in our design most of the sequential logic (dictionary) and combinatorial logic (coding and decoding functions) are clearly separated on the silicon making good use of fine granularity devices like the ProASIC's. The FPGA prototype while still offering good compression ratio and speed shows that a full implementation of X-Match would be a very useful data compressor.