

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 1: SPRING 1996, pp. 26-35(Editorial Information: Address questions or comments about this article to the author at State University of Informatics and Radioelectronics of Belarus, Computer Systems Department, P.Brovki 6, 220027 Minsk, Belarus; prihozhy% csd.bsuir.minsk.by@brc.minsk.by.)

Net Scheduling in High-Level Synthesis

ANATOLY PRIHOZHY

State University of Informatics and Radioelectronics of Belarus

A new net scheduling and allocation model generates net schedules that minimize either execution time or resources. The author tested the model within a VHDL-based high-level synthesis system called Ahiles.

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 1: SPRING 1996, pp. 48-57(Editorial Information: Address questions or comments about this article to the author at Tallinn Technical University, Computer Engineering Dept., Ehitajate tee 5, EE-0026 Tallinn, Estonia; raiub@pld.ttu.ee.)

Test Synthesis with Alternative Graphs

RAIMUND UBAR

Tallinn Technical University

Alternative graphs provide an efficient, uniform model describing the structure, functions, and faults in a wide class of digital circuits and for different representation levels. For test pattern generation, they provide a general theoretical basis for combining high-level approaches, symbolic techniques based on binary decision diagrams, and traditional topological algorithms.

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 1: SPRING 1996, pp. 70-78(Editorial Information: Send questions or comments about this article to Peter Schneider, Room 2912, Institute of Electronic Design Automation, Technical University of Munich, 80290 Munich, Germany; Schneider@regent.e-technik.tumuenchen.de.)

Fast Power Estimation of Large Circuits

PETER H. SCHNEIDER

Siemens AG Technical University of Munich

ULF SCHLICHTMANN

Siemens AG

BERND WURTH

Siemens AG Technical University of Munich

Our new technique for estimating transition probabilities of internal signals in combinational circuits uses Markov chains and reconvergence regions. To efficiently implement the computation, we use ROBDDs (reduced, ordered binary decision diagrams). Accounting for temporal dependence of signals, multiple concurrent transitions, and mutual dependence of internal signals, the technique provides an exact computation for small circuits and an approximate estimate for large circuits. Experimental results show the estimation technique is fast with only small inaccuracies.

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 2: SUMMER 1996, pp. 42-57(Editorial Information: Direct questions concerning this article to Stephen Brown, Dept. of Electrical and Computer Engineering, Univ. of Toronto, 10 Kings College Rd., Toronto, ONT, Canada M5S 3G4; brown@eecg.toronto.edu.)

FPGA and CPLD Architectures: A Tutorial

STEPHEN BROWN

University of Toronto

JONATHAN ROSE

University of Toronto

This tutorial surveys commercially available, high-capacity field-programmable devices. The authors describe the three main categories of FPDs: simple and complex programmable logic devices, and field-programmable gate arrays. They then give architectural details of the most important chips and example applications of each type of device.

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 2: SUMMER 1996, pp. 58-70(Editorial Information: Address questions or comments about this article to Raul San Martin, Nortel Technology, PO Box 3511, Station C, Ottawa, ONT K1Y4H7, Canada; raulsm@nortel.ca.)

Optimizing Power in ASIC Behavioral Synthesis

RAUL SAN MARTIN

Nortel Technology

JOHN P. KNIGHT

Carleton University

Attacking power consumption at the behavioral level exploits an application's inherent parallelism to maintain performance while compensating for slower, less power-hungry operators. The authors' method and tool optimize and evaluate the effects of power-saving strategies on performance and silicon area.

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 2: SUMMER 1996, pp. 72-82(Editorial Information: Direct questions concerning this article to Paul Landman, Integrated Systems Laboratory, Texas Instruments Inc., 13510 North Central Expressway, MS 446, Dallas, TX 75243; landman@hc.ti.com.)

An Integrated CAD Environment for Low-Power Design

PAUL LANDMAN

Texas Instruments

RENU MEHRA

University of California, Berkeley

JAN M. RABAEY

University of California, Berkeley

This CAD environment supports a high-level approach to power reduction, emphasizing optimizations at the algorithm and architecture levels of abstraction. An integrated set of analysis and optimization tools spans the design hierarchy, allowing the designer to make a systematic, top-down exploration and refinement of solutions in the area-time-power design space. In a case study a low-power implementation of a digital bandpass filter the CAD environment and tools yield more than an order of magnitude savings in power.

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 3: FALL 1996, pp. 12-22(Editorial Information: Address questions or comments about this article to Vijay K. Madiseti, School of Electrical and Computer Engineering, Georgia Institute of Technology, 777 Atlantic Drive, Atlanta, GA 30332-0250; vkm@ee.gatech.edu.)

Rapid Digital System Prototyping: Current Practice, Future Challenges

VIJAY K. MADISETTI

Georgia Institute of Technology VP Technologies

A top-down methodology that emphasizes modular, upgradable designs solves some problems with the current design process for application-specific signal processors. Virtual prototyping, model year architectures, and reuse-based design form the pillars of this new approach, which promises to raise industrial design productivity and competitiveness.

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 3: FALL 1996, pp. 43-53(Editorial Information: Direct questions concerning this article to the author in care of Stacey O'Malley, MS RC2/2640, TRW Avionics Systems Division, One Rancho Carmel, San Diego, CA 92128.)

Hardware-Software Codesign Using Processor Synthesis

CAROLYN KUTTNER

TRW Space and Electronics Group

A new tool aids concurrent hardware-software development of processors for embedded systems. Pilot programs using ProcSyn achieved significant cost and time savings and, in some cases, improved the quality of the final product.

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 3: FALL 1996, pp. 54-65(Editorial Information: Direct questions concerning this article to Vijay K. Madiseti, Dept. of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250; vkm@ee.gatech.edu.)

Conceptual Prototyping of Scalable Embedded DSP Systems

LAN-RONG DUNG

Georgia Institute of Technology

VIJAY K. MADISETTI

Georgia Institute of Technology

This systematic process uses model-based architectural synthesis and verification to ensure that early stages of the design are efficient, economical, and meet user requirements. Using a scalable, plug-and-play, model year methodology, designers can conceptually prototype complex, embedded digital systems in weeks instead of months.

---

IEEE DESIGN & TEST OF COMPUTERS 0740/7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 3: FALL 1996, pp. 66-78(Editorial Information: Direct questions concerning this article to Sandi Habinc, Microelectronics and Technology Section (WSM), European Space Research and Technology Centre, Postbus 299, NL-2200 AG Noordwijk, The Netherlands; sandi@ws.estec.esa.nl.)

Using VHDL for Board Level Simulation

SANDI HABINC

European Space Agency

PETER SINANDER

European Space Agency

Prototyping is necessary for successful development of printed circuit boards built with complex components such as microprocessors, ASICs, and ASSPs. The European Space Agency uses VHDL models for board level simulation, optimizing such models for high functional accuracy and simulation performance

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 3: FALL 1996, pp. 88-96(Editorial Information: Direct questions about this article to Ramesh Karri, Univ. of Massachusetts at Amherst, Dept. of Electrical and Computer Engineering, Amherst, MA 01003; karri@india.ecs.umass.edu.)

Computer-Aided Design of Fault-Tolerant VLSI Systems

RAMESH KARRI

University of Massachusetts at Amherst

KARIN HOGSTEDT

University of California, San Diego

ALEX ORAILOGLU

University of California, San Diego

The authors present a flexible methodology for compiling an algorithmic description into an equivalent fault-tolerant VLSI circuit and a CAD framework embodying this methodology. Experimental designs illustrate and validate algorithms for automated synthesis of ICs featuring either self-recovery capability or enhanced reliability

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 4: WINTER 1996, pp. 16-23(Editorial Information: Address questions or comments about this article to Stephen Brown, Department of Electrical and Computer Engineering, University of Toronto, 10 Kings College Road, Toronto, Ontario, Canada, M5S 3G4; brown@eecg.toronto.edu.)

Minimizing FPGA Interconnect Delays

STEPHEN BROWN

University of Toronto

MUHAMMAD KHELLAH

University of Toronto

ZVONKO VRANESIC

University of Toronto

Optimizing FPGA routing architectures for speed performance also involves improving the CAD tools for mapping circuits. Although their results are sensitive to the tools used, the authors draw several basic conclusions about both FPGA routing architectures and CAD tools.

---

IEEE DESIGN & TEST OF COMPUTERS 0740-7475/96/\$05.00 © 1996 IEEE

Vol. 13, No. 4: WINTER 1996, pp. 24-33(Editorial Information: Direct questions concerning this article to Todd A. DeLong, Dept. of Electrical Engineering, Thornton Hall, University of Virginia, Charlottesville, VA 22903-2442; tad2x@virginia.edu.)

A Fault Injection Technique for VHDL Behavioral-Level Models

TODD A. DELONG

University of Virginia

BARRY W. JOHNSON

University of Virginia

JOSEPH A. PROFETA III

Union Switch and Signal, Inc.

Designers are realizing the advantages of performing fault injection early, using simulation to inject faults into a model of the design rather than the actual system. The authors describe their technique for injecting faults into a system's VHDL behavioral-level model. To demonstrate the technique, they evaluate an embedded control system providing fail-safe operation in the railway industry.

---

COMPUTER 0018-9162/96/\$5.00 © 1996 IEEE

Vol. 29, No. 4: APRIL 1996, pp. 44-53(Editorial Information: Readers can contact Vemuri at the University of Cincinnati, P.O. Box 210030, Department of Electrical and Computer Engineering and Computer Science, Cincinnati, OH 45221-0030, e-mail ranga.vemuri@uc.edu. Readers can contact Mandayam at Motorola, 1303 E. Algonquin Rd., Annex E, Schaumburg, IL 60173, e-mail arm010@email.mot.com. Readers can contact Meduri at LSI Logic, e-mail vmeduri@lsil.com.)(Editorial Information: Readers can obtain further information about the authors' PDL research, including the PDL software, from the Web at <http://www.ece.uc.edu/~ddel/pdl.html>.)

Performance Modeling Using PDL

Ranga Vemuri

University of Cincinnati

Ram Mandayam

Motorola

Vijay Meduri

LSI Logic

Designers can write generic performance models in PDL. For individual designs, these models can be compiled into specific, executable models that yield valuable performance data.

---

IEEE TRANSACTIONS ON COMPUTERS 0018-9340/96\$05.00 © 1996 IEEE

Vol. 45, No. 1: JANUARY 1996, pp. 74-87

Retiming-Based Partial Scan\*

Dimitrios Kagaris, Member, IEEE

Spyros Tragoudas, Member, IEEE

Abstract<sup>3</sup>A generally effective criterion for the selection of flip-flops in the partial scan problem for sequential circuit testability is to select flip-flops that break the cyclic structure of the circuit and reduce its sequential depth. The selection of flip-flops may also be subject to a prescribed bound on the clock period of the modified circuit (timing-driven partial scan). In this paper we propose two techniques (for non-timing-driven and timing-driven partial scan) which address the above criterion based on a transformation of sequential circuits known as retiming. For non-timing-driven partial scan, we employ retiming to rearrange the flip-flops of the circuit, so that its functionality is preserved, while the number of flip-flops that are needed to break all cycles and bound the sequential depth is significantly reduced. For timing-driven partial scan, we propose a retiming-based technique that reduces the overall area overhead required to achieve the clock period bound. Experimental results on the ISCAS'89 circuits show the benefit of our approach in both timing-driven and non-timing-driven partial scan.

---

IEEE TRANSACTIONS ON COMPUTERS 0018-9340/96 \$05.00 © 1996 IEEE

Vol. 45, No. 2: FEBRUARY 1996, pp. 131-142

Automatic Synthesis of Self-Recovering VLSI Systems\*

Alex Orailoglu

Ramesh Karri

Abstract<sup>4</sup>In this paper, we will describe an integrated system for synthesizing self-recovering microarchitectures called  $\{\text{cal SYNCERE}\}$ . In the  $\{\text{cal SYNCERE}\}$  model for self-recovery, transient faults are detected using duplication and comparison, while recovery from transient faults is accomplished via checkpointing and rollback.

$\mathcal{SYNCERE}$  initially inserts checkpoints subject to designer specified recovery time constraints. Subsequently,  $\mathcal{SYNCERE}$  incorporates detection constraints by ensuring that two copies of the computation are executed on disjoint hardware. Towards ameliorating the dedicated hardware required for the original and duplicate computations,  $\mathcal{SYNCERE}$  imposes intercopy hardware disjointness at a sub-computation level instead of at the overall computation level. The overhead is further moderated by restructuring the pliable input representation of the computation.  $\mathcal{SYNCERE}$  has successfully derived numerous self-recovering microarchitectures. Towards validating the methodology for designing fault-tolerant VLSI ICs, we carried out a physical design of a self-recovering 16-point FIR filter.

---

IEEE TRANSACTIONS ON COMPUTERS 0018-9340/96\$05.00 © 1996 IEEE

Vol. 45, No. 3: MARCH 1996, pp. 294-306

A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach\*

Vojin G. Oklobdzija, Fellow, IEEE

David Villeger

Simon S. Liu

Abstract This paper presents a method and an algorithm for generation of a parallel multiplier, which is optimized for speed. This method is applicable to any multiplier size and adaptable to any technology for which speed parameters are known. Most importantly, it is easy to incorporate this method in silicon compilation or logic synthesis tools. The parallel multiplier produced by the proposed method outperforms other schemes used for comparison in our experiment. It uses the minimal number of cells in the partial product reduction tree. These findings are tested on design examples simulated in 1m CMOS ASIC technology.

---

IEEE TRANSACTIONS ON COMPUTERS 0018-9340/96\$05.00 ©1996 IEEE

Vol. 45, No. 3: MARCH 1996, pp. 307-318

The Differential CORDIC Algorithm: Constant Scale Factor Redundant Implementation without Correcting Iterations\*

Herbert Dawid, Student Member, IEEE

Heinrich Meyr, Fellow, IEEE

Abstract The CORDIC algorithm is a well-known iterative method for the efficient computation of vector rotations, and trigonometric and hyperbolic functions. Basically, CORDIC performs a vector rotation which is not a perfect rotation, since the vector is also scaled by a constant factor. This scaling has to be compensated for following the CORDIC iteration.

Since CORDIC implementations using conventional number systems are relatively slow, current research has focused on solutions employing redundant number systems which make a much faster implementation possible. The problem with these methods is that either the scale factor becomes variable, making additional operations necessary to compensate for the scaling, or additional iterations are necessary compared to the original algorithm.

In contrast we developed transformations of the usual CORDIC algorithm which result in a constant scale factor redundant implementation without additional operations. The resulting "Differential CORDIC Algorithm" (DCORDIC) makes use of on-line (most significant digit first redundant) computation. We derive parallel architectures for the radix-2 redundant number systems and present some implementation results based on logic synthesis of VHDL descriptions produced by a DCORDIC VHDL generator. We finally prove that, due to the lack of additional operations, DCORDIC compares favorably with the previously known redundant methods in terms of latency and computational complexity.

Index Terms CORDIC, carry save, signed digit, redundant number systems, radix-2, VLSI architecture, computer arithmetic.

---

IEEE TRANSACTIONS ON COMPUTERS 0018-9340/96\$05.00 © 1996 IEEE

Vol. 45, No. 4: APRIL 1996, pp. 425-438

Algorithm-Based Fault Tolerant Synthesis for Linear Operations\*

Jan-Lung Sung

G. Robert Redinbo

Abstract High-level synthesis is becoming more important in practical design environments to meet new system requirements and, increasingly, fault tolerance is one especially because of high-speed and low power demands. This paper explores several basic aspects of low-cost fault tolerant synthesis for practical linear systems. It deals with practical design constraints that require basic operations to be only performed by a limited processing resources and

do not normally assign each operation a separate processing resource. Two core issues, partitioning and allocation, for fault tolerant synthesis are examined. Results demonstrate a high-level abstraction and framework for fault tolerant synthesis which is almost totally independent of the physical hardware implementation. Issues in designing 1-fault detectable FFT system are considered in detail to illustrate the significance and effects of fault tolerant synthesis schemes. Our ultimate goal is to incorporate these techniques in future automated design tools so that fault tolerance features can be part of the design options.

Index Terms: 1-fault detectable (1-FD) system, algorithm-based fault tolerant (ABFT) synthesis, data flow graph (DFG), fast Fourier transform (FFT), gain matrix and error space

---

IEEE TRANSACTIONS ON COMPUTERS 0018-9340/96\$05.00 © 1996 IEEE

Vol. 45, No. 7: JULY 1996, pp. 769-781

Theory and Application of Nongroup Cellular Automata for Synthesis of Easily Testable Finite State Machines\*

Supratik Chakraborty

Dipanwita Roy Chowdhury

Parimal Pal Chaudhuri

Abstract: This paper reports some of the interesting properties and relationships of a nongroup Cellular Automata (CA) and its dual. A special class of nongroup Cellular Automata denoted as  $D1^*CA$  is analytically investigated. Based on such analysis,  $D1^*CA$  has been proposed as an ideal test machine which can be efficiently embedded in a finite state machine to enhance the testability of the synthesized design. A state encoding algorithm has been formulated to embed the  $D1^*CA$  based test machine in the synthesized FSM while minimizing the hardware overhead. The unique state transition properties of  $D1^*CA$  are then used in designing an easy testing scheme for the FSM. Experiments on FSM benchmarks have shown that the scheme achieves 100% coverage of all single stuck-at faults at the cost of hardware overhead and circuit delay that are comparable, if not better, to that incurred for scan path based designs. However, the major advantage of the scheme is the significant reduction of test time overhead due to integration of an embedded test machine in the design at the synthesis phase.

Index Terms: Cellular automata, synthesis for testability (SFT), testable sequential machines

---

IEEE TRANSACTIONS ON COMPUTERS 0018-9340/96\$05.00 © 1996 IEEE

Vol. 45, No. 7: JULY 1996, pp. 827-840

On Uniformization of Affine Dependence Algorithms\*

Weijia Shang, Member, IEEE

Edin Hodzic

Zhigang Chen

Abstract: This paper deals with the problem of transforming irregular data dependence structures of algorithms with nested loops into more regular ones. Algorithms under consideration are  $n$ -dimensional algorithms (algorithms with  $n$  nested loops) with affine dependences where dependences are affine functions of index variables of the loop. Methods are proposed to uniformize affine dependence algorithms, i.e., to transform affine dependence algorithms into uniform dependence algorithms where dependences are independent of the index variables (constant). Objectives are considered to guide the selection of feasible uniformizations. The first one is to reduce the number of dependences after uniformization. The second one is to maximize parallelism preserved by the uniformization. Some parallelism might be lost due to the uniformization. The parallelism preserved by the uniformization is measured by 1) the total execution time by the optimal linear schedule which assigns each computation in the algorithm an execution time according to a linear function of the index of the computation, and 2) the size of the cone spanned by the dependence vectors after uniformization.

#### REFERENCES

- [1] U. Banerjee, *Dependence Analysis for Supercomputing*. Boston: Kluwer Academic, 1988.
- [2] J.A.B. Fortes and D.I. Moldvan, "Data Broadcasting in Linear Schedule Array Processors," Proc. 11th Ann. Symp. Computer Architecture, pp. 224-231, 1984.
- [3] R.M. Karp, R.E. Miller, and S. Winograd, "The Organization of Computations for Uniform Recurrence Equations," J. ACM, vol. 14, no. 3, pp. 563-590, July 1967.
- [4] G.-J. Li and B.W. Wah, "The Design of Optimal Systolic Arrays," IEEE Trans. Computers, vol. 34, no. 1, pp. 66-77, Jan. 1985.
- [5] J.-K. Peir and R. Cytron, "Minimum Distance: A Method for Partitioning Recurrences for Multiprocessors," Proc. Int'l Conf. Parallel Processing, pp. 217-225, St. Charles, Ill., 1987.
- [6] P. Quinton and V. Van Dongen, "The Mapping of Linear Recurrence Equations on Regular Arrays," J. VLSI Signal Processing, vol. 1, no. 2, pp. 95-115, Oct. 1989.

- [7]W. Shang and J.A.B. Fortes, "On the Optimality of Linear Schedule," J. VLSI Signal Processing, pp. 209-220, Jan. 1989.
- [8]W. Shang and J.A.B. Fortes, "Time-Optimal and Conflict-Free Mappings of Uniform Dependence Algorithms into Lower Dimensional Processor Arrays," Proc. Int'l Conf. Parallel Processing, pp. 101-110 (I), St. Charles, Ill., 1990.
- [9]W. Shang and J.A.B. Fortes, "Time Optimal Linear Schedules for Algorithms with Uniform Dependence," IEEE Trans. Computers, vol. 40, no. 6, pp. 723-742, June 1991.
- [10]W. Shang and J.A.B. Fortes, "Independent Partitioning of Algorithms with Uniform Dependencies," IEEE Trans. Computers, vol. 41, no. 2, pp. 190-206, Feb. 1992.
- [11]A. Schrijver, Theory of Linear and Integer Programming. New York: John Wiley & Sons, 1986.
- [12]T. Tzen and L. Ni, "Data Dependence Analysis and Uniformization for Doubly Nested Loops," Proc. Int'l Conf. Parallel Processing, pp. 91-99(II), St. Charles, Ill, 1992.
- [13]Y. Wong and J.-M. Delosme, "Transformation of Broadcasting into Pipelining," Research Report YALEU/DCS/RR-544, June 1987.
- [14]V. Van Dongen, "The Transformation of n-Dimensional Linear Recurrences into (n + 1)-Dimensional Uniform Recurrences," technical report, Philips Research Laboratory, Belgium, 1988.
- [15]V. Van Dongen and P. Quinton, "Uniformization of Linear Recurrence Equations: A Step Toward the Automatic Synthesis of Systolic Arrays," Proc. Int'l Conf. Systolic Arrays, San Diego, 1988.
- [16]Y. Yaacoby and P.R. Cappello, "Converting Affine Recurrence Equations to Quasi-Uniform Recurrence Equations," Technical Report 18, Dept. of Computer Science, Univ. of California Santa Barbara, Feb. 1988.
- [17]L.J. Mordell, Diophantine Equations, p. 30. New York: Academic Press, 1969.
- [18]W. Shang and B.W. Wah, "Dependence Analysis and Architecture Design for Bit-Level Algorithms," Proc. Int'l Conf. Parallel Processing, pp. 30-38, St. Charles, Ill, Aug. 1993.
- [19]Z. Xing and W. Shang, "An Algorithm for Accurate Data Dependence Test," Proc. IEEE Int'l Conf. Application Specific Array Processors, L. Dadda and B.W. Wah, eds., pp. 404-415, Oct. 1993.
- [20]A. Darte, L. Khachiyan, and Y. Robert, "Linear Schedule Is Close to Optimality," Proc. IEEE Int'l Conf. Application Specific Array Processors, J. Fortes, E. Lee, and T. Meng, eds., pp. 37-46, Berkeley, Calif., Aug. 1992.
- [21]P. Feautrier, "Some Efficient Solutions to the Affine Scheduling Problem, Part II, Multidimensional Time," Int'l J. Parallel Programming, vol. 21, no. 6, Dec. 1992.
- [22]P. Feautrier, "Some Efficient Solutions to the Affine Scheduling Problem, Part I, One Dimensional Time," Int'l J. Parallel Programming, vol. 21, no. 5, Oct. 1992.
- [23]P. Clauss and C. Mongenet, "Synthesis Aspects in the Design of Efficient Processor Arrays from Affine Recurrence Equations," J. Symbolic Computation, vol. 15, no2. 5-6, pp. 547-569, May-June 1993.
- [24]M. Gusev and J. Tasic, "Comparative Analysis of Methods for Broadcasting Elimination," Parallel Computing, vol. 2, no. 8, pp. 857-866, Aug. 1992.
- [25]P.M. Lenders, "Multi-Rate Arrays and Affine Recurrence Equations," Parallel Processing Letters, vol. 2, no. 4, pp. 373-380, Dec. 1992.
- [26]S.V. Rajopadhye, "Synthesizing Systolic Arrays with Control Signals from Recurrence Equations," Distributed Computing, vol. 3, no. 3, pp. 88-105, 1989

---

IEEE TRANSACTIONS ON COMPUTERS 0018-9340/96\$05.00 © 1996 IEEE

Vol. 45, No. 9: SEPTEMBER 1996, pp. 1031-1044

Phased Logic: Supporting the Synchronous Design Paradigm with Delay-Insensitive Circuitry\*

Daniel H. Linder, Member, IEEE Computer Society

James C. Harden

Abstract<sup>3/4</sup>Phased logic is proposed as a solution to the increasing problem of timing complexity in digital design. It is a delay-insensitive design methodology that seeks to restore the separation between logical and physical design by eliminating the need to distribute low-skew clock signals and carefully balance propagation delays. However, unlike other methodologies that avoid clocks, phased logic supports the cyclic, deterministic behavior of the synchronous design paradigm. This permits the designer to rely chiefly on current experience and CAD tools to create phased logic systems. Marked graph theory is used as a framework for governing the interaction of phased logic gates that operate directly on Level-Encoded two-phase Dual-Rail (LEDR) signals. A synthesis algorithm is developed for converting clocked systems to phased logic systems and is applied to benchmark examples. Performance results indicate that phased logic tends to be tolerant of logic delay imbalances and has predictable worst-case timing behavior. Although phased logic requires additional circuitry, it has the potential to shorten the design cycle by reducing timing complexities.

Index Terms<sup>3</sup>Asynchronous circuitry, data flow, delay-insensitive circuitry, dual-rail encoding, LEDR, marked graphs, phased logic, synchronous circuitry.

---

IEEE TRANSACTIONS ON COMPUTERS 0018-9340/96\$05.00 © 1996 IEEE

Vol. 45, No. 12: DECEMBER 1996, pp. 1439-1444

Optimal Data Scheduling for Uniform Multidimensional Applications\*

Qingyan Wang

Nelson Luiz Passos, Member, IEEE

Edwin Hsing-Mean Sha, Member, IEEE

Abstract<sup>3</sup>Uniform nested loops are broadly used in scientific and multidimensional digital signal processing applications. Due to the amount of data handled by such applications, on-chip memory is required to improve the data access and overall system performance. In this study a static data scheduling method, carrot-hole data scheduling, is proposed for multidimensional applications, in order to control the data traffic between different levels of memory. Based on this data schedule, optimal partitioning and scheduling are selected. Experiments show that by using this technique, on-chip memory misses are significantly reduced as compared to results obtained from traditional methods. The carrot-hole data scheduling method is proven to obtain smallest on-chip memory misses compared with other linear scheduling and partitioning schemes.

Index Terms<sup>3</sup>Execution scheduling, data scheduling, nested loops, memory management, memory hierarchy, partitioning, digital signal processing, multidimensional application.

---

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 1045-9219/96\$05.00 © 1996 IEEE

Vol. 7, No. 3: MARCH 1996, pp. 274-287

Optimal Synthesis of Algorithm-Specific Lower-Dimensional Processor Arrays\*

Kumar N. Ganapathy

Benjamin W. Wah, Fellow, IEEE

Abstract<sup>3</sup>Processor arrays are frequently used to deliver high performance in many applications with computationally intensive operations. This paper presents the General Parameter Method (GPM), a systematic parameter-based approach for synthesizing such algorithm-specific architectures. GPM can synthesize processor arrays of any lower dimension from a uniform-recurrence description of the algorithm. The design objective is a general nonlinear and nonmonotonic user-specified function, and depends on attributes such as computation time of the recurrence on the processor array, completion time, load time, and drain time. In addition, bounds on some or all of these attributes can be specified. GPM performs an efficient search of polynomial complexity to find the optimal design satisfying the user-specified design constraints. As an illustration, we show how GPM can be used to find optimal linear processor arrays for computing transitive closures. We consider design objectives that minimize computation time, or processor count, or completion time (including load and drain times), and user-specified constraints on number of processing elements and/or computation/completion times. We show that GPM can be used to obtain optimal designs that trade between number of processing elements and completion time, thereby allowing the designer to choose a design that best meets the specified design objectives. We also show the equivalence between the model assumed in GPM and that in the popular dependence-based methods [1], [2]. Consequently, GPM can be used to find optimal designs for both models.

Index Terms<sup>3</sup>Design constraints, objective function, optimal design, polynomial-time search, processor arrays, transitive closure, uniform recurrence equations.

#### REFERENCES

- [1]R.H. Kuhn, "Optimization and Interconnections Complexity for Parallel Processors, Single Stage Networks, and Decision Trees," PhD dissertation, Dept. of Computer Science, Univ. of Illinois, Urbana, Oct. 1980.
- [2]D.I. Moldovan, "On the Analysis and Synthesis of VLSI Algorithms," IEEE Trans. Computers, vol. 31, no. 11, pp. 1,121-1,126, Nov. 1982.
- [3]H.T. Kung, "Why Systolic Architectures?" Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982
- [4]J.A.B. Fortes, K.-S. Fu, and B.W. Wah, "Systematic Design Approached for Algorithmically Specified Systolic Arrays," Computer Architecture: Concepts and Systems, V.M. Milutinovic, ed., pp. 454-494. North Holland, 1988.
- [5]Z. Chen and W. Shang, "On Uniformization of Affine Dependence Algorithms," Proc. Fourth Symp. Parallel and Distributed Systems, vol. 3, pp. 128-137, Dec. 1992.
- [6]W. Shand and J.A.B. Fortes, "On Mapping of Uniform Dependence Algorithms into Lower Dimensional Processor Arrays," IEEE Trans. Parallel and Distributed Systems, vol. 3, no. 5, pp. 350-363, May 1992.
- [7]P.-Z. Lee and Z.M. Kedem, "Mapping Nested Loop Algorithms into Multidimensional Systolic Arrays," IEEE Trans. Parallel and Distributed Systems, vol. 1, no. 1, pp. 64-76, Jan. 1990.



- [8]P.-Z. Lee and Z.M. Kedem, "Synthesizing Linear Array Algorithms from Nested for Loop Algorithms," IEEE Trans. Computers, vol. 37, no. 12, pp. 1,578-1,597, Dec. 1988.
- [9]V.P. Roychowdhury and T. Kailath, "Subspace Scheduling and Parallel Implementation of Non-Systolic Regular Iterative Algorithms," J.VLSI Signal Processing, vol. 1. Kluwer Academic, 1989.
- [10]G.-J. Li and B.W. Wah, "The Design of Optimal Systolic Arrays," IEEE Trans. Computers, vol. 34, no. 1, pp. 66-77, Jan. 1985.
- [11]M.T. O'Keefe, J.A.B. Fortes, and B.W. Wah, "On the Relationship Between Systolic Array Design Methodologies," IEEE Trans. Computers, vol. 41, no. 12, pp. 1,589-1,593, Dec. 1991.
- [12]J.A.B. Fortes, B.W. Wah, W. Shang, and K.N. Ganapathy, "Algorithm-Specific Parallel Processing with Linear Processor Arrays," Advances in Computers, M. Yovits, ed. Academic Press, 1994.
- [13]K. Ganapathy, "Mapping Regular Recursive Algorithms to Fine-Grained Processor Arrays," PhD dissertation, Univ. of Illinois, Urbana-Champaign, Apr. 1994.
- [14]K.N. Ganapathy and B.W. Wah, "Synthesizing Optimal Lower Dimensional Processor Arrays," Proc. Int'l Conf. Parallel Processing, pp. 96-103. Pennsylvania State Univ. Press, Aug. 1992.
- [15]J. Zue, "A New Formulation of the Mapping Conditions for the Synthesis of Linear Systolic Arrays," Proc. Application Specific Array Processors, pp. 297-308. IEEE CS Press, 1993.
- [16]K.N. Ganapathy and B.W. Wah, "Optimal Design of Lower Dimensional Processor Arrays for Uniform Recurrences," Proc. Application Specific Array Processors, pp. 636-648. IEEE CS Press, Aug. 1992.
- [17]S.Y. Kung, S.C. Lo, and P.S. Lewis, "Optimal Systolic Design for Transitive Closure and Shortest Path Problems," IEEE Trans. Computers, vol. 36, no. 5, pp. 603-614, May 1987.
- [18]G. Rote, "A Systolic Array for Algebraic Path Problem," Computing, vol. 34, pp. 192-219. Springer-Verlag, 1985.
- [19]K.N. Ganapathy and B.W. Wah, "Designing a Coprocessor for Regular Recurrent Computations," Proc. Fifth IEEE Symp. Parallel and Distributed Systems, pp. 806-813, Dec.

---

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 1045-9219/96\$05.00 © 1996 IEEE  
Vol. 7, No. 4: APRIL 1996, pp. 399-410

Valid Transformations: A New Class of Loop Transformations for High-Level Synthesis and Pipelined Scheduling Applications\*

Minjoong Rim

Rajiv Jain

Abstract<sup>3</sup>In this paper we present a new class of loop optimizing transformations called valid transformations, which are suitable for fine-grain parallelization applications such as high-level synthesis of VLSI designs or compilers for super-scalar or VLIW machines. This class of transformations are different from existing ones in that valid transformations can be illegal. Nevertheless, if a transformation is valid, the transformed loop has a feasible pipeline schedule. We present an example valid transformation called loop expansion which can help produce cost-performance efficient designs and explore a larger design space for a satisfactory design. Several examples are used to demonstrate the efficacy of the proposed technique.

Index Terms<sup>3</sup>High-level synthesis, super-scalar, VLIW, loop compilation, loop optimization, loop transformations, pipeline scheduling.

#### REFERENCES

- [1]M.C. McFarland, A.C. Parker, and R. Camposano, "The High-Level Synthesis of Digital Systems," Proc. IEEE, vol. 78, Feb. 1990.
- [2]C.-T. Hwang, Y.-C. Hsu, and Y.-L. Lin, "Scheduling for Functional Pipelining and Loop Winding," Proc. ACM/IEEE Design Automation Conf., 1991.
- [3]L.F. Chao, A. LaPaugh, and E.H. Sha, "Rotation Scheduling: A Loop Pipelining Algorithm," Proc. ACM/IEEE Design Automation Conf., 1993.
- [4]M. Lam, "Software Pipelining: An Effective Scheduling Technique for VLIW Machines," Proc. ACM SIGPLAN Conf. Programming Language Design and Implementation, 1988.
- [5]E. Girczyc, "Loop Winding<sup>3</sup>A Data Flow Approach to Functional Programming," Proc. IEEE Int'l Symp. Circuits and Systems 1987.
- [6]G. Goossens, J. Vandewalle, and H. De Man, "Loop Optimization in Register-Transfer Scheduling for DSP-Systems," Proc. ACM/IEEE Design Automation Conf., 1989.
- [7]C.Y. Wang and K.K. Parhi, "Loop List Scheduler for DSP Algorithms Under Resource Constraints," Proc. Int'l Symp. Circuits and Systems, IEEE, 1993.

- [8]A.E. Charlesworth, "An Approach to Scientific Array Processing: The Architectural Design of the AP-120B/FPS-164 Family," *Computer*, Sept. 1981.
- [9]J.C. Dehner, P.Y.T. Hsu, and J.P. Bratt, "Overlapped Loop Support in the Cydra 5," *Proc. ACM Int'l Conf. Architectural Support for Programming Languages and Operating Systems*, 1989.
- [10]K. Ebcioglu and T. Nakatani, "A New Compilation Technique for Parallelizing Loops with Unpredictable Branches on a VLIW Architecture," *Languages and Compilers for Parallel Computing*, D. Gelernter, A. Nicolau, and D. Padua, eds. MIT Press, 1990.
- [11]B.R. Rau and C.D. Glaeser, "Some Scheduling Techniques and an Easily Schedulable Horizontal Architecture for High Performance Scientific Computing," *Proc. IEEE 14th Ann. Workshop Microprogramming*, 1981.
- [12]B. Su, S. Ding, and J. Xia, "GURPR $\frac{3}{4}$ A Method for Global Software Pipelining," *Proc. ACM 20th Ann. Workshop Microprogramming*, 1987.
- [13]R.F. Touzeau, "A Fortran Compiler for the FPS-164 Scientific Computer," *Proc. ACM SIGPLAN '84 Symp. Compiler Construction*, 1984.
- [14]M.E. Wolf and M.S. Lam, "A Loop Transformation Theory and an Algorithm to Maximize Parallelism," *IEEE Trans. Parallel and Distributed Systems*, vol. 2, Oct. 1991.
- [15]M.J. Wolfe, *Optimizing Supercompilers for Supercomputers*. MIT Press, 1989.
- [16]S.Y. Kung, *VLSI Array Processors*. Prentice Hall, 1988.
- [17]J.A. Nestor, "Specification and Synthesis of Digital Systems with Interfaces," PhD thesis, Dept. of Electrical Eng., Carnegie Mellon Univ., Apr. 1987.
- [18]B.M. Pangrle and D.D. Gajski, "Design Tools for Intelligent Silicon Compilation," *IEEE Trans. Computer-Aided Design*, vol. 6, Nov. 1987.
- [19]N. Park and A.C. Parker, "Sehwa: A Software Package for Synthesis of Pipelines from Behavioral Specifications," *IEEE Trans. Computer-Aided Design*, vol. 7, Mar. 1988.
- [20]A.C. Parker, J. Pizarro, and M.J. Mlinar, "MAHA: A Program for Datapath Synthesis," *Proc. ACM/IEEE Design Automation Conf.*, 1986.
- [21]H. Trickey, "Flamel: A High-Level Hardware Compiler," *IEEE Trans. Computer-Aided Design*, vol. 6, Mar. 1987.
- [22]P.G. Paulin and J.P. Knight, "Force-Directed Scheduling for the Behavioral Synthesis of ASIC's," *IEEE Trans. Computer-Aided Design*, vol. 8, June 1989.
- [23]R. Camposano, "Path-Based Scheduling for Synthesis," *IEEE Trans. Computer-Aided Design*, vol. 10, Jan. 1991.
- [24]L. Lamport, "The Parallel Execution of DO Loops," *Comm. ACM*, vol. 17, Feb. 1974.
- [25]C.D. Polychronopoulos, D.J. Kuck, and D.A. Padua, "Utilizing Multidimensional Loop Parallelism on Large-Scale Parallel Processor Systems," *IEEE Trans. Computers*, vol. 38, Sept. 1989.
- [26]J.H. Saltz, R. Mirchandaney, and K. Crowley, "Run-Time Parallelization and Scheduling of Loops," *IEEE Trans. Computers*, vol. 40, May 1991.
- [27]W. Shang and J.A.B. Fortes, "Time Optimal Linear Schedules for Algorithms with Uniform Dependencies," *IEEE Trans. Computers*, vol. 40, June 1991.
- [28]P.S. Tseng, "Compiling Programs for a Linear Systolic Array," *Proc. ACM SIGPLAN Conf. Programming Language Design and Implementation*, 1990.
- [29]C. Wang and S. Wang, "Efficient Processor Assignment Algorithms and Loop Transformations for Executing Nested Parallel Loops on Multiprocessors," *IEEE Trans. Parallel and Distributed Systems*, vol. 3, Jan. 1992.
- [30]A. Aiken and A. Nicolau, "Perfect Pipelining: A New Loop Parallelization Technique," *Proc. 1988 European Symp. Programming*, 1988.
- [31]P.M. Kogge, "The Microprogramming of Pipelined Processors," *Proc. ACM/IEEE Int'l Symp. Computer Architecture*, 1977.
- [32]J.H. Patel and E.S. Davidson, "Improving the Throughput of a Pipeline by Insertion of Delays," *Proc. ACM/IEEE Int'l Symp. Computer Architecture*, 1976.
- [33]A. Zaky and P. Sadayappan, "Optimal Static Scheduling of Sequential Loops on Multiprocessors," *Proc. Int'l Conf. Parallel Processing*, 1989.
- [34]A. Aiken and A. Nicolau, "Fine-Grain Parallelization and the Wavefront Method," *Languages and Compilers for Parallel Computing*, D. Gelernter, A. Nicolau, and D. Padua, eds. MIT Press, 1990.
- [35]A. Aiken and A. Nicolau, "Optimal Loop Parallelization," *Proc. ACM SIGPLAN Conf. Programming Language Design and Implementation*, 1988.
- [36]Y. Wong and J.-M. Delosme, "Optimization of Computation Time for Systolic Arrays," *IEEE Trans. Computer-Aided Design*, vol. 11, Feb. 1992.

- [37]A. Wolfe and J.P. Shen, "A Variable Instruction Stream Extension to the VLIW Architecture," Proc. ACM Int'l Conf. Architectural Support for Programming Languages and Operating Systems, 1991.
- [38]H. Zima and B. Chapman, Supercompilers for Parallel and Vector Computers. ACM Press, 1990.
- [39]Z. Li, P.-C. Yew, and C.-Q. Zhu, "An Efficient Data Dependence Analysis for Parallelizing Compilers," IEEE Trans. Parallel and Distributed Systems, vol. 1, Jan. 1990.
- [40]D.E. Maydan, J.L. Hennessy, and M.S. Lam, "Efficient and Exact Data Dependence Analysis," Proc. ACM SIGPLAN Conf. Programming Language Design and Implementation, 1991.
- [41]M. Wolfe and C.-W. Tseng, "The Power Test for Data Dependence," IEEE Trans. Parallel and Distributed Systems, vol. 3, Sept. 1992.
- [42]V.P. Krothapalli and P. Sadayappan, "Removal of Redundant Dependences in DOACROSS Loops with Constant Dependences," IEEE Trans. Parallel and Distributed Systems, vol. 2, July 1991.
- [43]U. Banerjee, R. Eigenmann, A. Nicolau, and D.A. Padua, "Automatic Program Parallelization," Proc. IEEE, vol. 81, Feb. 1993.
- [44]D.A. Padua and M.J. Wolfe, "Advanced Compiler Optimizations for Supercomputers," Comm. ACM, vol. 29, Dec. 1986.
- [45]A. Darté and Y. Robert, "Constructive Methods for Scheduling Uniform Loop Nests," IEEE Trans. Parallel and Distributed Systems, vol. 5, Aug. 1994.
- [46]M. Rim, "High-Level Synthesis of VLSI Designs for Scientific Programs," PhD thesis, Dept. of Electrical and Computer Eng., Univ. of Wisconsin, Madison, Aug. 1993.
- [47]M. Rim and R. Jain, "Estimating Performance Characteristics for Loop Transformations," Proc. IEEE Int'l Symp. Circuits and Systems, 1994.
- [48]D.C. Ku and G. De Micheli, "Relative Scheduling Under Timing Constraints: Algorithms for High-Level Synthesis of Digital Circuits," IEEE Trans. Computer-Aided Design, vol. 11, June 1992.
- [49]Y. Liao and C. Wong, "An Algorithm to Compact a VLSI Symbolic Layout with Mixed Constraints," IEEE Trans. Computer-Aided Design, vol. 2, Apr. 1983.
- [50]R. Karp, R. Miller, and S. Winograd, "The Organization of Computations for Uniform Recurrence Equations," J. ACM, vol. 14, July 1967.
- [51]D. Callahan, J. Cocke, and K. Kennedy, "Estimating Interlock and Improving Balance for Pipelined Architectures," Proc. Int'l Conf. Parallel Processing, 1987.
- [52]M. Wolfe, "The Tiny Loop Restructuring Research Tool," Proc. Int'l Conf. Parallel Processing, 1991.
- [53]M. Wolfe, "Loop Skewing: The Wavefront Method Revisited," Int'l J. Parallel Programming, vol. 15, Aug. 1986.

---

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 1045-9219/96\$05.00 © 1996 IEEE  
Vol. 7, No. 5: MAY 1996, pp. 449-455

A Modular Systolic Linearization of the Warshall-Floyd Algorithm\*

Jean Frédéric Myoupo, Member, IEEE Computer Society

Anne Cécile Fabret

Abstract<sup>3</sup>In this paper, we use a variant of the geometric method to derive efficient modular linear systolic algorithms for the transitive closure and shortest path problems. Furthermore, we show that partially-pipelined modular linear systolic algorithms with an output operation, for matrix multiplication, can be as fast as the fully-pipelined existing ones and, moreover, they need less cells.

Index Terms<sup>3</sup>Modular linear systolic algorithms, transitive closure, shortest path, matrix multiplication.

#### REFERENCES

- [1]A. Benaini and M. Tchente, "Matrix Product on Linear Systolic Arrays," Parallel and Distributed Algorithms, M. Cosnard, P. Quinton, M. Raynal and Y. Robert, eds., North Holland, 1989.
- [2]A.L. Fischer and H.T. Kung, "Synchronizing Large VLSI Processor Arrays," Proc. Tenth Ann. IEEE/ACM Symp. Computer Architecture, pp. 54-58, June 1983.
- [3]L.J. Guibas, H.T. Kung, and C.D. Thompson, "Direct VLSI Implementation of Combinatorial Algorithms, Proc. Conf. Very Large Scale Integration: Architecture, Design, Fabrication, pp. 509-525, California Institute of Technology Inst., Jan. 1979.
- [4]H.T. Kung, "Why Systolic Architecture," IEEE Computer, vol. 15, no. 1, pp. 37-46, Jan. 1980.
- [5]F.T. Leighton and C.E. Leiserson, "Wafer-Scale Integration of Systolic Arrays," Proc. 23rd Symp. Foundations of Computer Science, pp. 297-311, Nov. 1982.
- [6]S.Y. Kung, S.C. Lo, and P.S. Lewis, "Optimal Systolic Design for Transitive Closure and the Shortest Path Problems, IEEE Trans. Computer, vol. 36, no. 5, pp. 603-614, 1987.

- [7]P. Lee and Z. Kedem, "Synthesizing Linear Array Algorithms From Nested For Loop Algorithms, IEEE Trans. Computer., vol. 37, pp. 1578-1598, 1988.
- [8]P. Lee and Z. Kedem, "On High Speed Computing with a Programmable Linear Array Synthesizing Linear Array Algorithms From Nested For Loop Algorithms," J. Supercomputing. vol. 4, pp. 223-249, 1990.
- [9]D.I. Moldovan and R.A.B. Fortes, "Partitioning and Mapping of Algorithms into Fixed Size Systolic Arrays," IEEE Trans. Computers, vol. 35, pp. 1-12, 1986.
- [10]J.F. Myoupo, "A Linear Systolic Array for Transitive Closure Problems, Proc. Int'l Conf. Parallel Process. (ICPP), vol. 1, pp. 617-618, 1990.
- [11]J. F. Myoupo and A.C. Fabret, "Designing Modular Linear Systolic Arrays Using Dependence Graph Regular Partitions, Rapport Interne, L.R.I. N° 760, Université Paris XI, 1992.
- [12]V.K. Prasanna Kumar and Y.C. Tsai, "Designing Linear Systolic Arrays, J. Parallel Distributed Computers, vol. 7, pp. 441-463, 1989.
- [13]S.K. Prasanna Kumar and Y.C. Tsai, "On Mapping Algorithms to Linear and Fault-Tolerant Systolic Arrays," IEEE Trans. Computers, vol. 38, no. 3, pp. 470-478, 1989.
- [14]I.V. Ramakrishnan, P.J. Varman, "Dynamic Programming and Transitive Closure on Linear Pipelines," Proc. Int'l. Conf. on Parallel Processing, (ICPP) 1984.
- [15]I.V. Ramakrishnan, D.S. Fussel, and A. Silberschatz, "Mapping Homogenous Graphs on Linear Arrays," IEEE Trans. Computers, vol. 35, pp. 189-209, 1986.
- [16]I.V. Ramakrishnan and P.J. Varman, "Synthesis of An Optimal Family of Matrix Multiplication Algorithms on Linear Arrays, IEEE Trans. Computers, vol. 35, no. 11, 1986.
- [17]T. Risset, "Linear Systolic Arrays for Matrix Multiplication: Comparison of Existing Synthesis Methods and New Results," Algorithms and Parallel VLSI Architectures II, pp. 163-174, P. Quinton and Y. Robert eds. North Holland: Elsevier, 1991.
- [18]G. Rote, "A Systolic Array Algorithm for the Algebraic Path Problem," Computing, vol. 34, pp. 192-219, 1985.
- [19]W. Shang and J.A.B. Fortes, "On Time Mapping of Uniform Dependence Algorithms into Lower Dimensional Processor Arrays," IEEE Trans. Parallel Distributed Computers, vol. 3, pp. 350-363, 1992.
- [20]U. Schwiegelshohn and L. Thiele, "Linear Systolic Arrays for Matrix Computation," J. Parallel Distributed Computers, vol. 7, pp. 28-39, 1989.
- [21]J. Teich and L. Thiele, "A Transformative Approach to the Partitioning of Processor Arrays," Proc. ASAP 1992, IEEE C. S. Press, pp. 4-20.
- [22]J.D. Ullman, Computational aspects of VLSI, IEEE C. S. Press, 1984.
- [23]J.L.A. Van de Snepscheut, "A Derivation of a Distributed Implementation of Warshall's Algorithm," Science of Computer Programming, vol. 7, pp. 55-60, 1986.
- [24]S.W. Warman Jr., "A Modification of Warshall's Algorithm for the Transitive Closure of Binary Relations," CACM, vol. 18, no. 4, pp. 218-220, 1975.
- [25]S. Warshall, "A Theorem on Boolean Matrices," J. ACM, vol. 9 no. 1, pp. 11-12, 1972.

---

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 1045-9219/96\$05.00 © 1996 IEEE  
Vol. 7, No. 7: JULY 1996, pp. 689-704

A Unified Framework for Optimizing Communication in Data-Parallel Programs\*

Manish Gupta

Edith Schonberg

Harini Srinivasan

Abstract<sup>3/4</sup>This paper presents a framework, based on global array data-flow analysis, to reduce communication costs in a program being compiled for a distributed memory machine. We introduce available section descriptor, a novel representation of communication involving array sections. This representation allows us to apply techniques for partial redundancy elimination to obtain powerful communication optimizations. With a single framework, we are able to capture optimizations like 1) vectorizing communication, 2) eliminating communication that is redundant on any control flow path, 3) reducing the amount of data being communicated, 4) reducing the number of processors to which data must be communicated, and (5) moving communication earlier to hide latency, and to subsume previous communication. We show that the bidirectional problem of eliminating partial redundancies can be decomposed into simpler unidirectional problems even in the context of an array section representation, which makes the analysis procedure more efficient. We present results from a preliminary implementation of this framework, which are extremely encouraging, and demonstrate the effectiveness of this analysis in improving the performance of programs.

Index Terms<sup>3</sup>Array section descriptors, communication optimizations, data availability, data-flow analysis, data-parallelism, High Performance Fortran, partial redundancy elimination.

---

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 1045-9219/96\$05.00 © 1996 IEEE

Vol. 7, No. 8: AUGUST 1996, pp. 830-840

The Block Distributed Memory Model\*

Joseph F. JáJá

Kwan Woo Ryu

Abstract<sup>3</sup>We introduce a computation model for developing and analyzing parallel algorithms on distributed memory machines. The model allows the design of algorithms using a single address space and does not assume any particular interconnection topology. We capture performance by incorporating a cost measure for interprocessor communication induced by remote memory accesses. The cost measure includes parameters reflecting memory latency, communication bandwidth, and spatial locality. Our model allows the initial placement of the input data and pipelined prefetching.

We use our model to develop parallel algorithms for various data rearrangement problems, load balancing, sorting, FFT, and matrix multiplication. We show that most of these algorithms achieve optimal or near optimal communication complexity while simultaneously guaranteeing an optimal speed-up in computational complexity. Ongoing experimental work in testing and evaluating these algorithms has thus far shown very promising results.

Index Terms<sup>3</sup>Parallel algorithms, parallel model, personalized communication, broadcasting, load balancing, sorting, Fast Fourier Transform, matrix multiplication.

---

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 1045-9219/96\$05.00 © 1996 IEEE

Vol. 7, No. 11: NOVEMBER 1996, pp. 1133-1149

A Framework for Resource-Constrained Rate-Optimal Software Pipelining\*

R. Govindarajan, Member, IEEE Computer Society

Erik R. Altman

Guang R. Gao, Senior Member, IEEE

Abstract<sup>3</sup>The rapid advances in high-performance computer architecture and compilation techniques provide both challenges and opportunities to exploit the rich solution space of software pipelined loop schedules. In this paper, we develop a framework to construct a software pipelined loop schedule which runs on the given architecture (with a fixed number of processor resources) at the maximum possible iteration rate (à la rate-optimal) while minimizing the number of buffers<sup>3</sup>a close approximation to minimizing the number of registers.

The main contributions of this paper are: First, we demonstrate that such problem can be described by a simple mathematical formulation with precise optimization objectives under a periodic linear scheduling framework. The mathematical formulation provides a clear picture which permits one to visualize the overall solution space (for rate-optimal schedules) under different sets of constraints. Secondly, we show that a precise mathematical formulation and its solution does make a significant performance difference. We evaluated the performance of our method against three leading contemporary heuristic methods. Experimental results show that the method described in this paper performed significantly better than these methods.

The techniques proposed in this paper are useful in two different ways: 1) As a compiler option which can be used in generating faster schedules for performance-critical loops (if the interested users are willing to trade the cost of longer compile time with faster runtime). 2) As a framework for compiler writers to evaluate and improve other heuristics-based approaches by providing quantitative information as to where and how much their heuristic methods could be further improved.

Index Terms<sup>3</sup>Instruction-level parallelism, instruction scheduling, integer linear programming, software pipelining, superscalar and VLIW architectures.

#### REFERENCES

- [1]A. Aiken, "Compaction-Based Parallelization," Technical Report TR 88-922, Dept. of Computer Science, Cornell Univ., Ithaca, N.Y., June 1988, PhD thesis.
- [2]A. Aiken and A. Nicolau, "Optimal Loop Parallelization," Proc. SIGPLAN '88 Conf. Programming Language Design and Implementation, pp. 308-317, Atlanta, Ga., June 22-24, 1988. SIGPLAN Notices, vol. 23, no. 7, July 1988.
- [3]A. Aiken and A. Nicolau, "A Realistic Resource-Constrained Software Pipelining Algorithm," Advances in Languages and Compilers for Parallel Processing, A. Nicolau, D. Gelernter, T. Gross, and D. Padua, eds., chapter 14, pp. 274-290, Research Monographs in Parallel and Distributed Computing. London, U.K., and Cambridge,

- Mass.: Pitman Publishing and MIT Press, 1991. Selected papers from the Third Workshop on Languages and Compilers for Parallel Computing, Irvine, Calif., Aug. 1-2, 1990.
- [4]J.C. Dehnert and R.A. Towle, "Compiling for Cydra 5," J. Supercomputing, vol. 7, pp. 181-227, May 1993.
- [5]K. Ebcioglu, "A Compilation Technique for Software Pipelining of Loops with Conditional Jumps," Proc. 20th Ann. Workshop Microprogramming, pp. 69-79, Colorado Springs, Colo., Dec. 1-4, 1987.
- [6]K. Ebcioglu and A. Nicolau, "A Global Resource-Constrained Parallelization Technique," Proc. 1989 Int'l Conf. Supercomputing, pp. 154-163, Crete, Greece, June 5-9, 1989.
- [7]R.A. Huff, "Lifetime-Sensitive Modulo Scheduling," Proc. ACM SIGPLAN '93 Conf. Programming Language Design and Implementation, pp. 258-267, Albuquerque, N.M., June 23-25, 1993. SIGPLAN Notices, vol. 28, no. 6, June 1993.
- [8]M. Lam, "Software Pipelining: An Effective Scheduling Technique for VLIW Machines," Proc. SIGPLAN '88 Conf. Programming Language Design and Implementation, pp. 318-328, Atlanta, Ga., June 22-24, 1988. SIGPLAN Notices, vol. 23, no. 7, July 1988.
- [9]S.-M. Moon and K. Ebcioglu, "An Efficient Resource-Constrained Global Scheduling Technique for Superscalar and VLIW Processors," Proc. 25th Ann. Int'l Symp. Microarchitecture, pp. 55-71, Portland, Ore., Dec. 1-4, 1992. SIG MICRO Newsletter, vol. 23, nos. 1-2, Dec. 1992.
- [10]A. Nicolau, K. Pingali, and A. Aiken, "Fine-Grain Compilation for Pipelined Machines," Technical Report TR 88-934, Dept. of Computer Science, Cornell Univ., Ithaca, N.Y., 1988.
- [11]Q. Ning and G.R. Gao, "A Novel Framework of Register Allocations for Software Pipelining," Conf. Record 20th Ann. ACM SIGPLAN-SIGACT Symp. Principles Programming Languages, pp. 29-42, Charleston, S.C., Jan. 10-13,
- [12]B.R. Rau and C.D. Glaeser, "Some Scheduling Techniques and an Easily Schedulable Horizontal Architecture for High Performance Scientific Computing," Proc. 14th Ann. Microprogramming Workshop, pp. 183-198, Chatham, Mass., Oct. 12-15, 1981.
- [13]B.R. Rau, M. Lee, P.P. Tirumalai, and M.S. Schlansker, "Register Allocation for Software Pipelined Loops," Proc. ACM SIGPLAN '92 Conf. Programming Language Design and Implementation, pp. 283-299, San Francisco, June 17-19, 1992. SIGPLAN Notices, vol. 27, no. 7, July 1992.
- [14]R.F. Touzeau, "A Fortran Compiler for the FPS-164 Scientific Computer," Proc. SIGPLAN '84 Symp. Compiler Construction, pp. 48-57, Montréal, Québec, Canada, June 17-22, 1984. SIGPLAN Notices, vol. 27, no. 7, July 1992.
- [15]V. Van Dongen, G.R. Gao, and Q. Ning, "A Polynomial Time Method for Optimal Software Pipelining," Proc. Conf. Vector and Parallel Processing, CONPAR-92, Lecture Notes in Computer Science 634, pp. 613-624, Lyons, France, Springer-Verlag, Sept. 1-4, 1992.
- [16]P. Feautrier, "Dataflow Analysis of Scalar and Array References," Int'l J. Parallel Programming, vol. 20, no. 1, pp. 23-53, 1991.
- [17]L.J. Hendren and G.R. Gao, "Designing Programming Languages for Analyzability: A Fresh Look at Pointer Data Structures," Proc. 1992 Int'l Conf. Computer Languages, pp. 242-251, Oakland, Calif., IEEE CS Press, Apr. 20-23, 1992.
- [18]J.J. Dongarra and A.R. Hinds, "Unrolling Loops in FORTRAN," Software Practice and Experience, vol. 9, pp. 219-226, Mar. 1979.
- [19]J. Wang, C. Eisenbeis, M. Jourdan, and B. Su, "Decomposed Software Pipelining: A New Approach to Exploit Instruction-Level Parallelism for Loop Programs," Research Report No. 1838, Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt, France, Jan. 1993.
- [20]F. Gasperoni and U. Schwiegelshohn, "Efficient Algorithms for Cyclic Scheduling," Research Report RC 17068, IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 1991.
- [21]N.J. Warter, S.A. Mahlke, W.-M.W. Hwu, and B.R. Rau, "Reverse If-Conversion," Proc. ACM SIGPLAN '93 Conf. Programming Language Design and Implementation, pp. 290-299, Albuquerque, N.M., June 23-25, 1993. SIGPLAN Notices, vol. 28, no. 6, June 1993.
- [22]E.R. Altman, R. Govindarajan, and G.R. Rao, "Scheduling and Mapping: Software Pipelining in the Presence of Structural Hazards," Proc. ACM SIGPLAN '95 Conf. Programming Language Design and Implementation, pp. 139-150, La Jolla, Calif., June 18-21, 1995. SIGPLAN Notices, vol. 30, no. 6, June 1995.
- [23]R. Reiter, "Scheduling Parallel Computations," J. ACM, vol. 15, pp. 590-599, Oct. 1968.
- [24]B.R. Rau, D.W.L. Yen, W. Yen, and R.A. Towle, "The Cydra 5 Departmental Supercomputer Design Philosophies, Decisions, and Trade-Offs," Computer, vol. 22, no. 1, pp. 12-35, Jan. 1989.
- [25]Q. Ning, "Register Allocation for Optimal Loop Scheduling," PhD thesis, McGill Univ., Montréal, Québec, Canada, 1993.

- [26]E.R. Altman, "Optimal Software Pipelining with Function Unit and Register Constraints," PhD thesis, McGill Univ., Montréal, Québec, Canada, Oct. 1995.
- [27]A.E. Eichenberger, E.S. Davidson, and S.G. Abraham, "Minimum Register Requirements for a Modulo Schedule," Proc. 27th Ann. Int'l Symp. Microarchitecture, pp. 75-84, San Jose, Calif., Nov. 30- Dec. 2, 1994.
- [28]J.R. Allen, K. Kennedy, C. Porterfield, and J. Warren, "Conversion of Control Dependence to Data Dependence," Conf. Record 10th Ann. ACM Symp. Principles of Programming Languages, pp. 177-189, Austin, Tex., Jan. 24-26, 1983.
- [29]L.J. Henren, G.R. Gao, E.R. Altman, and C. Mukerji, "A Register Allocation Framework Based on Hierarchical Cyclic Interval Graphs," Proc. Fourth Int'l Conf. Compiler Construction, CC '92, U. Kasterns and P. Pfahler, eds., pp. 176-191, Lecture Notes in Computer Science, Paderborn, Germany, Springer-Verlag, Oct. 5-7, 1992.
- [30]T.C. Hu, Integer Programming and Network Flows, p. 270. Addison-Wesley, 1969.
- [31]J. Wang and E. Eisenbeis, "A New Approach to Software Pipelining for Complicated Loops with Branches," research report, Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt, France, Jan. 1993.
- [32]G. Gao and Q. Ning, "Loop Storage Optimization for Dataflow Machines," Proc. Fourth Int'l Workshop Languages and Compilers for Parallel Computing, U. Banerjee, D. Gelernter, A. Nicolau, and D. Padua, eds., Lecture Notes in Computer Science 589, pp. 359-373, Santa Clara, Calif., Intel Corp., Springer-Verlag, Aug. 7-9, 1991, published 1992.
- [33]B.R. Rau and J.A. Fisher, "Instruction-Level Parallel Processing: History, Overview and Perspective," J. Supercomputing, vol. 7, pp. 9-50, May 1993.
- [34]J.C. Dehnert, P.Y.-T. Hsu, and J.P. Bratt, "Overlapped Loop Support in the Cydra 5," Proc. Third Int'l Conf. Architectural Support for Programming Languages and Operating Systems, pp. 26-38, Boston, Apr. 3-6, 1989. Computer Architecture News, vol. 17, no. 2, Apr. 1989; Operating Systems Review, vol. 23, Apr. 1989; SIGPLAN Notices, vol. 24, May 1989.
- [35]S. Ramakrishnan, "Software Pipelining in PA-RISC Compilers," Hewlett-Packard J., pp. 39-45, June 1992.
- [36]B.R. Rau, M.S. Schlansker, and P.P. Tirumalai, "Code Generation Schema for Modulo Scheduled Loops," Proc. 25th Ann. Int'l Symp. Microarchitecture, pp. 158-169, Portland, Ore., Dec. 1-4, 1992. SIG MICRO Newsletter, vol. 23, nos. 1-2, Dec. 1992.
- [37]M. Rajagopalan and V.H. Allan, "Efficient Scheduling of Fine Grain Parallelism in Loops," Proc. 26th Ann. Int'l Symp. Microarchitecture, pp. 2-11, Austin, Tex., Dec. 1-3, 1993.
- [38]K. Ebcioglu and T. Nakatani, "A New Compilation Technique for Parallelizing Loops with Unpredictable Branches on a VLIW Architecture," Languages and Compilers for Parallel Computing, D. Gelernter, A. Nicolau, and D. Padua, eds., chapter 12, pp. 213-229, Research Monographs in Parallel and Distributed Computing, London, U.K., and Cambridge, Mass.: Pitman Publishing and MIT Press, 1990. Selected papers from the Second Workshop on Languages and Compilers for Parallel Computing, Urbana, Ill., Aug. 1-3, 1989.
- [39]P. Feautrier, "Fine-Grain Scheduling Under Resource Constraints," Proc. Seventh Ann. Workshop Languages and Compilers for Parallel Computing, Ithaca, N.Y., Aug. 1994
- [40]C.-T. Hwang, J.-H. Lee, and Y.-C. Hsu, "A Formal Approach to the Scheduling Problem in High-Level Synthesis," IEEE Trans. Computer-Aided Design, vol. 10, pp. 464-475, Apr. 1991.

---

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 1045-9219/96\$05.00 © 1996 IEEE  
Vol. 7, No. 11: NOVEMBER 1996, pp. 1150-1163

Achieving Full Parallelism Using Multidimensional Retiming\*

Nelson Luiz Passos, Member, IEEE

Edwin Hsing-Mean Sha, Member, IEEE

Abstract<sup>3/4</sup>Most scientific and Digital Signal Processing (DSP) applications are recursive or iterative. Transformation techniques are usually applied to get optimal execution rates in parallel and/or pipeline systems. The retiming technique is a common and valuable transformation tool in one-dimensional problems, when loops are represented by data flow graphs (DFGs). In this paper, uniform nested loops are modeled as multidimensional data flow graphs (MDFGs). Full parallelism of the loop body, i.e., all nodes in the MDFG executed in parallel, substantially decreases the overall computation time. It is well known that, for one-dimensional DFGs, retiming can not always achieve full parallelism. Other existing optimization techniques for nested loops also can not always achieve full parallelism. This paper shows an important and counter-intuitive result, which proves that we can always obtain full-parallelism for MDFGs with more than one dimension. This result is obtained by transforming the MDFG into a new structure. The restructuring process is based on a multidimensional retiming technique. The theory and two algorithms to

obtain full parallelism are presented in this paper. Examples of optimization of nested loops and digital signal processing designs are shown to demonstrate the effectiveness of the algorithms.

Index Terms<sup>3/4</sup>Retiming, multidimensional data-flow graphs, instruction level parallelism, loop transformation, nested loops, VLIW, superscalar.

#### REFERENCES

- [1]A. Aiken, "Compaction Based Parallelization," PhD thesis, Technical Report 88-922, Cornell Univ., 1988.
- [2]A. Aiken and A. Nicolau, "Fine-Grain Parallelization and the Wavefront Method," Languages and Compilers for Parallel Computing, pp. 1-16, Cambridge, Mass.: MIT Press, 1990.
- [3]U. Banerjee, "Unimodular Transformations of Double Loops," Advances in Languages and Compilers for Parallel Processing, pp. 192-219, Cambridge, Mass: MIT Press, 1991.
- [4]L.-F. Chao and E.H.-M. Sha, "Static Scheduling of Uniform Nested Loops," Proc. Seventh Int'l Parallel Processing Symp., pp. 1,421-1,424, Newport Beach, Calif., Apr. 1993.
- [5]L.-F. Chao, "Scheduling and Behavioral Transformations for Parallel Systems," PhD dissertation, Princeton Univ., 1993.
- [6]L.-F. Chao, A. LaPaugh, and E.H.-M. Sha, "Rotation Scheduling: A Loop Pipelining Algorithm," Proc. 30th ACM/IEEE Design Automation Conf., pp. 566-572, Dallas, Tex., June 1993.
- [7]L.-F. Chao and E.H.-M. Sha, "Retiming and Unfolding Data-Flow Graphs," Proc. 1992 Int'l Conf. Parallel Processing, pp. 33-40, St. Charles, Ill., Aug. 1992.
- [8]L.-F. Chao and E.H.-M. Sha, "Unified Static Scheduling on Various Models," Proc. 1993 Int'l Conf. Parallel Processing, pp. 231-235, St. Charles, Ill., Aug. 1993.
- [9]E. Cohen and N. Megiddo, "Strongly Polynomial-Time and NC Algorithms for Detecting Cycles in Dynamic Graphs," Proc. 21st ACM Ann Symp. Theory of Computing, pp. 523-534, 1989.
- [10]R. Cytron, "Doacross: Beyond Vectorization for Multiprocessors". Proc. Int'l Conf. Parallel Processing, pp. 836-844, 1986.
- [11]D.E. Dudgeon and R.M. Mersereau, Multidimensional Digital Signal Processing. Englewood Cliffs, N.J.: Prentice Hall, 1984.
- [12]A. Fettweis and G. Nitsche, "Numerical Integration of Partial Differential Equations Using Principles of Multidimensional Wave Digital Filters," J. VLSI Signal Processing, vol. 3, pp. 7-24, 1991.
- [13]A. Fisher and B.R. Rau, "Instruction-Level Parallel Processing," Science, vol. 253, pp. 1,233-1,241, Sept. 1991.
- [14]G. Goosens, J. Wandewalle, and H. deMan, "Loop Optimization in Register Transfer Scheduling for DSP Systems," Proc. ACM/IEEE Design Automation Conf., pp. 826-831, 1989.
- [15]S. R. Kosaraju and G.F. Sullivan, "Detecting Cycles in Dynamic Graphs in Polynomial Time," Proc. 20th ACM Ann Symp. Theory of Computing, pp. 398-406, 1988.
- [16]S.Y. Kung, VLSI Array Processors. Englewood Cliffs, N.J.: Prentice Hall, 1988.
- [17]L. Lamport, "The Parallel Execution of DO Loops," Comm. ACM SIGPLAN, vol. 17, no. 2, pp. 82-93, Feb. 1974.
- [18]M. Lam, "Software Pipelining: An Effective Scheduling for VLIW Machines," ACM SIGPLAN Conf. Prog. Lang. Design and Implementation, pp. 318-328, 1988,.
- [19]T.-F. Lee, A.C.-H. Wu, D.D. Gajski, and Y.-L. Lin, "An Effective Methodology for Functional Pipelining," Proc. Int'l Conf. Computer Aided Design, pp. 230-233, Dec. 1992.
- [20]C.E. Leiserson and J.B. Saxe, "Retiming Synchronous Circuitry," Algorithmica, vol. 6, pp. 5-35, 1991.
- [21]D.I. Moldovan and J.A.B. Fortes, "Partitioning and Mapping Algorithms into Fixed Size Systolic Arrays," IEEE Trans. Computers, vol. 35, no. 1, pp. 1-12, Jan. 1986.
- [22]A. Nicolau, "Loop Quantization or Unwinding Done Right," Proc. 1987 ACM Int'l Conf. Supercomputing, Springer Verlag Lecture Notes on Computer Science, vol. 289, pp. 294-308, May 1987.
- [23]N. Park and A.C. Parker, "Sehwa: A Software Package for Synthesis of Pipelines from Behavioral Specifications," IEEE Trans. CAD, vol. 7, no. 3, pp. 356-370, 1988.
- [24]N.L. Passos, E.H.-M. Sha, and S.C. Bass, "Schedule-Based Multidimensional Retiming," to appear in Proc. Eighth Int'l Parallel Processing Symp., Cancun, MX, Apr., 1994.
- [25]N.L. Passos and E.H.-M. Sha, "Full Parallelism in Uniform Nested Loops Using Multidimensional Retiming," Proc. 23rd Int'l Conf. Parallel Processing, vol. 2, pp. 130-133, Aug. 1994.
- [26]N.L. Passos, E.H.-M. Sha, and S.C. Bass, "Loop Pipelining for Scheduling Multidimensional Systems Via Rotation," to appear in Proc. 31st Design Automation Conf., San Diego, Calif., June 1994.
- [27]N.L. Passos, E.H.-M. Sha, and S.C. Bass, "Partitioning and Retiming of Multidimensional Systems," to appear in Proc. IEEE Int'l Conf. Circuits and Systems, London, May 1994.



- [28]R. Potasman, J. Lis, A. Nicolau, and D. Gajski, "Percolation Based Scheduling," Proc. ACM/IEEE Design Automation Conf., pp. 444-449, 1990.
- [29]D.A. Schwartz, "Cyclo-Static Realizations, Loop Unrolling and CPM: Optimal Multiprocessor Scheduling," technical report, Georgia Inst. of Technology, School of Electrical Eng., 1987.
- [30]R. Tarjan, "Data Structures and Network Algorithms," SIAM, Philadelphia, Penn., 1983.
- [31]C.-Y. Wang and K.K. Parhi, "High Level DSP Synthesis Using the MARS Design System," Proc. Int'l Symp. Circuits and Systems, pp. 164-167, 1992.
- [32]M. Wolfe, "Loop Skewing: The Wavefront Method Revisited," Int'l J. Parallel Programming, vol. 15, no. 4, pp. 284-294, Aug. 1986.
- [33]M. Wolfe, Optimizing Supercompilers for Supercomputers. Cambridge, Mass.: MIT Press, 1989.
- [34]M.E. Wolf and M.S. Lam, "A Loop Transformation Theory and an Algorithm to Maximize Parallelism," IEEE Trans. Parallel and Distributed Systems, vol. 2, no. 4, pp. 452-471, Oct. 1991

---

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 1045-9219/96\$05.00 © 1996 IEEE  
Vol. 7, No. 11: NOVEMBER 1996, pp. 1164-1184

Decomposition Abstraction in Parallel Rule Languages\*

Shiow-Yang Wu

Daniel P. Miranker

James C. Browne

Abstract<sup>3</sup>/<sub>4</sub>Decomposition abstraction is the process of organizing and specifying decomposition strategies for the exploitation of parallelism available in an application. In this paper we develop and evaluate declarative primitives for rule-based programs that expand opportunities for parallel execution. These primitives make explicit, implicit relations among the data and similarly among the rules. The semantics of the primitives are presented in a general object-based framework such that they may be applied to most rule-based programming languages.

We show how the additional information provided by the decomposition primitives can be incorporated into a semantic-based dependency analysis technique. The resulting analysis reveals parallelism at compile time that is very difficult, if not impossible, to discover by traditional syntactic analysis techniques. Simulation results demonstrate scalable and broadly available parallelism.

Index Terms<sup>3</sup>/<sub>4</sub>Decomposition abstraction, parallel production systems, multiple rule firing systems, parallel decomposition, explicit parallel languages, decomposition mechanisms, set-oriented constructs, semantic-based dependency analysis, functional dependency.