

Research Accomplishments

Shantanu Dutt

Department of Electrical and Computer Engineering

Univ. of Illinois at Chicago

Phone: (312) 355-1314; Fax: (312) 996-6465

e-mail: dutt@ece.uic.edu; URL: <http://www.ece.uic.edu/~dutt>

My research areas include **VLSI CAD, FPGA testing and trust design, fault-tolerant computing and parallel processing**. We have also made a recent foray into **optimization**. Our work has received one *best paper* and one *most-influential paper awards*, one *featured speaker* recognition, and one *best paper nomination*, all at premier conferences. My research has been or is funded by NSF, DARPA and AFOSR, and companies like Intel and Xilinx. Highlights of my research contributions in the aforementioned areas are given below.

1 VLSI CAD

In VLSI-CAD I have worked in partitioning, placement, routing, (including incremental algorithms for the latter two), and logic and physical synthesis. I, along with my students, have made the following contributions in these areas.

1.1 Partitioning

1. In *swap-based partitioning* I developed an algorithm QuickCut that improves the complexity of the well-known Kernighan-Lin algorithm from $\Theta(n^2 \log n)$ to $\Theta(e \log n)$, and empirically shows runtime factor improvements of 5 to 50 [73]. Reducing the complexity of the Kernighan-Lin partitioning algorithm was a two-decade old open problem before the work in [73]. This work **found mention in the textbook** by Sabih H. Gerez, *Algorithms for VLSI Design Automation*, Wiley, 1998.
2. For *move-based partitioning* we have developed many novel and effective algorithms ranging from probability-based methods [13, 14, 65], to cluster-aware methods [10, 60, 64], to non-local information methods [14, 56], to methods for tackling constraints by intermediate relaxations [59], and to timing-driven partitioning [58]. All these techniques have been successful in transforming the local search nature of the basic iterative-improvement process in move-based partitioners to have more non-local optimality properties. These algorithms to date have among the best performance among *flat* partitioning methods. One of these works [65] earned a **best paper award** in 1996 at the prestigious *Design Automation Conference*.

1.2 Placement

1. A *placement* method SPADE for standard-cell VLSI circuits was created for wirelength optimization using the partition-driven paradigm and a number of novel concepts, chief of them being simultaneous-level partitioning and a logarithmically-graded balance-criterion as the partitioning proceeds hierarchically [53].
2. Novel techniques using analytical programming approaches and network-flow were developed for a *timing-driven (TD) incremental placement* method FlowPlace that can significantly improve critical path delays of wirelength-optimized placements (by up to 34%) and of timing-optimized placements (by up to 10%) with about a 9% deterioration in wirelength (WL) [38]; its runtime is about 12-18% of that for obtaining the original placements. Further, empirical evidence shows that FlowPlace's runtime grows

only linearly with circuit size, making our techniques very scalable. This paper was accorded a **featured speaker** recognition in the premier *International Conference on CAD (ICCAD)*, 2006. In [29], we extended FlowPlace by including WL cost (along with timing cost), based on a probabilistic HPBB metric, in network flow based detailed placement; this reduces WL deterioration to about 6% with only a 1.7% reduction in performance. We also prove in [29] that our white-space satisfaction technique (embedded in the network flow based detailed placer) can successfully yield valid placements with very high probability.

3. Effective and theoretically-robust algorithms have also been developed for *TD incremental placement under power constraints* [37] as well as *power-driven incremental placement under timing constraints* [32]. Results show that for power optimization, we can achieve average improvements of 12.1%, 10.8% and 9.1% with no delay constraint, 3% delay constraint and -3% delay constraint, respectively—a negative (positive) constraint signifies a metric (delay, in this case) improvement (deterioration) lower bound (upper bound). For delay optimization, we achieve average improvements of 16.8%, 11.6% and 9.1% under no constraints, 3% power constraint and -3% power constraint, respectively. I believe that our algorithms are significant advances in the state-of-the-art in placement algorithms for tackling both optimization and constraint metrics.

1.3 Routing

1. Two of the best academic *FPGA detailed routers* ROAD and ROAD-HOP were developed in [46, 48]. These techniques outperformed the previous state-of-the-art in important metrics. For example, ROAD is 13 times faster than VPR (the best flat router) and has the same quality of results (number of tracks used). ROAD is an optimal detailed router and incorporates optimality-preserving speedup methods that result in its efficacy and time efficiency. These works introduced concepts of learning-based search space pruning that can be applied to the solution of other combinatorial optimization problems using a depth-first search mechanism. A prime example is graph coloring that itself has many applications in computer engineering and science.
2. In *incremental routing*, we have introduced the concept of bump-and-refit. Incremental routing is used for engineering-change-order (ECO) applications and fault reconfiguration. Use of this novel concept has resulted in significantly better results in terms of routing completion rates, wire-lengths and via-usage than previous ripup-and-reroute approaches for both FPGA and ASICs [9, 44, 54]. Further novel concepts including that of Steiner-node slack tolerances were introduced in [41] to yield a near-wire-length optimal and guaranteed slack-satisfying *timing-driven incremental routing* method TIDE for ASICs that also obtains significant improvements over ripup-and-reroute approaches in the timing-driven context (e.g., 4-6 times fewer slack violations) while being about three times faster.

1.4 Logic and Physical Synthesis

1. In [36], we developed a network flow based timing-driven discrete cell-sizing algorithm that can incorporate total cell size constraints. We tested our algorithm on the ISCAS85 benchmark, and compared our results to an optimal solution produced by a dynamic programming method. The results for a 10% cell area increase constraint show that the improvement obtained by our method is only 1% worse (11.9% v.s. 12.9%) than the optimal solution, while being 60 times faster than it. A significant extension of our method uses network flow iteratively on primal-dual formulations (the dual formulation optimizes cell area of non-critical paths of the circuit under delay constraints and allocates the saved area to the primal problem of minimizing delay in critical paths under area constraints) [33]. We compared our technique to the timing-optimization variation of the state-of-the-art method of [Hu, et. al., DAC'07] and obtained 9% better timing results.

2. In [35], we proposed a post-placement physical synthesis algorithm, based on network flow, that can apply multiple circuit synthesis and placement transforms on a placed circuit to improve the critical path delay under area constraints by simultaneously considering the benefits and costs of all transforms (as opposed to considering them sequentially after applying each transform, as is done in most state-of-the-art methodologies). The circuit transforms we employed include (but are not limited to only these in our general technique), incremental placement, two types of buffer insertion, cell resizing and cell replication. We also tie the “transform selection network graph” to a “detailed placement network graph” with TD arc costs for cell movements. This enables our algorithms to perform both physical synthesis and detailed placement together, and thereby to incorporate the detailed placement cost for each synthesis transform along with the basic cost of applying the transform in the circuit. Results on three sets of benchmarks under 3-10% area increase constraints, show up to 48% and an average of 27.8% timing improvement. Our average improvement is relatively 40% better than applying the same set of transforms in a good sequential order that is used in many current techniques.

2 FPGA Testing and Trust Design

My students and I have developed several innovative and effective test and trust-design and verification techniques for FPGAs.

2.1 FPGA Trust Design and Verification

1. A novel trust design method for FPGA circuits that uses error-correcting code (ECC) structures for detecting design tampers—changes, deletion of existing logic, and addition of extra-design logic like Trojans—was proposed in [5]. We use two levels of randomization to thwart attempts by an adversary to discover the parity groups and inject tampers that mask each other and/or tamper with the testing circuit so that design tampers remain undetected: (a) randomization of the mapping of the ECC parity groups to the CLB (configuration logic block, i.e., logic cell) array; (b) randomization within each parity group of odd and even parities for different input combinations (classically, all ECC parity groups have even parities across all input combinations). These randomizations along with the error-detecting property of the underlying ECC lead to design tampers being uncovered with very high probabilities, as we show both analytically and empirically. Using the 2-D code as our underlying ECC and its 2-level randomization, our experiments with inserting 1-10 circuit CLB tampers and 1-5 extraneous logic CLBs in two medium-size circuits and a large RISC circuit implemented on a Xilinx Spartan-3 FPGA show very promising results of 100% tamper detection and 0% false alarms, obtained at a hardware overhead of only 7-10%.

2.2 FPGA Testing

1. We developed 1- and 2-diagnosable *built-in-self-testers* (BISTers) that achieve very high diagnostic coverages for high fault densities ($\approx 10\%$) that are expected to characterize permanent fault occurrences in future nano-scale CMOS and nanotechnology circuits [6, 45]. The 2-diagnosable BISTer design was the first time a diagnosability greater than one was achieved. The paper [45] was nominated for a **best paper award** in 2004 at the prestigious *Design Automation Conference*.
2. We proposed *probabilistic BIST techniques* using the novel concept of *iterative bootstrapping* that achieve far greater diagnostic coverage at not only high fault densities, but also for clustered faults, a pattern that occurs frequently for fabrication defects [43].
3. *Interconnect BIST techniques* were developed that can provably detect any number of interconnect faults as long as not all interconnects are faulty (a first), and that also have high diagnostic coverage [42].

4. We designed a methodology based on a formal analysis of iterative bootstrapping that addresses for the first time the problem of *detecting and diagnosing both interconnect and PLB (i.e., logic) faults* in FPGAs without making any assumptions of any component (interconnects, PLBs) being fault-free. Significantly improved diagnostic coverages and reduced false positives were achieved with this methodology compared to state-of-the-art BIST methods that erroneously make such fault-free assumptions [40].

3 Optimization

1. In [30] we proposed a new pivoting rule for the min-cost max-flow network Simplex method to determine the order of arc pivoting. In order to reduce the number of degenerate pivots (those that do not reduce the cost of the current solution), when choosing the pivoted-in arc, besides the standard reduced cost we also consider the probability that the resulting cycle is non-degenerate. A probability based reduced cost is devised to give priority to pivots that are likely to produce non-degenerate cycles. This technique can reduce the number of degenerate pivots by about 30%, and the total run time by 18% on average. However, this technique also causes an increase in the number of non-degenerate pivots, since some degenerate pivots are necessary steps for reaching non-degenerate cycles/pivots with large cost improvements. To address this issue, we developed the concept of necessary degenerate pivots and consider them for pivoting along with known and probabilistic non-degenerate pivots. This reduces the number of non-degenerate pivots (compared to not considering necessary degenerate pivots), helps in reaching negative cycles with large cost improvement, and ultimately reduces run time by an average of 29%.

4 Fault-Tolerant Computing

In fault-tolerant computing I, along with either my Ph.D. advisor or my students, have made the following contributions.

1. I have developed a range of novel and efficient methodologies for designing *fault-tolerant multi-processors* that include use of covering graphs and graph automorphisms, and a structural application of error correcting codes (ECCs) to yield multiprocessors with very high average fault tolerance [11, 17, 23, 24, 25, 27, 69, 71, 76, 77, 78, 79]. In 1995, one of these papers [79] (published in 1988) was awarded the recognition of a **most influential paper** published in the first 25 years, 1971-1995, of the premier conference on fault-tolerant computing, the *Fault Tolerant Computing Symp. (FTCS)*.
2. Novel mantissa based techniques were designed for significantly alleviating the well-known problem of round-off errors in *algorithm-based fault tolerance* techniques [2, 3, 20, 75].
3. The REMOD method for *concurrent testing and fault tolerance in arithmetic circuits* was developed that can accommodate any degree of fault tolerance desired, and has some of the lowest latency and hardware overheads [2, 19].
4. Very effective hardware and software techniques were designed for *fault tolerance in FPGAs* [2, 15, 54, 55, 61, 68].
5. Probably the first method for *off-chip control-flow-checking* of processors with on-chip caches [39].

5 Parallel Processing

Our (my students' and my) accomplishments in this area are:

1. The first *load-balancing method for irregular parallel computations*, QE, that has analytically-proved performance [22, 74]. QE empirically yields performance efficiency of 80-90% (speedup factor using P processors is $0.8P$ to $0.9P$; P is the ideal speedup) on large application problems like the Traveling Salesman Problem and Mixed Integer Programming on various large multicomputers like the nCUBE2 with 1024 processors. This is among the highest consistent speedup yielded by any general load-balancing method.
2. A low-overhead informed *randomized load-balancing* algorithm called Random Seeking that was shown theoretically and empirically to be more efficient than previous randomized load-balancing algorithms [12, 67]
3. The first *duplicate pruning strategies* for parallel best-first search that have provable scalability [18, 72].
4. An *adaptive load balancing* method QE* that adapts to node granularity and density of the application, and the communication latency of the multicomputer. This was the first adaptive load balancer of its kind (multiple dimensions of adaptivity) and it achieved near-ideal speedup on the IBM SP-2 multicomputer for Mixed Integer Programming problems [8, 57]
5. A very efficient *termination detection algorithm* for general parallel computations that achieves the best performance in several important metrics including the all-important one of detection latency for which it is optimal [7].
6. The above research in parallel processing and load balancing have **appeared in a major textbook**: V. Kumar, A. Grama, A. Gupta and G. Karypis, "Introduction to Parallel Computing: Design and Analysis of Algorithms," *Benjamin/Cummings Publishing Company*, Redwood City, CA, 1994, **and also appeared in the course**: Parallel/Distributed Artificial Intelligence Course, The University of Texas at Arlington, TX (<http://ranger.uta.edu/~cook/pai/pai.html>).
7. An useful *analysis of k-ary n-cube* multicomputer interconnection architectures on a wide class of real parallel algorithms (divide-and-conquer) [66], and the first of its kind. Previous analysis, while very useful and comprehensive, were for raw numerical message traffic and hypothetical message patterns.
8. NP-completeness proof for the *subcube allocation problem* and an effective algorithm for an approximate solution to this problem [26, 80].

References

[1] Book Chapters:

- [2] S. Dutt, F. Rota, F. Trovo and F. Hanchek, "Fault Tolerance in Computer Systems—From Circuits to Algorithms", invited article, in *Electrical Engineering Handbook*, Ed. Wai-Kai Chen, Academic Press, 2004.
- [3] S. Dutt and D. Boley, "Roundoff Errors", invited article, in *Wiley Encyclopedia of Electrical and Electronics Engineering*, Prof. John Webster, ed., Vol. 18, 1999, pp. 617-627.

[4] Journals:

- [5] S. Dutt and L. Li, "Trust-Based Design and Check of FPGA Circuits Using Two-Level Randomized ECC Structures, conditionally accepted (subject to minor revisions), *ACM Transaction on Reconfigurable Technology and Systems (TRETs)*, Special Issue on Security in Reconfigurable Systems Design , 2008.

- [6] S. Dutt, V. Verma and V. Suthar, "Built-in-Self-Test of FPGAs with Provable Diagnosabilities and High Diagnostic Coverage with Application to On-Line Testing", *IEEE Trans. Computer Aided Design of Integrated Circuits*, Feb. 2008, pp. 309-326.
- [7] N.R. Mahapatra and S. Dutt, "An efficient delay-optimal distributed termination detection algorithm", *Jour. Parallel and Distr. Computing*, vol. 67, 2007, pp. 1047-1066.
- [8] N.R. Mahapatra and S. Dutt, "Adaptive Quality Equalizing: High-Performance Load Balancing for Parallel Branch-and-Bound across Applications and Computing Systems", *Jour. of Parallel Computing*, June 2004.
- [9] S. Dutt, V. Verma and H. Arslan, "A Search-Based Bump-and-Refit Approach to Incremental Routing for ECO Applications in FPGAs", *ACM Trans. Design Automation of Electronic Systems (TODAES)*, 7(4), pp. 664-693, 2002.
- [10] S. Dutt and W. Deng, "VLSI Circuit Partitioning by Cluster-Removal Using Iterative Improvement Techniques", *ACM Trans. Design Automation of Electronic Systems*, Jan. 2002.
- [11] N.R. Mahapatra and S. Dutt, "Hardware-Efficient and Highly-Reconfigurable 4- and 2-Track Fault-Tolerant Designs for Mesh-Connected Arrays", *Jour. Parallel and Distr. Computing*, Vol. 61, No. 10, Oct 2001, pp. 1391-1411.
- [12] N. Mahapatra and S. Dutt, "Random Seeking: A General, Efficient and Informed Randomized Scheme for Dynamic Load Balancing", *Int. Jour. Foundations of Computer Science*, Special Issue on Randomized Computing, Vol. 11 No. 2, 2000, pp. 231-246.
- [13] S. Dutt and W. Deng, "Probability-Based Approaches to VLSI Circuit Partitioning", *IEEE Trans. CAD*, Vol. 19, No. 5, May 2000, pp. 534-549.
- [14] S. Dutt, H. Arslan and H. Theny, "Partitioning Using Second-Order Information and Stochastic-Gain Functions", *IEEE Trans. CAD*, Vol. 18, No. 4, April 1999, pp. 421-435.
- [15] F. Hanchek and S. Dutt, "Methodologies for Tolerating Logic and Interconnect Faults in FPGAs", *IEEE Trans. Computers*, Special Issue on Dependable Computing, Jan. 1998, pp. 15-33.
- [16] N. R. Mahapatra and S. Dutt, "Sequential and Parallel Branch-and-Bound Search Under Limited-Memory Constraint", *The IMA Volumes in Mathematics and its Applications, Parallel Processing of Discrete Problems*, Vol. 106, Panos, Pardalos (ed), Springer-Verlag New York, Inc. (1998), pp. 139-159.
- [17] S. Dutt and N.R. Mahapatra, "Node Covering, Error Correcting Codes and Multiprocessors with High Average Fault Tolerance", *IEEE Trans. Comput.*, Sept. 1997, pp. 997-1015.
- [18] N.R. Mahapatra and S. Dutt, "Scalable global and local hashing strategies for duplicate pruning in parallel A* graph search", *IEEE Trans. Parallel and Distr. Systems*, July 1997, pp. 738-756.
- [19] S. Dutt and F. Hanchek, "REMOD: A new hardware- and time-efficient methodology for designing fault-tolerant arithmetic circuits", *IEEE Trans. on VLSI Systems*, March 1997, pp. 34-56.
- [20] S. Dutt and F.T. Assaad, "Mantissa-preserving operations and robust algorithm-based fault tolerance for matrix computations", *IEEE Trans. Comput.*, Vol. 45, No. 4, April 1996, pp. 408-424.
- [21] N.R. Mahapatra and S. Dutt, "New anticipatory load balancing strategies for scalable parallel best-first search", *American Mathematical Society's DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, Vol. 22, 1995, pp. 197-232.
- [22] S. Dutt and N.R. Mahapatra, "Scalable load-balancing strategies for parallel A algorithms", Special Issue on Scalability of Parallel Algorithms and Architectures, *Journal of Parallel and Distr. Computing*, Vol. 22, No. 3, Sept. 1994, pp. 488-505.
- [23] S. Dutt and J.P. Hayes, "A local-sparing design methodology for fault-tolerant multiprocessors", Special Issue on Graph Theory in Computer Science and Other Fields, *Computers and Mathematics with Applications*, Volume 34, Issue 11, Pages 25-50, 1997, Elsevier Science.
- [24] S. Dutt and J.P. Hayes, "Some practical issues in the design of fault-tolerant multiprocessors", *IEEE Trans. Comput.*, Special Issue on Fault-Tolerant Computing, Vol. 41, May 1992, pp. 588-598.

- [25] S. Dutt and J.P. Hayes, "Designing fault-tolerant systems using automorphisms", *Journal of Parallel and Distr. Computing*, July 1991, pp. 249-268.
- [26] S. Dutt and J.P. Hayes, "Subcube allocation in hypercube computers", *IEEE Trans. Comput.*, Vol. 40, March 1991, pp. 341-352.
- [27] S. Dutt and J.P. Hayes, "On designing and reconfiguring k-fault-tolerant tree architectures", *IEEE Trans. Comput.*, Special issue on Fault-Tolerant Computing, Vol. 39, April 1990, pp. 490-503.
- [28] **Journal Papers Under Review:**
- [29] S. Dutt and H. Ren. "Discretized Network Flow Techniques for Timing and Wire-Length Driven Incremental Placement with High-Probability White-Space Satisfaction", under review at *IEEE Trans. of VLSI*, 2008. Available at www.ece.uic.edu/~dutt/papers/tvlsi-tdwlincrpl-submproof.pdf
- [30] H. Ren and S. Dutt, "Non-Degenerate Probabilities and Necessary Degenerate Pivots: New Concepts for Improved Pivoting Rules in the Network Simplex Algorithm", submitted to *Operations Research*. Available at www.ece.uic.edu/~dutt/papers/nw-speedup-or.pdf
- [31] **Journal Papers Under Preparation:**
- [32] H. Ren, and S. Dutt, "Incremental Placement Algorithms for Power Optimization under Timing Constraints", Technical report, UIC, April 2007 (to be submitted shortly to a journal). Available at www.ece.uic.edu/~dutt/papers/power-opt-trep.pdf
- [33] H. Ren and S. Dutt, "Network Flow Based Timing Driven Discrete Cell Sizing Using Primal-Dual Formulations". Available at www.ece.uic.edu/~dutt/papers/cellsizing-primal-dual.pdf
- [34] **Refereed Conference Papers:**
- [35] H. Ren, and S. Dutt, "Algorithms for Simultaneous Consideration of Multiple Physical Synthesis Transforms for Timing Closure", accepted for publication, *Proc. IEEE Int'l Conf. CAD (ICCAD)*, Nov. 2008.
- [36] H. Ren, and S. Dutt, "A Network-Flow Based Cell Sizing Algorithm", *17th International Workshop on Logic & Synthesis*, 2008 (regular presentation), pp. 7-14.
- [37] Incremental Placement with Application to Performance Optimization under Power Constraints", *Proc. IEEE Int'l. Conf. on Computer Design*, 2007, pp. 251-258.
- [38] S. Dutt, H. Ren, F. Yuan and V. Suthar, "A Network-Flow Approach to Timing-Driven Incremental Placement for ASICs", *Proc. IEEE Int'l Conf. CAD (ICCAD)*, Nov. 2006, pp. 375-382.
- [39] F. Rota, S. Krishna and S. Dutt, "Off-Chip Control Flow Checking of On-Chip Processor-Cache Instruction Stream", *Proc. 21'st IEEE Int'l Symp. on Defect and Fault Tolerance in VLSI Systems (DFT)*, Oct. 2006, pp. 507-515.
- [40] V. Suthar and S. Dutt, "Mixed PLB and Interconnect BIST for FPGAs without Fault-Free Assumptions", in *Proc. IEEE VLSI Test Symposium (VTS)*, April 2006, pp. 36-43.
- [41] S. Dutt and H. Arslan, "Efficient Timing-Driven Incremental Routing for VLSI Circuits Using DFS and Localized Slack-Satisfaction Computations," *Proc. Design Automation and Test in Europe (DATE)*, March 2006, pp. 768-773.
- [42] V. Suthar and S. Dutt, "Efficient On-line Interconnect Testing in FPGAs with Provable Detectability for Multiple Faults", *Proc. Design Automation and Test in Europe (DATE)*, March 2006, pp. 1165-1170.

- [43] V. Suthar and S. Dutt, "High-Diagnosability Online Built-In Self-Test of FPGAs via Iterative Bootstrapping", *Proc. ACM Int'l Great Lakes Symp. on VLSI*, April 2005.
- [44] H. Arslan and S. Dutt, "A Depth-First-Search Controlled Gridless Incremental Routing Algorithm for VLSI Circuits", *Proc. IEEE Int'l. Conf. on Computer Design (ICCD)*, Oct. 2004, pp. 86-92.
- [45] V. Verma, S. Dutt and V. Suthar, "Efficient On-Line Testing of FPGAs with Provable Diagnosabilities", *Proc. IEEE/ACM Design Automation Conference*, June 2004, pp. 498-503.
Nominated for a Best Paper Award.
- [46] H. Arslan and S. Dutt, "An Effective Hop-Based Detailed Router for FPGAs for Optimizing Track Usage and Circuit Performance", *Proc. ACM Int'l Great Lakes Symp. on VLSI*, April 2004, pp. 208-213.
- [47] V. Verma and S. Dutt, "Roving Testing Using Built-in-Self-Tester Designs for FPGAs with Effective Diagnosability" (poster paper), *ACM Int'l Symp. on Field Programmable Gate Arrays*, Feb. 2004.
- [48] H. Arslan and S. Dutt, "ROAD: An Order-Insensitive Optimal Detailed Router for FPGAs", *Proc. IEEE Int'l. Conf. on Computer Design*, May 2003, pp. 350-356.
- [49] F. Trovo, S. Dutt and H. Arslan, "Design and Simulation of an EM-Fault-Tolerant Processor with Micro-Rollback, Control-Flow Checking and ECC", *IEEE APS/URSI International Symposium*, (digest of abstracts), June 2003.
- [50] K. Zhong and S. Dutt, "Algorithms for Simultaneous Satisfaction of Multiple Constraints and Objective Optimization in a Placement Flow with Application to Congestion Control", *Proc. Design Automation Conference*, June 2002, pp. 854-859.
- [51] S. Dutt and H. Arslan, "Evaluation of Processor Faults Due to EM Interference—Concepts and Simulation Environment", *National Radio Science Meeting*, (no proceedings), Jan. 2002.
- [52] V. Verma and S. Dutt, "A Search-Based Bump-and-Refit Approach to Incremental Routing for ECO Applications in ", *Proc. IEEE Int. Conf. Comput.-Aided Design*, Nov. 2001, pp. 144-151.
- [53] K. Zhong and S. Dutt, "Effective Partition-Driven Placement with Simultaneous Level Processing and Global Net Views", *Proc. IEEE Int. Conf. Comput.-Aided Design*, pp. 254-259, Nov. 2000.
- [54] S. Dutt, V. Shanmugavel and S. Trimberger, "Efficient Incremental Rerouting for Fault Reconfiguration in Field Programmable Gate Arrays", *Proc. IEEE Int. Conf. Comput.-Aided Design*, pp. 173-176, Nov. 1999.
- [55] N. R. Mahapatra and S. Dutt, "Efficient Network-Flow Based Techniques for Dynamic Fault Reconfiguration in FPGAs", *Proc. 29th Annual International Symposium on Fault-Tolerant Computing (FTCS-29)*, June 1999, pp. 122-129.
- [56] S. Dutt and H. They, "Partitioning Using Second-Order Information and Stochastic-Gain Functions", *Proc. ACM Int'l Symp. on Physical Design*, April 1998, pp. 112-117.
- [57] N.R. Mahapatra and S. Dutt, "Adaptive Quality Equalizing: High-Performance Load Balancing for Parallel Branch-and-Bound Across Applications and Computing Systems", *Proc. Joint IEEE Parallel Processing Symposium/ Symp. on Parallel and Distr. Processing*, April 1998.
- [58] S. Dutt, "A Stochastic Approach to Timing-Driven Partitioning and Placement with Accurate Net and Gain Modeling", *TAU97: IEEE/ACM Int. Workshop on Timing Issues in Digital Systems*, Dec. 1997, pp. 246-256.
- [59] S. Dutt and H. They, "Partitioning Around Roadblocks: Tackling Constraints with Intermediate Relaxations", *IEEE/ACM International Conference on CAD*, Nov., 1997, pp. 349-355.
- [60] S. Dutt and W. Deng, "VLSI Circuit Partitioning by Cluster-Removal Using Iterative Improvement Techniques", *Proc. IEEE/ACM International Conference on CAD*, Nov. 1996.
- [61] F. Hanchek and S. Dutt, "Design Methodologies for Tolerating Cell and Interconnect Faults in FPGAs", *Proc. Int. Conf. on Computer Design*, Oct. 1996.
- [62] N.R. Mahapatra and S. Dutt, "Hardware-Efficient and Highly-Reconfigurable 4- and 2-Track Fault-Tolerant Designs for Mesh-Connected Processor Arrays", *Proc. Fault-Tolerant Computing Symp.*, June 1996, pp. 272-281.

- [63] N.R. Mahapatra and S. Dutt, “Sequential and parallel branch-and-bound search under limited-memory constraints”, in *Proc. Parallel Optimization Colloquium, Versailles, France*, March 1996, pp. 147-166.
- [64] S. Dutt and W. Deng, “VLSI Circuit Partitioning by Cluster-Removal Using Iterative Improvement Techniques”, *Proc. Physical Design Workshop*, April 1996, pp. 92-99.
- [65] S. Dutt and W. Deng, “A probability-based approach to VLSI circuit partitioning”, *Proc. Design Automation Conference*, June 1996, pp. 100-105.
Best-Paper Award.
- [66] S. Dutt and N.R. Trinh, “Are There Advantages to High-Dimension Architectures?: Analysis of k-ary n-cubes for the Class of Parallel Divide-and-Conquer Algorithms”, *Proc. International Conf. on Supercomputing*, May 1996, pp. 398-406.
- [67] N.R. Mahapatra and S. Dutt, “Random Seeking: A General, Efficient, and Informed Randomized Scheme for Dynamic Load Balancing”, *Proc. Tenth IEEE Parallel Processing Symposium*, April 1996, pp. 881-885.
- [68] F. Hanchek and S. Dutt, “Node-covering based defect and fault tolerance methods for increased yield in FPGAs”, *Proc. International Conference on VLSI Design*, Jan. 1996, pp. 225-229.
- [69] S. Dutt and N.R. Mahapatra, “Node Covering, Error Correcting Codes and Multiprocessors with High Average Fault Tolerance”, in *Proc. Fault-Tolerant Computing Symp.*, June 1995, pp. 320-329.
- [70] N.R. Mahapatra and S. Dutt, “New anticipatory load balancing strategies for scalable parallel best-first search”, *DIMACS workshop on Parallel Processing of Discrete Optimization Problems*, (informal proceedings), April 1994.
Invited Paper.
- [71] S. Dutt, “Fast polylog-time reconfiguration of structurally fault-tolerant multiprocessors”, *Proc. Fifth IEEE Symposium on Parallel and Distr. Processing*, Dec. 1993, pp. 762-770.
- [72] N.R. Mahapatra and S. Dutt, “Scalable duplicate-pruning strategies for parallel A graph search”, *Proc. Fifth IEEE Symposium on Parallel and Distr. Processing*, Dec. 1993, pp. 290-297.
- [73] S. Dutt, “New faster Kernighan-Lin-type graph-partitioning algorithms”, *Proc. IEEE/ACM International Conference on CAD*, Nov. 1993.
- [74] S. Dutt and N.R. Mahapatra, “Parallel A algorithms and their performance on hypercube multiprocessors”, *Proc. Seventh IEEE Parallel Processing Symposium*, 1993, pp. 797-803.
- [75] F.T. Assaad and S. Dutt, “More robust tests in algorithm-based fault-tolerant matrix multiplication”, *Proc. The Twenty-Second Fault-Tolerant Computing Symp.*, July 1992, Boston, pp. 430-439.
- [76] S. Dutt and J.P. Hayes, “Some practical issues in the design of fault-tolerant multiprocessors”, *Proc. Twenty-First Fault Tolerant Computing Symp.*, June 1991, Montreal, Canada, pp. 292-299.
- [77] S. Dutt and J.P. Hayes, “An automorphic approach to the design of fault-tolerant multiprocessors”, *Proc. Nineteenth Fault Tolerant Comput. Symp.*, June 1989, Chicago, pp. 496-503.
- [78] S. Dutt and J.P. Hayes, “On designing fault-tolerant multiprocessor systems”, *International Workshop on Hardware Fault Tolerance in Multiprocessors*, June 1989, Urbana-Champaign, pp. 48-51.
- [79] S. Dutt and J.P. Hayes, “Design and reconfiguration strategies for near-optimal k-fault-tolerant tree architectures”, *Proc. Eighteenth Fault Tolerant Comput. Symp.*, June 1988, Tokyo, pp. 328-333;
A Most Influential Paper award for the first 25 years of FTCS (1971-1995). Has reappeared in “Highlights from 25 Years—FTCS-25 Silver Jubilee”, IEEE Computer Society Press, pp. 68-73.
- [80] S. Dutt and J.P. Hayes, “On allocating subcubes in a hypercube multiprocessor”, *Proc. Third Conf. on Hypercube Computers*, Jan. 1988, pp. 801-810.