Reprint from IEEE Proc. of Int. Symp. on Circuits and Systems (ISCAS'99), May 30 -- June 2, 1999

# Design of Experiments in CAD:
# Context and New Data Sets for ISCAS'99

Franc Brglez
CBL, Dept. of Computer Science,
NC State University, Box 7550
Raleigh, NC 27695, USA
http://www.cbl.ncsu.edu/~brglez/

Rolf Drechsler
Institute of Computer Science,
Albert-Ludwigs-University
79110 Freiburg im Breisgau, Germany
http://www.informatik.uni-freiburg.de/~drechsle/

**Abstract** – *This paper introduces the background and motivation for the two special sessions at ISCAS'99. The sessions bring together eight papers, each rooted in the methodology of experimental design, and contributed by collaborating teams of distributed participants. The paper briefly outlines the premises of the companion papers that follow, provides a brief description of a typical experimental design, and introduces a design for archival of data sets and results that are to be readily accessible on the Web.*

**Keywords:** design of experiments, circuit equivalence classes, benchmarking.

## 1 Introduction

More than a thousand mathematical problems arising in engineering and science have been shown to be NP-hard. Data sets such as ISCAS'85, ISCAS'89, and extensions introduced at logic and layout synthesis workshops, e.g. [1, 2, 3], are representative of such problems. They have captured a large following of researchers in CAD that 'benchmark' their algorithms on various subsets of these data sets – as will undoubtedly the newer, larger benchmarks that are beginning to emerge [4, 5].

There is no shortage of published claims of 'incremental improvements' with a particular heuristic. However, such claims may well be subject to undocumented experimental errors, and have not been supported by a rigorous statistical analysis, including a test of hypothesis such as 'Is the improvement due to the choice of the algorithm or due to chance?', as illustrated in [6]. We argue that just as the evolution of new test cases is important, so is the evolution of the Experimental Design as a discipline for the core algorithms in CAD.

In fairness to researchers, the data sets as currently formulated, are not suitable for the 'design of experiments' as practiced in disciplines such as yield optimization in manufacturing and agriculture, bio-medical research, and others. Two fundamental principles of experimental design are randomization and replication: 'subjects' are drawn at random from an 'eligible class', and 'treated'. This discipline emerged from contributions pioneered, for several decades, by R. A. Fisher [7]. In bio-medicine, the class may have properties such as 'same species, same sex, same age, same weight'. Rigorous statistical methods are applied to analyze the observations reported for each treatment, e.g. [8].

A number of approaches that *can* yield equivalence classes with some common properties have been introduced recently, each with the objective to test the performance of algorithms in a variety of settings [9, 10, 11, 12, 13, 14, 15, 16]. Experiments with several types of algorithms, ranging from logic synthesis and verification to partitioning and placement, have demonstrated that the performance of heuristics can also be highly sensitive to both the *ordering and labeling* conventions of the circuit class representation [15, 17, 18, 19, 20]. This makes the netlist isomorphism class an important reference class for all equivalence classes: depending on the quality of heuristics that optimizes the given cost function $\zeta$, the class can induce variance in its distribution that is significant. Defining the two extreme points in this distribution as $\zeta_{min}$ and $\zeta_{max}$, the *relative range* of the distribution can be defined as

$$\rho = (\zeta_{max} - \zeta_{min})/\zeta_{min} \quad (1)$$

It is easy to find a netlist isomorphism class and a heuristic that induces $\rho > 0$; instances with $\rho >> 1$ have been recorded. Two researchers, reporting on an experiment with a single instance of the same netlist, each storing the netlist in a different internal order, can thus report results that differ by more than 100% – even if they both apply the same algorithm to the same circuit instance! The earlier experiments with relatively small isomorphism classes of netlists were reported in [21].

We next outline the premises of the companion papers, provide a brief description of a typical experimental design, and introduce a design for archives of data sets and results that are to be readily accessible on the Web.

## 2 Background and Motivation

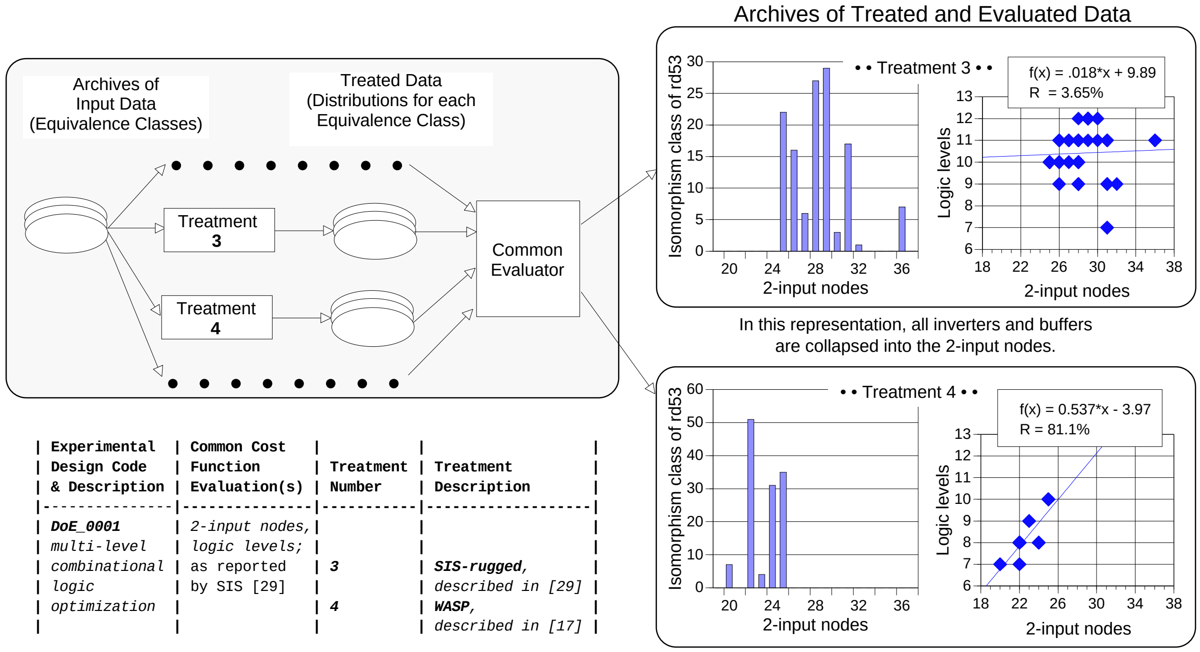
This section introduces the background and motivation for each of the companion papers in the two special sessions. The shared theme of each paper is the experimental design based on the principles illustrated in Figure 1 and explained in more details in the next section.

Paper 1.2 [22] analyzes 'practical netlist graphs' and introduces a netlist characterization that forms the basis on which an equivalence class of related 'clone circuits' can be generated, in particular for experiments exploring architectural issues in FPGAs, and physical design in general.

Paper 1.3 [23] addresses a similar problem as the preceding paper. It introduces a canonical k-partite form for a graph as its signature and a set of well-defined wiring perturbations that induce signature-invariant wiring mutations, giving rise to equivalence classes of 'circuit mutants'.

Paper 1.4 [24] introduces isomorphism classes of Boolean functions and analyzes exact and heuristic multi-level minimization of such functions. Significantly, even for the isomorphism classes of 4-to-8 input boolean functions, heuris-

Franc Brglez was supported by contracts from the Semiconductor Research Corporation (94–DJ–553), SEMATECH (94–DJ–800), and DARPA/ARO (P–3316–EL/DAAH04–94–G–2080 and DAAG55–97–1–0345).




| Experimental Design Code & Description | Common Cost Function Evaluation(s) | Treatment Number | Treatment Description |
|---|---|---|---|
| ***DoE_0001*** *multi-level combinational logic optimization* | *2-input nodes, logic levels;* as reported by SIS [29] | ***3*** | ***SIS-rugged,*** *described in [29]* |
| | | ***4*** | ***WASP,*** *described in [17]* |

Fig. 1. An experimental design flow and treatment comparisons for 128 instances from the isomorphism class of rd53.

tic minimization can have large variance – and is thus far from optimal.

Paper 2.1 [25] introduces user-configurable workflow toolkits that have the potential to make scientific and collaborative planning, executing, reporting, and archiving of experiments with core CAD algorithms a routine rather than the exception.

Paper 2.2 [26] reports new results of experiments on the bigraph crossing minimization. This NP-hard problem has important applications in area and wire length reduction in VLSI circuits as well as drawing of graphs.

Paper 2.3 [27] applies the methodology of using equivalence classes of 'clone circuits' to evaluation of algorithms in physical design, with illustrative experiments using several netlist partitioners.

Paper 2.4 [28] examines the performance of BDD variable ordering algorithms, using the equivalence classes on two versions of BDD packages – and drawing inferences on statistical significance of differences between the two releases.

## 3 Design of Experiments

An experimental design involves three basic steps: (1) selection of an equivalence class of experimental subjects, eligible for treatments, (2) application of one or more treatment to the same class, and (3) statistical evaluation of each treatment effectiveness. In CAD, we can consider 'treatments' as algorithms applied to data equivalence classes such as netlists (or hypergraphs) that have a number of properties in common: the same or 'nearly-the-same' size, distributions of fanins and fanouts, Boolean functionality, Boolean entropy, etc. The simplest to formulate is the netlist isomorphism class $\mathcal{N}_{\rm iso}$: (1) take a reference netlist represented as a hypergraph $G_r(V, E)$, (2) apply, uniformly to all nodes in $G_r$, a *random re-order, and random re-label* procedure $rr(V)$:

$$\mathcal{N}_{\rm iso} = \{N_j \in G_r(rr(V), E)\} \qquad (2)$$

Releative to all other instances in $\mathcal{N}_{\rm iso}$, each instance $N_j$ has the following properties:

*P1:* the order of nodes in $N_j$ is uniformly random;

*P2:* the labels of nodes in $N_j$ are uniformly random.

Properties *(P1, P2)* are essential to good experimental design and must be maintained universally for all equivalence classes, not just the isomorphism class. The purpose of *P1* is clear. Without *P2*, programs that rely on hashing input data may *unknowingly undo* the randomization of input presentations and thus confound the experiments. Important lessons have been learned already [15, 18], and are reported again in one of the session's papers [28].

An illustration of the experimental design flow just described, and a summary of experimental results with the netlist isomorphism class of **rd53** [3], is shown in Figure 1. The reference circuit **rd53_ref** consists of 68 2-input nodes on 10 levels. In this experiment, we created an isomorphism class of 128 instances and submitted them to two treatments (algorithms): labeled as Treatment 3 (SIS-rugged [29]) and Treatment 4 (WASP [17]). The primary objective of each treatment is to reduce the number of 2-input logic nodes while maintaining the logic-invariance of the netlist.

As shown in the histograms, the two algorithms perform very differently – *while both exhibit sensitivity to the initial order of nodes* in the netlist!! Treatment 3 returns netlists that range from 25 to 35 nodes, with levels that range from 7 to 12. Treatment 4 returns the netlists that range from 20 to 25, with levels in the 7–10 range. Either a visual or a $t-$test [8] can declare the differences in distributions as statistically significant. Moreover, the *correlations* of minimized logic nodes to logic levels are very different for the two algorithms: there apparently is no correlation when applying Treatment 3, while the correlation is significant when applying Treatment 4 (81.1%). Experiments with other treatments reveal even larger spreads in the respective distributions: 38–42 nodes, 8–8 levels for Treatment 1 (SIS-algebraic [29]), and 29–49 nodes, 9–16 levels for Treatment 2 (SIS-boolean [29]).

Treatment results for a much larger isomorphism class, based on **c1355_ref** [3], are shown in Figure 2. The ref-

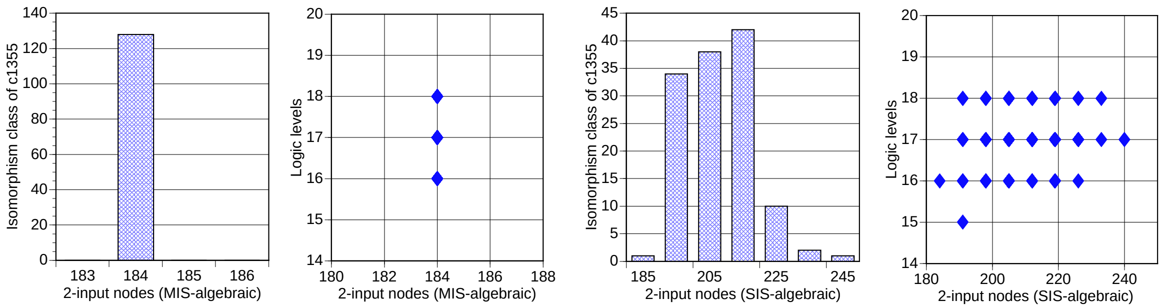



Fig. 2. Comparisons of two treatments applied to 128 instances from the isomorphism class of c1355.

erence circuit **c1355_ref** consists of 518 2-input nodes on 25 levels. Two treatments, applied to an isomorphims class of 128 reveal surprising distributions: 184–184 nodes, 16–18 levels for a treatment based in MIS-algebraic [30], and 184–245 nodes, 15–18 levels for a treatment based in SIS-algebraic [29]. This experiment points out the merits of formalizing the experimental design: older, more mature algorithms, may in fact be more stable and perform better than the more recently introduced algorithms. Clearly, more experimentation is required to make any conclusive statements on the performance of these and any other algorithms.

Design and execution of experiments on the scale outlined in this section cannot be carried out without relying on automated execution and careful design of archival data structures. These are briefly discussed next.

## 4 DoE Archives on the Web

The flow in Figure 1 can be automated for a *distributed* Design of Experiments (DoE) in CAD. An approach and an implementation are described in [25].

The unique feature of the proposed experimental design is the separation of the experiment into three parts: (1) the generation of experimental data (e.g. programs that generate equivalence classes based on actual circuits), (2) the execution of heuristics, and (3) the evaluation of the cost function for the results produced by a heuristic. This separation of execution and evaluation makes it possible for other researchers to run their own heuristics on posted data, submit new data for others, and evaluate any posted results independently with evaluators that are readily available. We next discuss a number of experimental design classifications, point to the respective evaluator(s), and relate each experiment to the companion papers.

The current experiments have been classified under a common archival directory on the Web under `http://www.cbl.ncsu.edu/experiments/DoE_Archives/`. The results of each experiment class are posted in the manner so they can be re-evaluated independently by evaluator tools that are freely accessible from the Web. The following classes of experiments will have been posted and evaluated by the time of ISCAS'99 presentations in June:

*DoE_0001:* Multi-level combinational logic optimization. The cost functions being evaluated are total 2-input nodes and maximum logic levels. The evaluator used in this process is `SIS` [29]. The latest additions to this directory are discussed in [24].

*DoE_0002:* Variable ordering optimization to construct minimum-size BDDs of combinational circuits. The cost function being minimized is the total number of MDD nodes. The evaluator used in this process is `VIS` [31]. The latest additions to this directory are discussed in [28].

```
DoE_0001
  \
  |_ circuitClasses
          \
          |_c1355_FX
          |_c1355_WD
          ...........
          |_rd53_FX
          |_rd53_WD
              \
              |_ 0001.blif
              |_ 0002.blif
              |_ 0003.blif
              .........
              .........
  |_ referenceCircuits
          \
          |_ c1355_ref.blif
              .......
          |_ rd53_ref.blif
          .......
  |_ treatmentComparisons
  |_ treatment_0001_MIS_alg
  |_ treatment_0001_SIS_alg
  ......
  |_ treatment_0003_SIS_rug
  |_ treatment_0004_WASP
  |_ treatment_0005_EXACT

treatment_0001_SIS_alg
  \
  |_ c1355_allClasses
  .......
  |_ rd53_allClasses
        \
        |_ allClasses_level.sum1
        |_ allClasses_level.table
        |_ allClasses_node.sum1
        |_ allClasses_node.table
        |_ FX_circuits
        |_ FX.sum1
        |_ FX.table
        |_ WD_circuits
            \
            |_0001.blif
            |_0002.blif
            |_0003.blif
            .......
            .......
        |_ WD.sum1
        |_ WD.table
```

**file: treatment_0001_SIS_alg/c1355_allClasses/WD.sum1**
Sample sizes per column: 128 128

| Column Parameter | Population Mean (95% conf. int.) | Sample Mean | Sample Std. Dev. | Sample Min. | Sample Max. |
|---|---|---|---|---|---|
| Nodes | [206.2, 210.0] | 208.1 | 10.75 | 184 | 240 |
| Levels | [ 16.8, 17.1] | 16.9 | 0.67 | 15 | 18 |

Program name: /cblopt/cblsw/bin/quick_stats.pl

Fig. 3. Experiments directory for DoE_0001 under http://www.cbl.ncsu.edu/experiments/DoE_Archives/ and an example of a statistical summary of results based on the equivalence class c1355_WD

*DoE_0003:* Bigraph embedding optimization. The cost function being optimized is the edge crossing number. The evaluator used in this process is `cn_eval` [20]. The latest additions to this directory are discussed in [26].

*DoE_0004:* Linear cell placement optimimization. The evaluators used in this process are `hn_eval` [19], `OASIS` [32], and `VPR` [33]. The latest additions to this directory are discussed in [19].

*DoE_0005:* Experiments in netlist bipartitioning using equivalence classes of circuit clones, as introduced in [27].

The archive, such as *DoE_0001* outlined in Figure 3, is representative of the directory structure that is maintained for each experimental design. The major directories include *referenceCircuits*, used to generate *circuitClasses*, followed by the directories of all treatment archives. The directory *treatment_0001_SIS_alg* is typical: it contains not only 'raw tables' of results but also statistical summaries for *all equivalence classes.* In addition, for each instance from say *rd53_WD* (an isomorphism class under *referenceCircuits*), we have the corresponding directory *WD_circuits*, generated by *treatment_0001_SIS_alg* for the WD (isomorphism) class. It is thus possible to relate properties of each input instance to corresponding instances of treatment-specific solutions. Solutions stored for other types of experimental designs may include variable orders *(DoE_0002)*, graph embeddings *(DoE_0003)*, etc.

For more details about the organizational structure of any experiment, see the postings on the Web.

ACKNOWLEDGMENTS. A number of netlist equivalence classes and experiments, now archived under *DoE_0001*, were contributed by D. Ghosh. Experiments with WASP on the class of rd53_WD were contributed by A. Žemva.

REFERENCES

[1] F. Brglez and H. Fujiwara. Special Session on ATPG (Also introducing 'A Neutral Netlist of 10 Combinational Benchmark Circuits'). In *IEEE 1985 International Symposium on Circuits and Systems – ISCAS85*, 1985. Now a benchmark directory ISCAS85 at http://www.cbl.ncsu.edu/benchmarks/Benchmarks-upto-1996.html.

[2] F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *IEEE 1989 International Symposium on Circuits and Systems – ISCAS89*, pages 1924–1934, May 1989. Now a benchmark directory ISCAS89 at http://www.cbl.ncsu.edu/-benchmarks/Benchmarks-upto-1996.html.

[3] S. Yang. Logic Synthesis and Optimization Benchmarks User Guide. Technical report, MCNC, Research Triangle Park, NC, January 1991. This report is now available from http://www.cbl.ncsu.edu/-publications/#1991-IWLS-UG-Saeyang and the benchmarks from http://www.cbl.ncsu.edu/benchmarks/Benchmarks-upto-1996.html.

[4] C. Alpert. The ISPD98 Circuit Benchmark Suite. In *ACM International Symposium on Physical Design*, April 1998. Also available, with the benchmark directory, from http://nexus6.cs.ucla.edu/-~cheese/ispd98.html.

[5] ITC99 Benchmark Home Page, 1999. For more information, see http://www.cerc.utexas.edu/itc99-benchmarks/bench.html.

[6] F. Brglez. Design of Experiments to Evaluate CAD Algorithms: Which Improvements Are Due to Improved Heuristic and Which Are Merely Due to Chance? Technical Report 1998-TR@CBL-04-Brglez, CBL, CS Dept., NCSU, Box 7550, Raleigh, NC 27695, April 1998. Also available at http://www.cbl.ncsu.edu/publications/.

[7] R. A. Fisher. *Statistical Methods, Experimental Design, and Scientific Inference*. Oxford University Press, 1993. Reprinted, with corrections, from earlier versions, 1925-1973.

[8] G. E. P. Box, W. G. Hunter, and J. S. Hunter. *Statistics for experimenters: An Introduction to Design, Data Analysis, and Model Building*. John Wiley & Sons, 1978.

[9] K. Iwama and K. Hino. Random generation of test instances for logic optimizers. In *Proceedings of the 31st ACM IEEE Design Automation Conference*, pages 430–434, 1994.

[10] J. Darnauer and W. Dai. A Method for Generating Random Circuits and its Application to Routability Measurement. In *4th ACM/SIGDA Int'l Symp. on FPGAs, FPGA96*, pages 66–72, February 1996.

[11] Michael Hutton, J.P. Grossman, J. Rose and D. Corneil. Characterization and Parametrized Random Generation of Digital Circuits. In *Proceedings of the 33st ACM IEEE Design Automation Conference*, June 1996.

[12] Michael Hutton, J.P. Grossman, J. Rose and D. Corneil. Generation of Synthetic Sequential Benchmark Circuits. In *ACM Symposium on FPGAs*, pages 149–155, February 1997. Available from http://www.eecg.toronto.edu/~/pubs/pubs.html.

[13] Michael Hutton, J.P. Grossman, J. Rose and D. Corneil. Characterization and Parameterized Random Generation of Combinational Benchmark Circuits. *IEEE Trans. Computer–Aided Design*, 1999. To appear. Postscript available from http://www.eecg.toronto.edu/~jayar/-pubs/pubs.html.

[14] N. Kapur, D. Ghosh, and F. Brglez. Towards A New Benchmarking Paradigm in EDA: Analysis of Equivalence Class Mutant Circuit Distributions. In *ACM International Symposium on Physical Design*, April 1997. Also available at http://www.cbl.ncsu.edu/publications/-#1998-ISPD-Kapur.

[15] D. Ghosh, N. Kapur, J. E. Harlow, and F. Brglez. Synthesis of Wiring Signature-Invariant Equivalence Class Circuit Mutants and Applications to Benchmarking. In *Proceedings, Design Automation and Test in Europe*, pages 656–663, Feb 1998. Also available at http://www.cbl.ncsu.edu/publications/#1998-DATE-Ghosh.

[16] J. E. Harlow and F. Brglez. Design of Experiments for Evaluation of BDD Packages Using Controlled Circuit Mutations. In *Proceedings of the International Conference on Formal Methods in Computer-Aided Design (FMCAD'98)*. Springer Verlag, Lecture Notes in Computer Science, November 1998. Also available from http://www.cbl.ncsu.edu/-publications/#1998-FMCAD-Harlow.

[17] A. Žemva, F. Brglez, and B. Zajc. Multi-level Logic Optimization Based on Wave Synthesis of Permissible Mutation Functions (WASP). In *International Workshop on Logic Synthesis (IWLS'98)*, June 1998. Also available from http://www.cbl.ncsu.edu/publications/-#1998-IWLS-Zemva.

[18] J. E. Harlow and F. Brglez. Design of Experiments in BDD Variable Ordering: Lessons Learned. In *Proceedings of the International Conference on Computer Aided Design*. ACM, November 1998. Also available from http://www.cbl.ncsu.edu/publications/#1998-ICCAD-Harlow.

[19] D. Ghosh, F. Brglez, and M. Stallmann. Hypercrossing Number: A New and Effective Cost Function for Cell Placement Optimization. Technical Report 1998-TR@CBL-12-Ghosh, CBL, CS Dept., NCSU, Box 7550, Raleigh, NC 27695, December 1998. Also available at http://www.cbl.ncsu.edu/publications/#1998-TR@CBL-12-Ghosh.

[20] M. Stallmann, F. Brglez, and D. Ghosh. Heuristics and Experimental Design for Bigraph Crossing Number Minimization. In *Proceedings of the First Workshop on Algorithm Engineering and Experimentation (ALENEX'99)*. Springer Verlag, Lecture Notes in Computer Science, January 1999. Also available at http://www.cbl.ncsu.edu/publications/-#1999-ALENEX-Stallmann.

[21] M. R. Hartoog. Analysis of Placement Procedures for VLSI Standard Cell Layout. In *Proceedings of the 23st IEEE Design Automation Conference*, pages 314–319, July 1986.

[22] M. Hutton and J. Rose. Equivalence Classes of Clone Circuits for Physical-Design Benchmarking. In *IEEE 1999 International Symposium on Circuits and Systems – ISCAS'99*, May 1999. A reprint also accessible from http://www.cbl.ncsu.edu/experiments/DoEArchives/-I999-ISCAS-session.

[23] D. Ghosh and F. Brglez. Equivalence Classes of Circuit Mutants for Experimental Design. In *IEEE 1999 International Symposium on Circuits and Systems – ISCAS'99*, May 1999. A reprint also accessible from http://www.cbl.ncsu.edu/publications/#I999-ISCAS-Ghosh.

[24] W. Guenther and R. Drechsler. Creating Hard Problem Instances in Logic Synthesis Using Exact Minimization. In *IEEE 1999 International Symposium on Circuits and Systems – ISCAS'99*, May 1999. A reprint also accessible from http://www.cbl.ncsu.edu/experiments/DoEArchives/-I999-ISCAS-session.

[25] H. Lavana, F. Brglez, and R. Reese. User-Configurable Experimental Design Flows on the Web: The ISCAS'99 Experiments. In *IEEE 1999 International Symposium on Circuits and Systems – ISCAS'99*, May 1999. A reprint also accessible from http://www.cbl.ncsu.edu/-publications/#I999-ISCAS-Lavana.

[26] M. Stallmann, F. Brglez, and D. Ghosh. Evaluating Iterative Improvement Heuristics for Bigraph Crossing Minimization. In *IEEE 1999 International Symposium on Circuits and Systems – ISCAS'99*, May 1999. A reprint also accessible from http://www.cbl.ncsu.edu/-publications/#I999-ISCAS-Stallmann.

[27] M. Hutton and J. Rose. Applications of Clone Circuits to Issues in Physical Design. In *IEEE 1999 International Symposium on Circuits and Systems – ISCAS'99*, May 1999. A reprint also accessible from http://www.cbl.ncsu.edu/experiments/DoEArchives/I999-ISCAS-session.

[28] J. Harlow and F. Brglez. Mirror, Mirror On The Wall .... Is The New Release Any Different At All? In *IEEE 1999 International Symposium on Circuits and Systems – ISCAS'99*, May 1999. A reprint also accessible from http://www.cbl.ncsu.edu/publications/#I999-ISCAS-Harlow.

[29] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. SIS: A system for sequential circuit synthesis. Technical Report UCB/ERL M92/41, University of California, Berkeley, CA 94720, May 1992. Version 1.2 is available from UC Berkeley Design Technology Warehouse at http://www-cad.eecs.berkeley.edu/Software/software.html.

[30] A. Sangiovanni-Vincentelli R. Brayton, R. Rudell and A. Wang. MIS: A Multiple-Level Logic Optimization System. *IEEE Transactions on Computer-Aided Design*, pages 1062–1081, November 1987. Some of the results reported under *DoE_0001* are based on the Version MIS2.2, installed at eos.ncsu.edu.

[31] The VIS Group. VIS: A system for verification and synthesis. In R. Alur and T. Henzinger, editors, *Proceedings of the 8th International Conference on Computer Aided Verification*, number 1102 in Lecture Notes in Computer Science, pages 428–432, New Brunswick, NJ, July 1996. Springer. Version 1.3 is available from UC Berkeley Design Technology Warehouse at http://www-cad.eecs.berkeley.edu/Software/software.html.

[32] K. Kozminski, (Ed.). OASIS2.0 User's Guide. MCNC, Research Triangle Park, N.C. 27709, 1992. (Over 600 pages, distributed to over 60 teaching and research universities worldwide).

[33] V. Betz and J. Rose. VPR: A New Packing, Placement and Routing Tool for FPGA Research. In *Proceedings of the 7th International Workshop on Field-Programmable Logic*, pages 213–222, August 1997. Software and postscript of paper can be downloaded from http://www.eecg.toronto.edu/~vaughn/vpr/vpr.html .