B. Becker, R. E. Bryant, O. Coudert, Ch. Meinel (Hrsg.)

Computer Aided Design and Test

Dagstuhl–Seminar–Report; 105
13.2. – 17.2.1995 (9507)

Contents

Report on the Dagstuhl-Seminar

‘‘Computer Aided Design and Test’’

(13.2. – 17.2.1995)

Organizers: B. Becker, R. E. Bryant, O. Coudert, Ch. Meinel

Introduction

The third workshop on *“Computer Aided Design and Test”* at the IBFI Schloß Dagstuhl was organized by Bernd Becker (Univ. Frankfurt), Randy Bryant (Carnegie Mellon Univ.), Oliver Courdert (Synopsis) and Christoph Meinel (Univ. Trier). It was attended by 40 scientists. The organizers took the opportunity to bring together researchers from different areas in computer science, electrical engineering and industry.

The workshop focussed on *Binary Decision Diagrams (BDDs)* and related data structures in practical applications as well as in theoretical research. The success of BDDs in the CAD area has spawned research efforts on a number of fronts, including:

- Theoretical work on BDD-based data structures and algorithms;
- Applications in domains such as protocol verification, artificial intelligence, logic programming, and automated theorem proving;
- Extensions beyond Boolean functions to represent matrices, Markov systems, integer programming problems, multi-variate polynomials, and word-level circuit functions.

The workshop fostered deep and creative interactions among researchers in the CAD field, researchers from other domains making use of BDDs in their applications and theoreticians contributing to the core technology. Besides the common interest in BDD-based techniques which united all the participants the excellent atmosphere of the Dagstuhl-center provided the framework for an exceptional workshop.

More detailed information including some full papers can be found on the WWW-pages with the URL

http://www.informatik.uni-trier.de/Design_and_Test/

Probabilistic Analysis of Large Finite State Machines

Gary D. Hachtel, Enrico Macii, Abelardo Pardo, Fabio Somenzi
University of Colorado
Dept. of Electrical and Computer Engineering Boulder, CO 80309

Regarding finite state machines as Markov chains facilitates the application of probabilistic methods to very large logic synthesis and formal verification problems. In this paper we present symbolic algorithms to compute the steady-state probabilities for very large finite state machines. These algorithms, based on Algebraic Decision Diagrams (ADDs) – an extension of BDDs that allows arbitrary values to be associated with the terminal nodes of the diagrams – determine the steady-state probabilities by regarding finite state machines as homogeneous, discrete-parameter Markov chains with finite state spaces, and by solving the corresponding Chapman-Kolmogorov equations. We first consider finite state machines with state graphs composed of a single terminal strongly connected component; for this type of systems we have implemented two solution techniques: One is based on the Gauss-Jacobi iteration, the other is based on simple matrix multiplication. Then we extend our treatment to the most general case of systems which can be modeled as finite state machines with arbitrary transition structures; here out approach exploits structural information to decompose and simplify the state graph of the machine. We report experimental results for problems with over 10^{27} unknowns.

On Improving OBDD-Based Verification in a Synthesis Environment

Wolfgang Kunz
Max-Plank-Gesellschaft
Potsdam

We present a new methodology for formal logic verification for combinatorial circuits. Specifically, a structural approach is used based on indirect implications derived by recursive learning. This is extended to formulate a hybrid approach where the complexity of a subsequent functional method based on OBDDs. It is demonstrated how OBDD based verification can take advantage from structural preprocessing in a typical synthesis environment. The results show the effective compromise achieved between memory efficient structural methods and functional methods. One more advantage of these methods lies in the fact, that resources that go into logic synthesis can efficiently be reused for verification purposes.

On Computing the Maximum Power Cycle of Sequential Circuits

Abelardo Pardo
University of Colorado
Dept. of Electrical and Computer Engineering
Boulder, CO 80309

The maximum power dissipated by a combinatorial circuit can be obtained by computing the pair of input combinations that force the circuit to dissipate the most power. The counterpart of this problem in the case of a sequential circuit requires computing the cycle that represents the maximum sustained power that the circuit can dissipate. We present a technique that first computes the power dissipated in the circuit for every input combination and then solves the maximum average cycle in a dual graph obtained from the original one. Despite the fact that we have the problem involves extra complexity we have been able to obtain meaningful results for some medium size circuits.

Timed Boolean Functions and their BDD's

R. K. Brayton and W. Lam
University of California, Berkley CA

Timed Boolean functions (TBF's) represent an integrated view of a circuit's logic and timing behavior. Using TBS's, one can reason about and complete properties of the timing behavior of circuits, both combinatorial and sequential circuits. Using fixed delay model, for example, the TBF of a circuit is a function $f(x,t)$ where x are the Boolean input variables and t is time. Given the TBF's for a circuit's inputs, a circuit TBF can be seen as a partitioning of the real rime axis into a finite number of intervals. On each interval, the circuit is a fixed (time invariant) logic function. A TBFBDD is a canonical structure for r representing TBF's. A circuits TBF can be recursively computed starting from its inputs and using normal BDD operators (e.g. compose) combined with "time-decision' variables. These are Boolean variables of the form $t > 1.52$, for example. By using a simple reduction rule for time-decision variables (in addition to the usual BDD reduction rules) TBFBDD's can be made canonical. Further, it can be shown that a switching event can occur a time t_0 at a gate in a circuit if and only if its canonical TBFBDD has a time decision switching variable $t > t_0$ at a node in the TBFBDD. We illustrate several applications of TBFBDD's to computing the true delay of a circuit. In particular, for a sequential circuit, one can compute its exact minimum cycle time.

Ordered Kronecker Functional Decision Diagrams

Bernd Becker, Rolf Drechsler, Michael Theobald
J.W.G.-University Frankfurt Columbia University, NY

We have introduced the concept of Ordered Kronecker Functional Decision Diagrams (OKFDDs) as a means for the representation and manipulation of Boolean functions.

It is shown by an investigation of all possible decomposition types that only Shannon, positive Davio and negative Davio decompositions (the types considered in OKFDDs) are necessary to obtain the "full power" of Ordered Decision Diagrams. On the other bound exponential gaps between specific subclasses of OKFDDs (like OBDDs, OFDDs, pOFDDs, nOFDDs, ...) are presented. Combining these results it follows that a restriction of the OKFDD concept to subclasses leads to families of functions which lose their efficient representation.

Finally we demonstrate that these (more theoretical) results have a counterpart in practical applications. There is a package for comfortable and easy manipulation of OKFDDs available. First applications in logic synthesis and design for testability lead to significant progress. With dynamic variable reordering strategies we have obtain a reduction in size of more than 50% for the MCNC benchmarks compared to OBDDs. To obtain the package contact the second author under `drechsler@kea.informatik.uni-frankfurt.de`.

Lists of Ternary Vectors – Concepts – Properties – Comparison with BDDs

Bernd Steinbach
TU Bergakademie Freiberg

Boolean functions may be represent by lists of ternary vectors (TVL). These TVLs can be interpreted as disjunctive forms, conjunctive forms, antivalance forms or equivalence forms. If one ternary vector includes s stroke elements it describes 2^s binary vectors. This weaks the memory requirements enormous. We prefer orthogonal TVL in order to get very fast calculation time.

The implementation of the data structure TVL and more than 100 operations in the XBOOLE – Tool is characterized by unrestricted number of boolean variables, by computation of ternary operation in parallel, by restricted processing in sequence and by the possibility of calculation in several boolean spaces.

The comparison of TVLs and BDDs by means of the whole MCNC-benchmark set shows that TVL's needs in this case three times fewer memory space. If a boolean function has good decomposition properties and the number of boolean variables is greater than 40, the TVL needs in a single boolean space more memory

than the BDD, but using the multiple space representation the TVL needs less memory than the BDD. The comparison of memory requirements of a random boolean function (160 variables, 200 conjunctions) shows that TVL needs 8044 Bytes and BDD 547.294 Bytes.

On the Improvement of Variable Orderings for OBDDs

Beate Bollig, Martin Löbbing, Ingo Wegener
Universität Dortmund

Ordered binary decision diagrams are a useful representation of Boolean functions, if a good variable ordering is known. Variable orderings are computed by heuristic algorithms and then improved with local search and simulated annealing algorithms. This approach is tested on the conjecture that the following problem is NP-complete. Given an OBDD G representing f and a size bound s, does there exist an OBDD f^* (respecting an arbitrary variable ordering) representing f with at most s nodes? This conjecture is proved.

Afterwards, a simulated annealing approach with a new type of neighborhood is presented and analyzed. Better results as by known simulated annealing algorithms and heuristics are obtained. Some theoretical results underlining the experiments are stated.

An Efficient OBDD-Rebuilding: Algorithm, Implementation, and Applications

Jochen Bern, Christoph Meinel, Anna Slobodovà
Universität Trier

The size of the Ordered Binary Decision Diagram (OBDD) is very sensitive to the ordering of the variable. The minimization problem (and particularly the problem of finding an optimal variable ordering) of OBDDs is known to be NP-hard. There are several heuristics for finding a "good" variable ordering. Some of these are based on the improvement by local changes on the ordering.

We describe an implementation of the global rebuilding algorithm that transforms a given OBDD with respect to a given order. The algorithm runs in average time that is polynomial to the product of the sizes of the input and of the output and in linear space (up to the small factor that depends on the number of variables).

We show the usability of the algorithm on two basically different applications. In one of these the algorithm is used only once. The typical situation is the

requirement of testing equivalence or of performing the synthesis of two OBDDs that are inconsistent with respect to their variable ordering.

Multi-Level Logic Synthesis Based on Kronecker and Boolean Ternary Decision Diagrams for Incompletely Specified Functions

Marek Perkowski, Portland University
Andisheh Sarabi, Viewlogic Systems, Inc.
Ingo Shäfer, Arkus Design, Inc.

We introduce a framework as well as several new families of decision diagrams for representation for Boolean functions. The families include several diagrams known from literature (BDDs, FDDs) as subsets. Due to this property, these diagrams can provide a more compact representation of functions than either of the two decision diagrams. By classifying decision diagrams as free / repeated-variable, ordered, unordered, binary / ternary, homogeneous / non-homogeneous, bit-level / word-level and the type of reductions used, a framework for generalization of decision diagrams is presented. Kronecker functional Decision Diagrams (KFDDs) with negated edges are based on three orthogonal expansions (Shannon, positive Davio, negative Davio) and are created here for incompletely specified Boolean functions. Four other new families of functional decision diagrams are also presented: Pseudo Kronecker Decision Diagrams, Free Kronecker Decision Diagrams, Boolean Ternary Decision Diagrams and Boolean Kronecker Ternary Decision Diagrams. The last two families introduce nodes with three edges and require AND, OR and XOR gates for circuit realization. There are two variants of the last two families: canonical and noncanonical. While the canonical diagrams can be used as efficient general purpose Boolean function representation, the noncanonical variants are also applicable to incompletely specified functions and create don't cares in the process of their construction. These decision diagrams, as generalization of BDDs and FDDs lead to even more compact circuits in logic synthesis and technology mapping.

Using Functions Properties for Multi-Level and Multi-Domain Logic Synthesis

Udo Kebschull
Forschungszentrum Informatik
Karlsruhe

A new approach for the minimization of multi-level Boolean networks is presented. It combines conventional minimization techniques with methods in the

spectral domain. Therefore, it is a multi-domain approach, which may bee seen as an addition to other minimization techniques.

The basic idea is to insert ExOR gates into the multi-level design.

Common data structures for all Boolean normal terms based on decision diagrams are presented. Function properties like the symmetry, the linearity and the monotony for the operational and the spectral minimization. The principle of properly concentration of subcircuits is demonstrated. Further optimization steps like polarity assignment and don't care assignment for decision diagrams are presented.

EXOR Ternary Decision Diagrams on EXOR Logic Synthesis

Tsatoma Sasao
Kyasha Institure of Technology
Izuka, Japan

Compared with AND-OR based networks, AND-EXOR based networks require fewer gates, and are easily testable. This talk summarized EXOR logic synthesis using EXOR ternary decision diagrams. The outline were as follows:

1. Three types of expansions:
   Positive Davio expansion, Negative Davio expansion, Shannon expansion.

2. Decision Trees:

   - BDD (Binary Decision Diagram)
   - FDD (Functional Decision Diagram)
   - KDDD (Kronecker Decision Diagram)
   - PKDD (Pseudo Kronecker Decision Diagram)

3. AND-EXOR expressions:

   - PPRM (Positive Polarity Reed-Muller expression)
   - FPRM (Fixed Polarity Reed-Muller expression)
   - KRO (Kronecker expression)
   - PSDRM (Pseudo Reed-Muller expression)
   - PSDKRO (Pseudo Kronecker expression)
   - GRM (Generalized Reed-Muller expression)
   - ESOP (Exclusive-or sum of products expression)

4. EXOR Ternary Decision Diagram (ETDD): A data structure for optimization of AND-EXOR expressions and decision diagrams.

5. Optimization of AND-EXOR expressions:
   FPRM, KRO, PSDKRO, GRM and ESOP

6. Optimization of decision diagrams: FDD, KDD and PKDD

7. Generation of multi-level AND-EXOR networks.

Hybrid Fault Simulation for Synchronous Sequential Circuits Based on OBDDs

Rolf Krieger
J.W.G.-Universiät
Frankfurt / Main

Fault simulation for synchronous circuits is a very time-consuming task. The complexity of the task increases if the there is no information about the initial state of the circuit. In this case an unknown initial state is assumed which is usually handled by introducing a three-valued logic. As its is well-known fault simulation based on this three-valued logic is not accurate. It only determines a lower bound for the fault coverage which can be achieved by the test sequence. Therefore we present a hybrid fault simulator which improves the accuracy by performing a symbolic simulation based on OBDDs. It tries to combine the advantages of OBDDs with the advantages of the three-valued logic. Moreover it supports the single and the multiple observation time test strategy.

Our experimental results have shown that our approach improves the accuracy of fault simulation considerably.

A Symbolic Method to Reduce Power Consumption of Circuits Containing False Paths

R. I. Bahar, Gary Hachtel, E. Macii, F. Somenzi
University of Colorado
Dept. of Electrical and Computer Engineering
Boulder, CO 80309

Power consumption in technology mapped circuit can be reduced by performing gate re-sizing. Recently we have produced a symbolic procedure which exploits the compactness of the Algebraic Decision Diagrams (ADD) data structure to

accurately calculate the arrival times at each node of a circuit for an primary input vector. In the paper we extend our timing analysis tool to the symbolic calculation of required times and slacks, and we use this information to identify gates of the circuit that can be re-sized. The nice feature of our approach is that it takes into account the presence of false paths naturally. As shown by the experimental results, circuits re-synthesized with the technique we present in this paper are guaranteed to be at least as fast as the original implementations, but smaller and substantially less power-consuming.

Model Checking Commercial Designs

Carl Pixley, Bernard Plessier, Hyumoro Cho

Formal Verification of two commercial designs was discussed. The fact, a microprocessor cache flush unit had been extensively simulated by a product group and was thought to have no errors. Model Checking simple properties revealed one error. However after the error was corrected, the revised chip was checked with the same properties and was found to have yet another error. This is an example of Ex Post Facto verification.

Another design, a small / parallel bus interface for an automobile chip was designed with the help of a product group and verified by our team. Numerous properties were verified by Motorolas Verdict model checking tool (based upon SMV). The fist verification involved about 150 latches. Errors were discovered early in the design process. The fist silicon for the design passed all tests.

Efficient OBDD-based Boolean Manipulation

Christoph Meinel, Anna Slobodovà
Universität Trier

We present the concept of TBDDs which considerably enlarges the class of Boolean functions that can be efficiently manipulated in terms of small size OBDDs. This is done by applying the concept of domain transformations, which is well-known in many areas of mathematics, physics or technical sciences, to the context of BDD-based Boolean functions manipulation in CAD: Instead of verifying with the OBDD-representation of a function f, TBDDs allow to work with an OBDD-representation of a suited cube transformation function. Beside of giving some theoretical insights into the new concept, we investigate in some detail cube transformations which are base on complete types. We

- show that such TBDDs can be derived similarly as OBDD representations;

- give evidence of the practical importance of such TBDDs by presenting very small size TBDDs for the hidden weighted bit function (HWB) which were proved to have only exponential size OBDD - representations, and
- report some promising experimental results with the ISCAS benchmark multiplier circuit C6288.

BDD Minimization Based on Truth Table Permutations

Masahiro Fujita
Fujitsu Labs America
San Jose, CA95134
USA

Yuji Kukimoto
University of California
Berkley
USA

We show a way to reduce sizes of BDDs by permuting truth table rows. By appropriately permuting truth table rows, the resulting functions can have compact representation in BDD, still keep canonicity since all minterms in on -set can get together and constitute large cubes. However, permutations realizing such functions can be extremely complicated in the sense that BDDs representing those permutations blow up. So the problem is to find good permutations which reduce BDD sufficiently but their BDD representations are reasonable small.

Here we present a set of local transformations on BDDs which permute truth table rows on hopefully obtain significant reductions of BDD sizes. Since they are based on local restructuring of BDDs, they can be efficiently implemented and invoked whenever necessary, just like dynamic reordering. We present basic algorithms and initial experimental results.

Efficient ROBDD Computation of Common Decomposition Function of Multi-Output Boolean Functions

Christoph Scholl
Dept. of Computer Science
Universität des Saarlandes
D 66041 Saarbrücken, FRG

Paul Molitor
Dept. of Computer Science
Martin-Luther Universität, Halle
D 06099 Halle (Saale), FRG

One of the crucial problems multi level logic synthesis techniques for multi output boolean functions $f = (f_1, \ldots, f_m) : \{0,1\}^n \rightarrow \{0,1\}^m$ have to deal with is finding sublogic which can be shared by different outputs, i.e., finding boolean functions $\alpha = (\alpha_1, \ldots, \alpha_h) : \{0,1\}^p \rightarrow \{0,1\}$ which can be used as common

sublogic of good realizations of $f_1, \ldots, f_m$. We present an efficient ROBDD based implementation of this *common decomposition functions problem* (CDF). The key concept of our method is the exploitation of "equivalences" of the functions $f_1, \ldots, f_m$ which considerably reduces the running time of the tool. Formally, CDF is defined as follows: Given m boolean functions $f_1, \ldots, f_m : \{0,1\}^n \rightarrow \{0,1\}^m$, and two natural numbers p and h, find h boolean functions $\alpha_1, \ldots, \alpha_h : \{0,1\}^p \rightarrow \{0,1\}$ such that $\forall 1 \leq k \leq m$ there is a decomposition of the form

$$\begin{aligned} f_k(x_1, \ldots, x_n) = \; g^{(k)}(\; & \alpha_1(x_1, \ldots, x_p), \ldots, \alpha_h(x_1, \ldots, x_p), \\ & \alpha_{k+1}^{(k)}(x_1, \ldots, x_p), \ldots, \alpha_{r_k}^{(k)}(x_1, \ldots, x_p), \\ & x_{p+1}, \ldots, xn) \end{aligned}$$

using a minimal number r_k of single-output boolean decomposition functions.

Inductive Boolean Function Manipulation

Arti Gupta
NEC USA, C&C Research Lab
Princeton, USA

Allan Fisher
Carnegie Mellon University

A methodology for formal verification of inductively-defined hardware is described, based on automatic symbolic manipulation of inductive Boolean Functions (IBFs). It combines reasoning by induction and symbolic tautology-checking in a way that incorporates the advantages of both. The key idea is that by building an inductions argument into the IBF representations, explicit proofs by induction can be avoided. Our research has focused on two classes of IBFs – linearly in ductive functions (LIFs) and exponentially inductive functions (EIFs), intended to capture serial style of induction, and parallel divide-and-conquer style for induction, respectively. Their canonical representations and symbolic manipulation algorithms are based on extensions of Ordered Binary Decision Diagrams (OBDDs), where the complexity is independent of the induction parameters. The LIF schema is particularly useful since it naturally provides a canonical representation and symbolic manipulation for sequential functions in much the same way as OBDDs do for combinatorial functions. An interesting deterministic finite state automaton (DFA), i.e. a minimal DFA that accepts input strings in reverse order. Its practical usefulness is demonstrated by showing that in comparison to classic (forward) DFAs, several datapath circuits have exponentially more compact reverse DFAs. Symbolic IBF manipulation can be used to perform both functional verification of size-parametric hardware (regular in structure), and behavioral verification of sequential circuits (regular in time). Furthermore, the unified framework for handling space and time is useful for

handling a combination of both. Through techniques such as symbolic simulation of space-parametric circuits, and language containment for obtain network invariants.

BDD Trees

Ken McMillan
Cadence Berkley Labs

BDD Trees provide a canonical representation for boolean functions, based on hierarchical decomposition of the input space. Like OBDDs, BDD trees have a polynomial time "apply" algorithm, and an exponential, though heuristically efficient algorithm for many - variable boolean quantification. Thus BDD trees can be used for boolean functions comparison, or symbolic model checking in the same way as OBDDs. The advantage of BDD trees is that they are asymptotically efficient for circuits of bounded "tree width". This is a measure at the number of wires passing through any given node when the circuit is layed out as a tree. Experimentally, we find that BDD trees can be greatly more efficient than OBDDs in verifying tree structured systems, using symbolic model checking in the same cases we observe logarithmic growth of the BDD tree representation as a functions of circuit size, for suitable circuits.

Verification of Large Scale Microprocessors

Andreas Kuehlmann, David P. LaPotin, Florian Krohm, Arjon Mets, Mark Williams
IBM T.J. Watson Research Center, Yorktown Heights, NY

In an effort to optimize the performance of digital systems, designers of high performance circuits are moving from correct-by-construction synthesized methodologies to hand-crafted custom design. This fundamental shift has necessitated more complete methods for verifying correct system behavior. The verification program Verity addresses the problem of formally proving the correctness of a system implementation with respect to the specification.

Verity applies BDD-based comparison techniques which implicitly prove the functional equivalence between a CMOS circuit implementation and an RTL specification for all possible input patterns. The underlying extraction algorithms are based on a switch-level model of the MOS circuit, effectively removing the need for expensive simulation.

The algorithms applied in Verity permit the verification of entire microprocessors. THe success of formal methods on such a large scale requires a strict design-for-verification methodology. This significantly affects the overall design

partitions, the hierarchical circuit structure, the chosen register methodology and the assertions of logical boundary conditions.

Verity is in daily use within IBM and successfully been used for a Power PC implementation. This application demonstrates that traditional ROBDD-based techniques can handle the combinational verification problem of large chips without compromising the design process.

Edge-Valued Binary Decision Diagrams and their Applications

Yung-Te Lai Sarma B. K. Vrudhula
Hitachi, Inc ECE Dept, Univ. of Arizona

We describe a data structure called Edge-Valued Binary Decision Diagrams (EVBDD). An EVBDD provides a canonical an compact representation of functions that involve both integer and boolean quantities. This makes them suitable for a variety of important applications such as Integer Linear Programming, Spectral Transformation of Boolean Functions, Multiple Output Decomposition of Boolean functions and hierarchical verification. In this talk, we describe the structure of EVBDDs, present a general algorithm for applying an arbitrary binary operator that can be exploited to significantly improve the Computation efficiency.

We present an applications of EVBDDs to logic Verification where the objective is to show the equivalence between a specification and an implementation. In this regard EVBDDs provide two significant advantages over OBDDs (Ordered Binary Decision Diagrams). First they allow equivalence checking between Boolean functions and arithmetic functions. Second, they allow hierarchical designs, i.e. the implementation of the design can be described by previously Verified components, rather than having to flatten the design to the gate level.

Two Questions

Olivier Coudert
Synopsis, 700 East Middlefield Rd..
Mountain View, CA 94040

Let f be a Boolean function from $\{0,1\}^n$ into $\{0,1\}$. We make no distinction between a Boolean formula and the subset of $\{0,1\}^n$ it is the characteristic function of. A product (or cube) is the conjunction of literals, e.g.,

$$\overline{x}y\overline{z}$$

is a product. A product p is an implicant of f iff f contains p. An implicant p of f is prime iff there is no other implicant of f that contains p. We note Prime(f) the set of all prime implicants of f.

Brayton, McGeer and Sanghavi introduced a function, called signature cube, that maps minterms of f on products. The signature cube $\sigma(x)$ of a minterm x of f is defined by:

$$\sigma(x) = \cap_{p \in \text{Prime}(f), p \ni x} p$$

They described a technique to produce a covering matrix expressing the two-level logic minimization of f. The interest of this technique is that the size of the covering matrix can be far less than the number of prime implicants of f. This enabled to minimize function with huge number of prime implicants.

To compare this technique with a BDD/ZBDD based one, one can state the following questions:

1. Does it exist a function f such that

   $$|\max_{\subseteq} \sigma(x) | x \in f| | \max_{\subseteq} \sigma(x) | x \in f|$$

   is polynomial, and such that the BDD of f has a non polynomial size whatever the variable ordering is?

2. How are related the size of the BDD of f and the size of the ZBDD of Prime(f)?

Programmschemata and OBDD's

Hans Eveking
Computer Science Department/Design Methodologies
University of Frankfurt
Robert-Mayer-Str. 11-15
D-60054 Frankfurt, Germany

Transition systems are represented by means of a notation that includes constructs of ordinary programming languages like conditional assignments and while-loops. A theory of programmschemata which goes back to Glushkov (1965) provides axioms and rules-of-inferences for equivalence preserving transformations of such constructs. Theorems for the transformation of while-loops are derived on this basis.

Equivalence preserving transformations are combined with pre/post-condition equivalence proofs of small sequences of operations, e.g., that two shift operations

of one step are replaced by a single shift of two steps. This type of proof is performed by means of OBDD's starting from a VHDL- specification employing generic bit-vector functions like "add-with-carry".

The procedure is applied to derive complex algorithms from a primitive original one, and - in this sense - to demonstrate the correctness of the complex algorithms. Examples include the derivation of multibit scanning multiplication-algorithms from a primitive algorithm, the transformation of restoring into non-restoring division, and pipelining problems.

Algebraic Methods of Finite-State Machine Verification

Stefan Krischer
Universität Trier

The talk shows how state homomorphisms can be constructed in order to prove the equivalence of two finite-state machines (FSMs) whose state spaces depend on each other. The FSMs represent synchronous sequential circuits. Typically one machine is derived from the other one by using an optimization tool. If this tool only performs homomorphic transformations, then our method can be applied. The best would be that the optimization yields information about the transformation it did to a verification component, which computes a mapping and verifies that this mapping is in fact a homomorphism.

Comparing kOBDDs and kIBDDs

Beate Bollig, Martin Sauerhoff, Detlef Sieling, Ingo Wegener
Universität Dortmund

(This talk should have been given by D. Sieling, who could not come, therefore a short survey was given by I. Wegener)

The representation of kOBDDs (k layers of OBDDs which repeat to the same variable ordering) and kIBDDs (k layers of OBDDs which perhaps different variable orderings) are compared. SAT is NP-hard already for 2IBDDs but in P for kOBDDs and constant k. The test whether a matrix X is a permutation matrix is representable by linear-size 2IBDDs but kOBDDs need exponential size. Finally, using complex arguments from communication complexity, the following hierarchy result is proved – An appropriate pointer jumping function can be represented by kOBDDs of polynomial size but only with (k-1)IBDDs of exponential size ($k \leq (1-a) \log \log n$).

The Theory of Zero-Suppressed BDDs and the Number of Knights Tours

Olaf Schrör, Ingo Wegener, part also Martin Löbbing
Universität Dortmund

Minato has presented ZBDDs and given some applications. Here, the complete theory of ZBDDs is wrenched out (decomposition type, structure theory, reduction rules, a generic synthesis algorithm, replacement by constants and functions, equality test, SAT and SAT-COUNT). Surprisingly, synthesis with O-preserving operators is simpler than with others. The maximal trade-off between ZBDDs and OBDDs is worked out. Then ZBDDs and MODs are applied to the estimation of the number of Knight's tours. The exact number of cycle coloring on 8×8 chess board is determined and the quotient of the upper and lower bound on the number of Knights tours is decreased from $2 \cdot 10^{14}$ to $3 \cdot 10^6$.

Symmetry Detection and Dynamic Variable Ordering

Shipra Panda, Fabio Somenzi
Dept. of Electrical and Computer Engeneering
University of Colorado at Boulder

Bernard Plessier
Motorola Inc.
Austin, TX

Knowing that some variables are symmetric in a function has numerous applications; in particular it can help produce better variable orders for Binary Decision Diagrams (BDDs) and related data structures (e.g., Algebraic Decision Diagrams). It has been conjectured that there always exists an optimum order for BDD wherein symmetric variables are contiguous. We propose a new algorithm for the detection of symmetries, based on dynamic reordering, and we study its interaction with the reordering algorithm itself. We show that combining sifting with an efficient symmetry check for contiguous variables result in the fastest symmetry detection algorithm reported to date and produces better variable orders for many BDDs. The overhead on the sifting algorithm is negligible.

Symmetry Based Variable Reordering for ROBDDs

Dirk Möller, Paul Molitor
Institurt für Informatik
Martin-Luther-Universität
Halle

Rolf Drechsler
Fachbereich Informatik
J. W. Goethe-Universität
Frankfurt

The ROBDDs are a data structure frequently used for representation and manipulation of Boolean functions. Since the size of the ROBDD for a function is extremely sensitive to the variable order a lot of heuristics to get a good variable order have been developed. For the class of partially symmetric Boolean functions we present a new general method to improve quality of reordering heuristics based on the exchange of variables. Statistical and benchmark results are given to show the efficiency of out approach.

On Generation for Free BDDs

Jawahar Jain
Fujitsu Labs of America
77 Rio Robles
San Jose, CA 95134
USA

Orthogonal correspondence theorem was shown as a basis for generating typed BDDs as well as generating a new data structure we will call as Multiple Rooted Free BDD (MRFBDD). MRFBDDs can be shown to be exponentially more compact than a singly rooted free BDD. Also IBDD simplification heuristics can be used to generate free BDDs. Such heuristics were currently being used to generate an equivalence proof between 2 different IBDDs. Such simplification heuristics can generate small free BDDs with "arbitrary" structure as contrasted with typed free BDD. Parallels between snDDs, MTBDDs, EVBDDs & ADDs were discussed & it was shown how snDD techniques can be used to augment such representations. Augmenting snDDs using EVBDD techniques was also discussed.

Arithmetic Circuit Verification with Binary Moment Diagrams

Randal E. Bryant
Carnegie Mellon University
Pittsburgh, PA
USA

Binary Moment Diagrams (BMDs) provide a canonical representation of functions mapping Boolean variables to numeric (typically integer) values. They are especially well-suited for representing the word-level functions of arithmetic circuits. I describe two approaches to verifying arithmetic circuits – a hierarchical method developed by my research group, and a non-hierarchical method due to Hamaguchi. We have verified integer multipliers with word sizes up to 256 bits. I present ideas on how to verify dividers.

Participants

Iris **Bahar**
University of Colorado
Dept. of Electrical and Computer Engineering
Campus Box 425
Boulder CO 80309
USA
iris@billie.colorado.edu
Tel: +1-303-492-1135

Bernd **Becker**
Universität Frankfurt
FB 20 Informatik
Robert-Mayer-Str. 11-15
D-60325 Frankfurt
Germany
becker@kea.informatik.uni-frankfurt.de
Tel: +49-69-798-8250

Jochen **Bern**
Universität Trier
Fachbereich IV - Informatik
D-54296 Trier
Germany
bern@uni-trier.de
Tel: +49-651-201-2840

Beate **Bollig**
Universität Dortmund
Lehrstuhl Informatik II
D-44221 Dortmund
bollig@pandora.informatik.uni-dortmund.de
Tel: +49-231-755-2598

Thierry **Bravier**
Dassault Aviation
DGT/DEA/IA2
78 Quai Marcel Dassault
F-92552 St. Cloud Cedex 300
bravier@dassault-avion.fr
Tel: +33-1-47 11 53 07

Robert K. **Brayton**
University of California at Berkley
Dept. of Electrical Engeneering and Computer Science
519 Cory Hall
Berkley CA 94720
USA
brayton@eecs.berkley.edu
Tel: +1-510-643-9801

Randy **Bryant**
Carnegie Mellon University
School of Computer Science
Pittsburgh PA 15213
USA
randy.bryant@cs.cmu.edu
Tel: +1-412-268-8821

Olivier **Coudert**
Synopsis Inc.
700 East Middlefield Road
Mountain View 94043-4033
USA
coudert@synopsis.com
Tel: +1-415-694-1985

Rolf **Drechsler**
Universität Frankfurt
FB 20 Informatik
Robert-Mayer-Str. 11-15
D-60325 Frankfurt
Germany
drechsler@kea.informatik.uni-frankfurt.de
Tel: +49-69-798-8472

Hans **Eveking**
Universität Frankfurt
FB 20 Informatik
Robert-Mayer-Str. 11-15
D-60325 Frankfurt
Germany
eveking@kea.informatik.uni-frankfurt.de
Tel: +49-69-798-8149

Thomas **Filkorn**
Siemens AG München
Abt. ZFE T SE1
Otto-Hahn-Ring 6
D-60325 München
Germany
filkorn@zfe.siemens.de
Tel: +49-89-636-45760

Masahiro **Fujita**
Fujitsu Labroratories of America Inc.
77 Rio Robles
San Jose CA 95134
USA
fujita@fla.fujitsu.com
Tel: +1-408-456-1151

Aarti **Gupta**
NEW C&C Research Laboratories
4 Independent Way
Princeton NJ 08540
USA
aguptaa@cs.cmu.edu / agupta@ccrl.nj.nec.com
Tel: +1-609-951-2966

Gary D. **Hachtel**
University of Colorado
Dept. of Electrical and Computer Engineering
Boulder CO 80309-0430
USA
hachtel@dufay.colorado.edu
Tel: +1-303-492-8728

Günter **Hotz**
Universität des Saarlandes
Fachbereich 14 – Informatik
Postfach 15 11 50
D-66041 Saarbrücken
Germany
hotz@cs.uni-sb.de
Tel: +49-681-302-2414

Jawahar **Jain**
Fujitsu Labroratories of America Inc.
77 Rio Robles
San Jose CA 95134
USA
jawahar@logic.fla.fujitsu.com
Tel: +1-408-456-1194

Andreas **Kühlmann**
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights NY 10598
USA
kuehl@watson.ibm.com
Tel: +1-914-945-3458

Udo **Kebschull**
FZI Karlsruhe
Haid-und-Neu-Str. 10-14
D-75131 Karlsruhe
Germany
kebschull@fzi.de
Tel: +49-721-9654-402

Rolf **Krieger**
Universität Frankfurt
FB 20 Informatik
Robert-Mayer-Str. 11-15
D-60325 Frankfurt
Germany
becker@kea.informatik.uni-frankfurt.de
Tel: +49-69-798-8155

Stefan **Krischer**
Universität Trier
Fachbereich IV - Informatik
D-54296 Trier
Germany
krischer@uni-trier.de
Tel: +49-651-201-2832

Wolfgang **Kunz**
Universität Potsdam
MPAG Fehlertolerantes Rechnen
Am Neuen Palais 10
D-14415 Potsdam
Germany
wkunz@mpag-inf.uni-potsdam.de
Tel: +49-331-977-1191

David **LaPotin**
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights NY 10598
USA
dpl@watson.ibm.com
Tel: +1-914-945-2586

Dirk **Möller**
Universität Halle
Inst. für Informatik
Weinbergweg 17
D-06099 Halle
Germany
moeller@infsparc2.informatik.uni-halle.de
Tel: +49-345-622-522

Kenneth **McMillan**
Cadence Berkley Labs
Suite 303
1919 Addison St.
Berkley CA 94704
USA
mcmillan@cadence.com
Tel: +1-408-894-2488

Christoph **Meinel**
Universität Trier
Fachbereich IV - Informatik
D-54296 Trier
Germany
meinel@uni-trier.de
Tel: +49-651-201-2827

Shipra **Panda**
University of Colorado
Dept. of Electrical and Computer Engineering
Campus Box 425
Boulder CO 80309=0425
USA
pandra@rtt.colorado.edu
Tel: +1-303-492-1135

Abelardo **Pardo**
University of Colorado
Dept. of Electrical and Computer Engineering
Campus Box 425
Boulder CO 80309-0425
USA
pandra@duke.colorado.edu
Tel: +1-303-492-1135

Carl **Pixley**
Motorola Inc.
Suite 200
Bridgepoint Plaza I
5918 West Countyard Dr. Suite 200
Austin TX 78730
USA
carl_pixley@email.mot.com
Tel: +1-512-794-4049

Jan **Roßmann**
Universität Trier
Fachbereich IV - Informatik
D-54296 Trier
Germany
rossmann@uni-trier.de
Tel: +49-651-201-2830

Andisheh **Sarabi**
Viewlogic Systems
47211 Lakeview Blvd.
Fremont CA 94538
USA
andisheh@viewlogic.com
Tel: +1-510-659-4001

Tsutomu **Sasao**
Kyushu Institute of Technology
Kawazu 680-4
Ilzuka 820
Japan
sasao@cse.kyutech.ac.jp
Tel: +81-948-29-7675

Christoph **Scholl**
Universität des Saarlandes
Fachbereich 14 – Informatik
Zimmer 309
Postfach 15 11 50
D-66041 Saarbrücken
Germany
scholl@cs.uni-sb.de
Tel: +49-681-302-2274

Anna **Slobodovà**
Universität Trier
Fachbereich IV - Informatik
D-54296 Trier
Germany
anna@uni-trier.de
Tel: +49-651-201-2833

Fabio **Somenzi**
University of Colorado
Dept. of Electrical and Computer Engineering
Boulder CO 80309-0425
USA
fabio@colorado.edu
Tel: +1-303-492-3466

Bernd **Steinbach**
TI - Bergakademie Freiberg
Fakultät für Mathematik und Informatik
B.-v.-Cotta-Str. 1
D-09599 Freiberg
Germany
steinb@informatik.tu-freiberg.de
Tel: +49-3731-39-2568

Michael **Theobald**
Columbia University
Computer Science Department
500 W 120th Street
450 Computer Science Building
New York NY 10027
USA
theobald@cd.columbia.edu
Tel: +1-212-939-7065

Sarma **Vrudhula**
Department of ECE
University of Arizona
Tucson AZ 85721
USA
sarma.arizona.edu

Ingo **Wegener**
Universität Dortmund
Lehrstuhl Informatik II
D-44221 Dortmund
wegener@ls2.informatik.uni-dortmund.de
Tel: +49-231-755-2776

Bernd **Wurth**
TU München
Lehrstuhl für Rechnergestütztes Entwerfen
D-80290 München
Germany
bew@e-technik.tu-muenchen.de
Tel: +49-89-2105-3659