Ch. Meinel, H. Sack, A. Wagner:
WWW.BDD-Portal.ORG: An Experimentation Platform for Binary Decision Diagram Algorithms
In: Experimental Algorithmics - From Algorithm Design to Robust and Efficient Software, R. Fleischer, E. Meineche Schmidt, B. Moret, LNCS 2547, Springer Verlag, Berlin, Heidelberg, New York 2002, pp. 127-137, ISBN: 978-3-54000-346-5.

WWW.BDD-Portal.ORG: An Experimentation Platform for Binary Decision Diagram Algorithms

Christoph Meinel, Universität Trier
Harald Sack, Universität Trier
Arno Wagner, ETH Zürich

Abstract

There is an upcoming need for World Wide Web portal sites to facilitate access to resources for specific research communities. The portal site described in this paper provides a testbed functionality besides additional information resources that are of interest in the research in Binary Decision Diagrams (BDDs). In the last decade, BDDs have proven to be the state-of-the-art data structure in computer aided design of integrated digital circuits. To assess the strengths and weaknesses of manipulation algorithms for BDDs, benchmark calculation is one of the most important methods in BDD-research. Due to the inherent high sensitivity of these algorithms to the particular experimental setup it is rather difficult or even impossible to reproduce benchmark results for comparison or for independent result verification. We have designed and implemented a WWW based BDD-testbed that overcomes these problems and greatly facilitates BDD algorithm comparison.

1 Introduction

Today the Internet offers the possibility of standardized and global communication without a need for special hardware or expensive infrastructure. While the World Wide Web (WWW) has become the largest information resource, esp. also for sciences and research, its inherent lack of any structure is responsible for the fact that the direct access to specific information is difficult. Search engines are still far from indexing even large fractions of the entire web [LG99] and even concerning the part being indexed the results of their searches are almost unstructured. Thus, in most cases, the user is faced to an all-or-nothing situation, where queries return even far too many hits or none at all.

1.1 WWW Portal Sites for Research Communities

One solution is the recent notion of specialized "portal"-sites that organize WWW-content into categories and in some cases grade the quality of the provided information. But, most portal-sites are targeted to the general public and relatively small communities as, e.g. research communities are not commercially interesting to the existing portals. On the other hand, research oriented portal sites might also provide access to highly specialized research tools available to the research community via an appropriate web interface.

Today, in computer science, experimental evaluation of algorithms resulting from theoretical work has become more important. This comes due to the fact that more and more important algorithms, although having bad theoretical average-case properties due to their heuristic nature, perform well in practical applications.

1.2 Binary Decision Diagrams

Algorithms for Binary Decision Diagrams (in particular we address Ordered Binary Decision Diagrams, but they will be further simply referred to as BDDs or OBDDs) are a good example for that case. In computer aided design of very large scale integrated circuits BDDs have been established as the state-of-the-art data structure. They are extensively used for simulation, modeling, and verification of digital circuits, often being orders of magnitude more powerful than other techniques. For an overview of BDD related research, see [MT98].

But, while BDD-based methods perform well in many cases, the underlying problem of representing subsets of a Boolean vector space is known to be hard. This means that circuit descriptions given as a relatively small Boolean formula can have an extremely large BDD representation. Unfortunately, formulas and other alternative representation techniques are unsuitable for use in computations and at present it seems that BDDs are the most convenient representation for these purposes.

The important optimization algorithms for BDDs have exponential (time and memory) worst case complexity, but perform still better than any algorithm for alternative data structures in that particular area. To achieve meaningful results at reasonable expenses, heuristics are applied and for most applications they can be utilized to solve practical problems. But, the problem remains to get a meaningful assessment of the power of these heuristical methods. And for that reason, benchmark computations for comparing the qualities of different optimization heuristics are applied.

Algorithms for BDDs are rather sensitive to the details of the environment that is used for experimentation. Usual research papers will state something like "benchmark computation was done on a UltraSparc with 512MB of memory". While this will be adequate for some classes of algorithms and will allow reasonable predictions for the performance on other machines, it is insufficient for BDD benchmarking. Small details like a slightly different compiler or a different OS version can have a large impact on BDD performance. For that reason it is often impossible to verify published research results, sometimes to the extend that not even the authors of a paper are able to reconstruct their results a year later without extensive reverse engeneering. To make the situation even worse, BDD algorithm performance is very much dependent on the exact nature of the computed benchmark. Performance figures for some benchmarks do not allow a reasonable prediction of the performance of a new heuristic for other types of benchmarks. This fact drastically limits the usability of published benchmark results and forces research groups to reimplement the heuristics they are interested in, in order to obtain their own benchmark results for their setting.

To address this problem, a finely standardized experimentation platform is required. Because of the necessary degree of accuracy, the only feasible way to achieve this, seems to be to provide a set of identical benchmark servers, where researchers can perform benchmark computations with their very own benchmarks. In this way they are able to evaluate new heuristics on their own circuits and verify other published results.

The authors have created a powerful and versatile experimentation environment that is fit for real world use and has actually been available since 1999. This environment offers access to a number of research tools that contain recent BDD algorithms. Questions of scheduling, load balancing, and error recovery have been solved in a satisfying manner within this system. It is embedded into a larger environment, a WWW portal site that supports the BDD research community in other ways as well.

When researchers do real world benchmark computations on systems other than their own, security and confidentiality questions arise. The authors have addressed these questions by allowing the use of encoded BDDs as input data format for benchmark computations. Because OBDDs have a canonical structure, circuit details are hidden. The purely Boolean function represented by an BDD is usually not a secret. Furthermore, this abstraction step does not reduce the meaningfulness of the computation as BDD-algorithms usually do this as a first step anyway.

An important step to gain acceptance with such an approach is the inclusion of methods developed by researchers other than the authors. The current system already features several heuristic methods provided by other researchers [HS99, DBG95, BLW95, MST98], with the prospect that this number will grow.

2 A Benchmarking Platform for BDDs

When new BDD heuristics are developed, they are usually added to some existing software package. After optimization, when a sufficient level of performance is reached, the algorithm is presented to the community. Usually, evaluation is carried out by using benchmarks from a standard benchmark set first, for example the ISCAS 89 ([BBK89]) set of circuits. Due to the nature of the problem, performance both in speed and memory consumption can vary extremely between different circuits of the benchmark. Predictions about the behavior of a heuristic on other circuits are extremely difficult or even impossible. Because of this, there often arises the whish to allow other researchers to use the software, to evaluate the suitability of the new heuristic applied to their own problems.

2.1 To Publish Code is not Optimal

But, here the problem manifests itself. Let us consider giving away the source code of the implemented algorithm:

- The code usually only works within the software package it was written for.

- The code has research level quality and may still lack documentation or even worse, contain errors.
- Due to some ideas that are not yet published, one may not want to give away all the details of the implementation.

So, this is not an appealing option. But giving away executables means no improvement either:

- The software has to be adapted to different platforms.
- There is need for support for every other possible platform.
- There is need to develop the software to a higher level of stability.

Another option is reimplementation of the algorithm by interested parties. But this requires even larger effort and special experience. Almost nobody is willing to undertake such an effort for an uncertain outcome.

Further complications arise because the published results are difficult to reproduce as fine-tuning often is required and usually, it seems to be critical for good results. This makes comparisons outside of the published results extremely difficult. We think most of these issues can be addressed in a satisfying way by providing the possibility to access the tools containing heuristics by using the Internet.

We believe that some requirements need to be addressed to make an Internet based solution truly usable. Ease of use is achieved by using a WWW interface, but other aspects are important as well. The WWW Interface alone does not automatically improve the quality of the program code, but a well suited wrapper for external code might prevent a lot of possible ommisions and errors, which the authors did not pay attention in their coding. On the other hand, the problem of implementing the software within your own computing environment including all earlier mentioned error probabilities can be prevented. As the computations done are time and memory intensive (i.e. taking hours of cpu-time and up to hundreds of megabytes of memory), just writing some cgi-scripts clearly would represent an inadequate solution. A significantly larger effort is required, even if the system is only used for the evaluation of heuristics. Usability would be critically lowered if getting computations done would take very long or only very small "toy" examples could be computed.

2.2 What is Really Needed

Flexibility

It should be possible to add tools of various nature to the system with reasonable effort. We do not want to limit the system to a specific tool. If some researcher has a new heuristic and wants to publish it via our system, the amount of customization necessary should be as low as possible, and ideally, no more than a recompilation for the computing platform in use should be necessary.

Speed

To achieve an appropriate overall speed it is necessary to distribute the actual computations over several computers. The number of engaged computers should be easily adjustable, and it should even be possible to include computers that are remote and only reachable via an Internet connection. To maintain comparability of the results, it is mandatory that the pool of computers can be divided into groups of machines with comparable computing power.

Reliability

As computations can take hours and as there might be a number of still pending computations, there should be a mechanism that allows automated crash recovery without loss of submitted requests.

3 A Web-Based Testbed

With OHO (for OBDD Heuristics Online) we have developed a testbed environment that meets the requirements mentioned above [MW99].

3.1 The WWW Interface

The system is accessed by a web interface. After an introductory page containing general information, the user can access a menu that allows the selection of submission forms for individual tool and heuristic types, as well as browsing the specific documentation. When one of the possible types of computation is selected, the user is guided to a submission form, where all the relevant information needed for a first meaningful evaluation of an algorithm can be enterd. This includes an (optional) e-mail address for notification about results, tool options and selection of input data. Options include the choice between the featured heuristic for reordering or some reference heuristic. Input data can be provided by either, some predefined circuit, or by a circuit description file transferred by the user. File upload is done with the browser, a feature available with most ordinary Web browsers.

After the submission the request is queued and computations begin as soon as the required resources will become available. If a notification email address has been provided, the results will be delivered immediately after completion.

3.2 Implementation

The basic structure of the system is simple. Individual tools have individual user interfaces that have the task of accepting new requests, giving feedback about the status of a computation submitted previously, and returning the results to the submitter. A central scheduler manages every submitted request during its whole life-cycle. A group of computers perform the actual computations necessary to complete a request. If a computation is terminated because of unavailable resources due to local usage of the computer it was scheduled on, the request will be rescheduled on an other computer or restarted later. We use Linux as operating system.

To achieve **flexibility** every contained tool is fitted with a small wrapper script that controls its I/O. As long as the tool communicates via command line, processing the standard input and output with files, the customization of these scripts is quite simple. We believe that this approach covers the majority of research tools.

Speed is achieved by distributing the computations, as mentioned earlier. This distribution is managed by the central scheduler in such a way that non-dedicated computers can be used. Queuing of requests is performed here as well.

As requests will be queued and queues might get longer, **reliability** becomes an important issue. In order to achieve this, the scheduler performs frequent dumps of the current system status. In case of a crash the scheduler will automatically recover with the help of these dump files and will resume computations without the need for manual intervention.

3.3 Available BDD Tools

At the moment we have integrated recently published heuristics added to nanotrav (part of the CUDD system, [Som96]) by a number of researchers. These heuristics are not part of the CUDD standard distribution itself, but rather actual research code provided by the developers of the specific heuristics. There are as well some heuristics that are part of the CUDD standard distribution for the purpose of comparison. We also have heuristics online that were added recently to the well known model checker SMV ([oCS]). At the moment, only new heuristics for model checking are provided that have been implemented by our research group. Additionally we had made available another online BDD heuristic from external researchers for the verification and synthesis tool VIS ([BHSVS96]), before it was included into the last recent release of VIS.

Of course, we intend to incorporate many more tools and heuristics in the near future and therefore, we invite every researcher, who plans the publication of a specific new heuristic or any new tool within our portal to contact us. As stated above, we do not need access to source code and the addition of usual research tools will not be a problem at all.

4 Added Value: A BDD Portal Site

The WWW has the potential for becoming something like a "super library", allowing access to all information that somebody finds relevant enough to put up on some web pages. Today, its size and growth are impressive and the basic technology for accessing the WWW and maintaining a presence on the WWW seem to be adequately developed. But, by its very nature as a decentralized, somehow "anarchistic" medium, *structure* is the important point that is missing

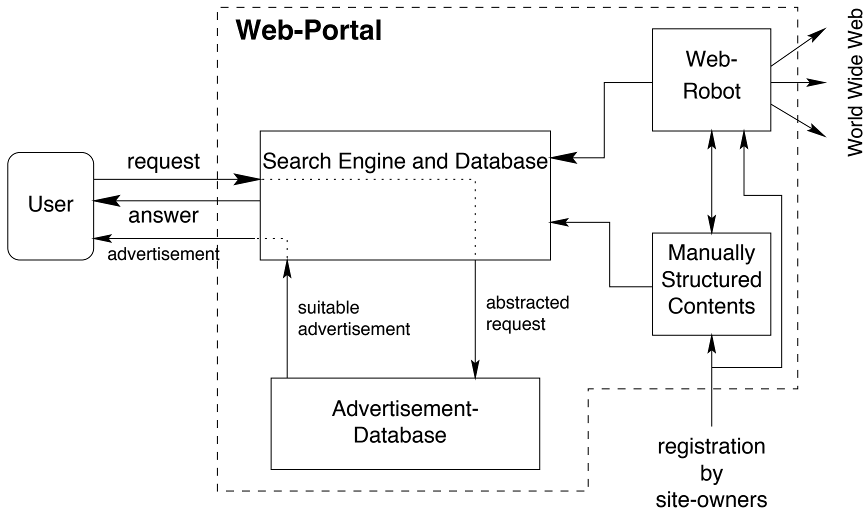

Figure 1: A Conventional Portal

within the WWW. Despite the existence of sophisticated search engines it is often still difficult to find specific content. If someone only relies on traditional search engines the situation is likely to get even worse [LG99], and especially so for research content of limited public interest. We will discuss some of the reasons now.

4.1 Structure of a Conventional Portal

By a *conventional portal* we mean a general search site, as, e.g. *Yahoo!* [YAH] that supports general (raw) searches and in most cases has a set of prearranged categories where handpicked sites are accessible in a more structured way. Figure 1 depicts the basic structure of such a conventional portal site. The main information gathering mechanism is shown to the right: A web-robot or "spider" that searches known pages within the WWW for embedded links to other pages and includes the pages linked into the database of known pages. Usually, there is also some manual insertion of appropriate starting points for these searches, when new popular sites become online. As a third mechanism to insert new pages into the database, most conventional portals also offer the possibility that pages might be registered by someone, which usually means the site owners.

The central component of a conventional portal is the database including more or less sophisticated its search mechanisms. This component processes the search requests submitted by the users. In most cases, there is also a database of commercial advertisements that are sent back with the pages found for the users requests. Often additional effort is spent to match the advertising to the request the user has made.

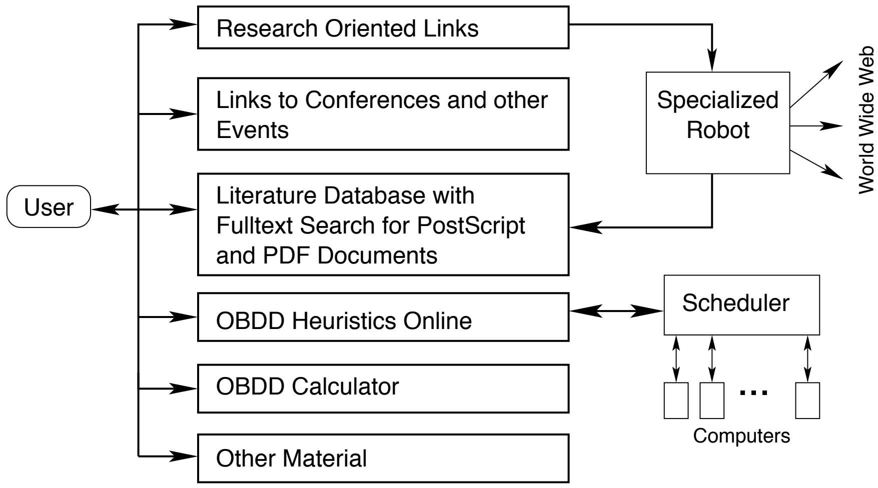

Figure 2: The BDD Portal

4.2 Shortcomings of Conventional Portals

Here exactly, the inadequacy of a conventional portal manifests itself: The primary motivation for operating a conventional portal is to earn money. This is achieved by being paid for advertisements sent to the user together with the answers retrieved for his query. Hence, the main focus is to attract as many people as possible that are interested at least in some of the advertising delivered by the portal. And here the time for manual improvement of the content will be spent. Web pages of interest only to a small group, as e.g. researchers in a specific field, will usually only be added to the database if they are registered manually by their owners. With the number of conventional portals out there, registering pages most times requires a significant amount of work and additional maintenance.

For these reasons we believe that there is need for small, specialized portal sites that support specific research communities. These specialized portals should contain a significant amount of structured and preselected contents aimed at the specific needs of the addressed research community.

4.3 The BDD Portal

The structure of our BDD Portal can be found in Figure 2. The topmost component consists out of a collection of research oriented links. This includes a list of links to homepages of active researchers. "Active" in that particular sense means that they have to have at least one publication in the area of Binary Decision Diagrams. This list aims to be a complete representation of the BDD research community. For researchers, where no homepage could be found, the email address is provided instead. Additionally, links to specific BDD oriented research projects and working groups are gathered, and not forget to mention links to companies that are commercially promoting BDD technology and research.

The second component is serving as a datebook for all important events concerning BDDs, like conferences and workshops, including all relevant information as dates, deadlines and links to the according homepages. This collection is subject to permanent update.

The third component comprises a database of literature on BDDs available on the WWW. This database contains information about technical reports, papers and other texts, which can be searched in their full text, even if the documents are only available in Postscript or PDF (portable document format). Search results will include the first lines of text for documents found and links to their original location. The documents for this database are periodically collected by a specialized robot, that is driven by the links to homepages of researchers. The robot is capable of recognizing document formats and creates a searchable index out of them. This component represents a significant improvement compared to conventional search engines, which, in most cases are not able to access the contents of non-html documents. This database is endorsed by a collection of links to journals, where articles on BDDs are published, specific series of technical reports related to the subject, and, as well, by links to available monographs and textbooks. Additionally, links to lecture notes and course materials of several international universities are provided and maintained.

The fourth component, OBDD Heuristics Online, is our effort to address some of the specific problems with the publication of results from research in heuristics for BDDs, which was explicitely described in the previous section.

In addition to these four components, a tool providing the possibility to test and to visualize BDD computations is added, the OBDD Calculator. The OBDD Calculator offers the possibility to perform BDD manipulation operations and symbolic simulation of formula input or circuit descriptions. It serves as a graphical front-end to the CUDD BDD-package and provides the possibility of a graphical visualization of the computed BDDs. In addition to the BDD Calculator, a recently developed visualisation tool for BDD algorithms is to be included, which offers the possibility to gain insight into the BDD synthesis algorithm. These features might be particularly useful for students to understand BDD algorithms and BDD manipulation operations, especially also for the purpose of tele-teaching. The user has the possibility to access a personalized database of circuit descriptions and BDDs, which can be subject to further research and studies during a course schedule.

Other material, as e.g. the links to relevant benchmarks, links providing download access to the relevant BDD packages, model checkers, and also related tools as SAT-solvers are added to the site as well.

5 Online Operation Experiences

OHO was released to the WWW in 1999 and soon started to raise the interest of the BDD research community. Besides the basic idea of maintaining an independent WWW-based platform for benchmark and algorithm evaluation for BDDs, the requirement for a centralized archive site dedicated to BDD research soon became obvious and the efforts of our working group were driven towards that desired goal.

In December 1999 the portal site `www.bdd-portal.org` went online and page requests starting at about 8000 pages a month, now, have reached 30000 and more page requests. This seems to be not much, compared to portal sites addressing the general public. But, for our small research community, the number is quite impressive. Several publications [MW99, MW00a, MW00b] about OHO and the related BDD portal site point out the importance of this contribution for the BDD research community on the one hand, and also for the development of specialized portal sites, providing the benefit of information structure to the WWW on the other hand.

More resources and links for BDDs will be added to our portal site, in order to further establish this portal as a central announcement and link site for all information in the WWW connected to BDDs. We are thinking about a mechanism that allows remote administration of the conference and workshop database by the organizers of the events.

More Decision Diagram heuristics and tools will be added to the section that allows online evaluation. In fact the experimentation platform should be kept current with ongoing research by our efforts as well as by external contributions. There might also be other kinds of interfaces than the WWW interface that would prove beneficial.

The portal can be accessed at `http://www.bdd-portal.org`.

6 Related Work

As far as the authors know, there is no other effort to create a specialized portal site for BDD research and there are only very few sites targeted at specific research communities. The authors also do not know of any effort to make BDD-tool functionality available via the WWW on a comparable scale. There seem to be a few efforts capable of computing only small examples, e.g. the WWW interface for the word-level DD package developed by Stefan Höreth ([HÖ1]). There, only small Boolean functions in terms of Boolean formulas can be transformed into several different types of decision diagrams that can be visualized in a way similar to the BDD Calculator of our BDD portal.

There have been efforts to use the WWW as an unified interface to a heterogenous set of EDA (Electronic Design Automation)tools. In [BBMR96] such an application has been described. The main focus there is the integration of different tools on different platforms with different data formats into a seemingly homogenous environment. The framework is intended to do actual work rather than to allow the comparison of the power of different approaches to a specific problem (like BDD heuristics) as in our case.

In [Ku97] the vision of integrating a great number of different EDA CAD services running in different places into one "global" EDA CAD system using the Internet has been described. The main techniques here are the use of proxies to abstract the actual tool being used. In [SK95] another approach to the integration of different EDA tools is mentioned. This approach is mostly centered on the notion of "active messages".

References

[BBK89] F. Brglez, D. Bryan, and K. Kozminski. Combinational Profiles of Sequential Benchmark circuits. In *Proc. IEEE Int. Symposium on Circuits and Systems*, pages 1929–1934, May 1989.

[BBMR96] A. Bogliolo, L. Benini, G. De Micheli, and B. Ricc. PPP: A Gate-Level Power Simulator - A World Wide Web Application. Technical Report Stanford Technical Report No. CSL-TR-96-691, Stanford University, 1996.

[BHSVS96] R. K. Brayton, G. D. Hachtel, A. Sangiovanni-Vincenteli, and F. Somenzi. VIS: A System for Verification and Synthesis. In *Proceedings of Computer-Aided Verification '96, vol 1102 of Lecture Notes in Computer Science*, pages 428–432. Springer, 1996.

[BLW95] B. Bollig, M. Löbbing, and I. Wegener. Simulated annealing to improve variable orderings for OBDDs,. IWLS'95, Lake Tahoe, 1995.

[DBG95] R. Drechsler, B. Becker, and N. Göckel. A Genetic Algorithm for Variable Reordering of OBDDs. IWLS'95, Lake Tahoe, 1995.

[H0̈1] Stefan Höreth. A word-level graph manipulation package. *Software Tools For Technology Transfer*, 3(2):182–192, 2001.

[HS99] H. Higuchi and F. Somenzi. Lazy Group Sifting for Efficient Symbolic State Traversal of FSMs. In *Procceedings of ICCAD 99*, pages 45–49, 1999.

[Ku97] D. Ku. Embedded Tutorial: EDA and the Network. In *Procceedings of ICCAD 97*, 1997.

[LG99] S. Lawrence and C.L. Gilles. Accessibility of information on the web. *Nature*, 400(6740), 1999.

[MS97] C. Meinel and A. Slobodová. Speeding up Variable Ordering of OBDDs. In *Proc. of IEEE ICCD'97, Austin(Texas,USA)*, 1997.

[MS98] C. Meinel and A. Slobodová. Sample Method for Minimization of OBDDs. IWLS'98, Granlibakken Resort, Lake Tahoe (Kalifornien, USA), 1998.

[MST98] C. Meinel, F. SOmenzi, and T. Theobald. Functional Decomposition and Synthesis Using Linear Sifting. In *Proc. ASP-DAC*, pages 81–86, 1998.

[MT96] C. Meinel and T. Theobald. Local Encoding Transformations for Optimizing OBDD-Representations of Finite State Machines. In *Proc. FM-CAD'96, LNCS 1166*, pages 404–418. Springer-Verlag, 1996.

[MT98] C. Meinel and T. Theobald. *Algorithms and Data Structures in VLSI Design.* Springer, 1998.

[MW99] C. Meinel and A. Wagner. OBDD Heuristics Online. In *Proc. IWLS'99, Granlibakken Resort, Granlibakken, USA*, 1999.

[MW00a] C. Meinel and A. Wagner. WWW.BDD-PORTAL.ORG. In *Proc. IWLS'2000, Dana Point, CA., USA*, pages 349–353, 2000.

[MW00b] C. Meinel and A. Wagner. WWW.BDD-PORTAL.ORG - An Electronical Basis for Cooperative Research in EDA. In *CRIS 2000, Helsinki (Finland)*, 2000.

[oCS] CMU School of Computer Science. Formal Methods - Model Checking. http://www.cs.cmu.edu/~ modelcheck/.

[SK95] M. J. Silva and R. H. Katz. The Case for Design Using the World Wide Web. In *Procceedings of the 32th Design Automation Conference*, 1995.

[Som96] F. Somenzi. Colorado University Decision Diagram Package . ftp://vlsi.colorado.edu/pub/, 1996.

[YAH] http://www.yahoo.de.