MAKING A CHOICE BETWEEN FDDS AND BDDS

IN THE PROCEEDINGS OF IWLS2005

J. E. Rice

Dept. of Math & Computer Science
University of Lethbridge
Lethbridge, Alberta, CANADA
Email: j.rice@uleth.ca

ABSTRACT

Much work has recently focused on DD variants; in particular those that use alternate decompositions such as FDDs and KDDs. The problem is that with addition of these decompositions the complexity of the structure is increased, with the possibility of additional computation time being required for determining variable ordering or decomposition choices. We propose to reduce this by adding a preprocessing step that selects, in general, whether a BDD or FDD structure is a better choice for a given function or class of functions. This decision is based on the analysis of a function's autocorrelation coefficients. Preliminary work beginning an investigation into how best to make this decision is presented.

1. INTRODUCTION

Decision diagrams (DDs) are often used for efficient representation and manipulation of Boolean functions. There are many variants of DDs, each with a variety of uses and applications. The original data structure of this type is the binary decision diagram (BDD) which was introduced by Akers [1, 2] and popularized as reduced, ordered BDDs (ROBDDs or OBDDs) by Bryant [3]. Many other types of decision diagrams have since been proposed [4].

Some of the simplest variants remain true to the original structure, simply allowing alternative decompositions of the function at each node. One example of this type of DD is the ordered functional decision diagram (OFDD) [5, 6]. It has been shown that there are certain types of Boolean functions for which there is no "good" (meaning non-exponential in size) OBDD representation [7]. Futher work has shown that for some Boolean functions the OFDD representation can be exponentially smaller than the OBDD representation, and vice versa [8, 9]. These same authors have suggested a hybrid DD consisting of decompositions types allowed in both the OBDD and OFDD, called an ordered Kronecker decision diagram (OKDD) [10], or sometimes an ordered Kronecker functional decisional diagram (OKFDD). However, in order to build a minimally (or close to) sized OKFDD, not only the variable ordering must be considered, but also the decomposition types at each node. Both of these can have a considerable effect on the size of the resulting DD, and are also very difficult problems [11]. Various heuristics have been developed to solve this problem [10].

In this work we suggest an additional preprocessing step that may be used to reduce the time required in deciding upon decomposition types for each node. This preprocessing is based on the generation of the function's autocorrelation coefficients. Based on these values we determine whether *in general* a function is better suited to a FDD type or BDD type. We cannot always achieve minimality in the KDDs thus generated, but in many cases, particularly those heavily dependent on decomposition choice and/or variable ordering, we can always improve upon the non-optimized sized, and in some cases this preprocessing can also reduce the time required to optimize the DD.

2. BACKGROUND

In this section we present some background and notation pertinent to the remainder of the paper.

2.1. Decision Diagrams

Decision diagrams were first introduced as BDDs by Lee [2] and later by Akers [1]. A decision diagram (DD) over $x_n := (x_1, x_2, \ldots, x_n)$ is a rooted directed acyclic graph $G = (V, E)$ with vertex set v containing two types of vertices, non-terminal (or non-leaf) vertices and terminal (or leaf) vertices. A non-terminal vertex v is labeled with a variable from X_n, which is the decision variable for v, and has exactly two successors denoted by $low(v)$ and $high(v)$. A terminal vertex v is labeled with a 0 or 1 and has no successors [9].

BDDs are commonly used in their canonical form of a Reduced Ordered BDD (ROBDD). It is generally under-

stood that when using the term BDD one is referring to a ROBDD. A ROBDD meets two main specifications:

- a BDD is a reduced BDD if it contains no vertex whose left subgraph is equal to its right subgraph, nor does it contain distinct vertices v and v' such that the subgraph rooted by v and v' are isomorphic.
- a BDD is an ordered BDD if on every path from the root node to an output, the variables are encountered in the specified order.

There are many variants on the basic BDD structure, including edge-inverters, edge-values, and alternate sink values. The reader is directed to [4] for an overview of DD variants. In this work we are interested in the use of alternate decompositions for the nodes. A BDD is based on the Shannon decomposition,

$$f_v = \overline{x}_i f_{x_i=0} + x_i f_{x_i=i} \tag{1}$$

while a functional decision diagram (FDD) is based on the Davio decompositions. The positive Davio decomposition is defined as

$$f = f_{x_i=0} \oplus x_i (f_{x_i=0} \oplus f_{x_i=0}) \tag{2}$$

and the negative Davio decomposition as

$$f = f_{x_i=1} \oplus \overline{x}_i (f_{x_i=0} \oplus f_{x_i=1}). \tag{3}$$

In each definition the notation $f_{x_i=1}$ ($f_{x_i=0}$) denotes the cofactor of f with respect to $x_i = 1$ ($x_i = 0$). In the case where a decision diagram is built using one or more of these decompositions then the notation used is usually that of

$$\begin{array}{ll} d_i = S: & \overline{x}_i f_{low(v)} + x_i f_{high(v)} \\ d_i = pD: & f_{low(v)} \oplus x_i f_{high(v)} \\ d_i = nD: & f_{low(v)} \oplus \overline{x}_i f_{high(v)} \end{array}$$

where d_i refers to the decomposition for the i^{th} level, v is a node in the decision diagram at level i and $f_{low(v)}(f_{high(v)})$ is the function represented by the decision diagram rooted at $low(v)$ ($high(v)$).

A Kronecker functional decision diagram (KFDD) is a DD in which a variable label and a decomposition choice from S (Shannon), pD (positive Davio) or nD (negative Davio) are associated with each node. For pD and nD nodes the reduction rules differ from those used for S nodes:

S,D: delete a node v' whose label is identical to the label of another node v and whose successors are identical to the successors of v and redirect the edges points to v' to point to v.

S: delete a node v whose two outgoing edges point to the same node v' and connect the incoming edges of v to v'.

D: delete a node v whose successor $high(v)$ points to the terminal 0 and connect the incoming edges of the deleted node to $low(v)$.

2.2. The Autocorrelation Transform

The autocorrelation function is defined as follows [12]:

$$B^{ff}(\tau) = \sum_{v=0}^{2^n-1} f(v) \cdot f(v \oplus \tau) \tag{4}$$

The superscripts ff are generally omitted. Values for τ range from 0 to $2^n - 1$ where n is the number of inputs to the Boolean function $f(X)$. The autocorrelation function or transform, when applied to the outputs of $f(X)$, transforms the outputs from a two-valued domain to the domain of the real numbers. The resulting coefficients may be referred to as the autocorrelation spectra of the function.

The outputs of the function may be encoded as 0 for false and 1 for true, or $+1$ for false and -1 for true. The first is referred to as $\{0, 1\}$ encoding, and if used results in the autocorrelation coefficients being referred to as $B(\tau)$. The $\{+1, -1\}$ coefficients are referred to as $C(\tau)$. The coefficients are generally grouped by the weight of the value of τ used in their computation; for instance, first order coefficients are all those coefficients for which $|\tau| = 1$.

3. TECHNIQUE

Work in [13] identified the following:

Theorem 3.1 *$C(\tau_i) = -2^n$ if and only if the function $f(X)$ has a decomposition*

$$f(X) = f^*(X) \oplus x_i$$

where $f^*(X)$ is independent of x_i and τ_i is a first-order autocorrelation coefficient relating to variable x_i.

From the nature of the autocorrelation transform, negative first-order coefficients indicate that the variable in question is connected to some sort of internal structure best described with the XOR operator. Our hypothesis is that this information can also be used to identify functions which are better described using a FDD over a BDD, since the nature of an efficient FDD is also inherently linked to the XOR operator.

Our technique was as follows. For each benchmark,

- generate the n first order autocorrelation coefficients,
- visually examine the coefficients to determine if they were primarily negative in value, and
- select a decomposition for the ENTIRE DD based on this examination.

4. EXPERIMENTAL RESULTS

Based on the technique above, we used PUMA [14, 15] to build KDDs for each of the benchmarks. Three sets of KDDs were generated:

- one set using the option -b 3 which used the sifting heuristic *OKFDD_DTL_Sifting* to select both variable ordering and decomposition choice,
- one set which fixed the decomposition choice at either S or pD, depending on the autocorrelation values, but allowed the sifting heuristic *OKFDD_Sifting* to select the variable ordering, and
- a comparison set which did not use any ordering or decomposition choice heuristic.

Table 1 shows the first-order autocorrelation coefficients for each of the benchmarks used. The spectra are listed in order, with the notation of $C \times j$ being used to reduce space. Where this notation is used the first number C is the coefficient value and the second number j is the number of times C appears in the coefficient list. Table 2 shows a comparison of the timings and DD-sizes for each of the generated sets of DDs.

5. DISCUSSION

When examining the first-order autocorrelation coefficients, clearly the functions ex1, ex2, parity, xor5, mylinear6 and mylinear9 result in negative coefficients. In particular many of them have coefficient values of -2^n, thus fulfilling the requirements of having a structure compatible with the $\oplus$ operator, or even of being a linear function as described in Theorem 3.1. This leads to our selection of a FDD for each of these benchmarks. Examination of Table 2 shows that for these functions, if the non-minimized version was not already minimal then a reduction in size as compared to this size was achieved. However for the benchmark ex2 although we still achieve reduction in size over the non-minimized version, the KDD version built using the decomposition and ordering heuristics achieves still further reductions.

The remaining benchmarks did not result in any negative coefficients, with the exception of a single negative coefficient for the majority benchmark. Thus for each of these we chose a BDD structure. Again, in all cases if reduction from the non-minimized version was possible, then our decision-diagram choice achieved it. However, it is interesting to note that restricting the decomposition-type choice did not always result in a lower processing time. In fact, as can be seen in Table 2, many of the benchmarks for which a BDD structure was chosen required 0.1 second longer to build and minimize, as compared to the KDD structure. We

filename	**n**	**time (sec)**	**first order {+1, -1} AC Coefficients**
9sym	9	0.1	288 288 288 288 288 288 288 288 288
9symml	9	0.1	288 288 288 288 288 288 288 288 288
cm150a	21	19.4	0 1048576x4 1966080x16
cm152a	11	0.1	0 0 0 1536x8
co14	14	0.1	16328x14
ex1	5	0.1	-32 -32 -32 -32 -32
ex2	5	0.1	-12 12 -12 12 -20
ex3	6	0.1	16 16 16 8 8
life	9	0.1	120 120 120 120 120 120 120 120 400
majority	5	0.1	20 -12 20 20 20
max46	10	0.1	264x3 288 296 288 296x2 312
mux	21	17.9	0 1048576x4 1996080x16
parity	16	5.9	-65536x16
ryy6	16	0.1	60280x4 602296x2 57976 62296x2 39256x4 40696 21256 48256
sym10	10	0.2	652x10
t481	16	0.2	24576x4 40960x8 24576x4
xor5	5	0.5	-32 -32 -32 -32 -32
mylinear6	6	0.1	-64 -64 -64 -64 -64 -64
mylinear9	9	0.1	-512x9
6var1clique	6	0.1	48x6
6varxorclique	6	0.1	16 16 32 16 32 16
10var1clique	10	0.1	984x10
10varxorclique	10	0.2	128x10
15var1clique	15	0.1	32688x15
15varxorclique	15	2.3	2048 0 2048 0 2048 4096 2048x9

Table 1. First order autocorrelation spectra for each of the benchmarks.

filename	choice/timing/size	kdd timing/size	no min. timing/size
9sym	bdd/0.8/24	0.7/24	0.7/24
9symml	bdd/0.8/24	0.7/24	0.7/24
cm150a	bdd/46.6/32	313.9/25	1.9/131070
cm152a	bdd/0.8/15	0.7/12	0.7/382
co14	bdd/0.9/26	0.7/26	0.7/26
ex3	bdd/0.9/8	0.7/8	0.7/10
life	bdd/0.9/25	0.7/25	0.7/37
majority	bdd/0.8/7	0.7/7	0.7/8
max46	bdd/0.9/74	0.7/74	0.7/74
mux	bdd/28.1/32	158.8/25	2.8/131070
ryy6	bdd/0.9/21	0.7/21	0.7/23
sym10	bdd/0.9/30	0.7/30	0.7/30
t481	bdd/0.9/20	0.8/18	0.7/20
ex1	fdd/0.7/5	0.7/5	0.7/5
ex2	fdd/0.7/9	0.7/8	0.7/12
mylinear6	fdd/0.7/6	0.7/6	0.7/6
mylinear9	fdd/0.7/16	0.7/16	0.7/16
parity	fdd/1.6/16	1.6/16	1.6/16
xor5	fdd/0.7/5	0.7/5	0.7/5
6var1clique	bdd/1.0/11	0.7/10	0.7/16
6varxorclique	bdd/0.9/10	0.7/8	0.8/11
10var1clique	bdd/0.8/41	0.7/41	0.7/47
10varxorclique	bdd/0.9/79	0.8/19	0.7/182
15var1clique	bdd/0.9/92	0.7/93	0.7/106
15varxorclique	bdd/2.1/414	1.9/35	1.7/558

Table 2. Comparisons of DD-sizes (nodes) and time to build (sec.) for the autocorrelation-selected decomposition type, using the DTL and ordering heuristic, and using no ordering or decomposition choice heuristic.

suspect that this is due to some overhead required in restricting the DD type to a BDD.

Of particular interest are the benchmarks cm150a and cm152a. If no minimization is performed then the number of nodes in the KDD is 131070 and 782, respectively. The KDD heuristics can bring this down to 25 and 12, but the time required, particularly for cm150a, is significant (313.9 seconds). However, if we restrict the decomposition type to Shannon and perform ordering heuristics we can minimize to 32 and 15 nodes, respectively, requiring (for cm150a) only 46.6 seconds. The same test, only using a FDD structure, was also performed, resulting in 32 nodes and requiring 64.9 seconds; still an improvement over the non-minimized KDD in terms of size, and an improvement over the more than 5 minutes required for the full decomposition and ordering heuristics, although not as good in either case as the BDD results.

The reader will likely have noticed that there are a number of circuits included that are not part of the standard benchmark set: mylinear6, mylinear9, and 4 clique circuits. These were created in an effort to build circuits that would have a definite advantage when built using a BDD or FDD. For the linear circuits this was not as successful as we had anticipated. The DD representations for the linear circuits were equally small using either a FDD or BDD structure. This leads us to the conclusion that although small autocorrelation coefficients certainly have some relationship to an XOR structure, this does not always indicate that a FDD representation will be advantageous over a BDD representation. This is an area where future investigation will continue.

The 6 clique circuits also presented some interesting developments. Designed based on the results in [9], the authors of [9] identify the following theorem:

Theorem 5.1 *For any order π of the variables $x_{i,j}$ and and decomposition type list $d = (d_{1,2}, \ldots, d_{n-1,n}) \in \{pD, nD\}^{n(n-1)/2}$ it holds:*

1. *$1 - cl^d_{n,3}$ has OBDDs of size $O(n^5)$ but only free FDDs of size $2^{\Omega(n^2)}$.*
2. *$\oplus - cl^d_{n,3}$ has OFDDs of size $O(n^3)$ but only free BDDs of size $2^{\Omega(n^2)}$.*

We therefore created the circuits 6var1clique, 6varxorclique, 10var1clique, 10varxorclique 15var1clique and 15varxorclique based on the definitions of the $1 - cl^d_{n,3}$ and $\oplus - cl^d_{n,3}$ functions as given in [9]. The intent was to test first of all if, for these relatively small-sized circuits, the theoretical results were evident, and secondly, whether the autocorrelation coefficients would provide any indication of these theoretical results. Table 3 shows the results listed as the timing in seconds/number of nodes for a BDD, FDD, and

KDD structure, each using the ordering and/or decomposition choice heuristics, with the last column listing the KDD results if no heuristics are used. It is clear from Table 3

filename	bdd results	fdd results	kdd results	kdd results with no heuristics
6var1clique	1.0/11	0.7/13	0.7/10	0.7/16
6varxorclique	0.9/10	0.7/8	0.7/8	0.8/11
10var1clique	0.8/41	0.7/96	0.7/41	0.7/47
10varxorclique	0.9/79	0.7/20	0.8/19	0.7/182
15var1clique	0.9/92	0.8/645	0.7/93	0.7/106
15varxorclique	2.1/414	1.8/33	1.9/35	1.7/558

Table 3. Results (timing, sec/number of nodes) for files representing $1 - cl^d_{n,3}$ and $\oplus - cl^d_{n,3}$ functions as defined in [9].

that a BDD is the better choice for the 1-clique functions while a FDD is the better choice for the $\oplus$-clique functions. In fact, for the larger functions (10var, 15var) the BDD/FDD representation is smaller than the KDD representation, which seems unusual. However, the point to be noted here is that the autocorrelation coefficients as listed in Table 1 did not identify, via the existence of negative coefficients, the fact that a FDD structure should be chosen for the $\oplus$-clique functions. Either our subset of coefficients does not contain the information required, or our identification of negative coefficients is not sufficient for determining the choice between a BDD or FDD. Table 4 shows the complete autocorrelation spectra for the 6 clique functions. Note that the ordering of the values is not preserved, unlike in Table 1. This is useful as an analytic tool to determine an overall pattern, but as a preprocessing tool, particularly for use on large functions, generation of the complete spectra is infeasible due to the fact that each function results in 2^n autocorrelation coefficients, and generation of these can be extremely computationally time-consuming. However, for our purposes of investigation, this table highlights some very interesting things. In particular, we can identify

filename	n	complete AC spectra
6var1clique	6	64x1 48x57 56x6
6varxorclique	6	64x1 16x48 0x9 32x6
10var1clique	10	1024x1 994x993 992x15 1000x15
10varxorclique	10	1024x1 128x100 0x315 64x552 -64x40 256x15 -256x1
15var1clique	15	32768x1 32688x32622 32704x45 32696x100
15varxorclique	15	32768x1 2048x162 0x16182 -2048x3 512x8058 4096x28 1024x757 -1024x99 -512x1142 -256x1956 256x4380

Table 4. The complete autocorrelation spectral for each of the clique functions.

filename	sum of ac spectra	avg of ac spectra
6var1clique	3072	48.76
6varxorclique	960	15.24
10var1clique	1016922	994.06
10varxorclique	48152	48.05
15var1clique	1071089216	32688.05
15varxorclique	5275648	161

Table 5. The sums and averages for the autocorrelation spectral of the 1-clique and $\oplus$-clique functions.

a significant difference in the autocorrelation values in the 1-clique functions as compared to the $\oplus$-clique functions. The 1-clique functions result in quite high values for the coefficients, in many cases close to the maximum value of 2^n. The $\oplus$-clique functions result in much lower values, close to or below zero in many cases. The exception is, of course, the first coefficient, or zero coefficient, which always has the value 2^n. Table 5 gives the sums and averages of the coefficients for these files. Note that the zero coefficient is disregarded in both computations. For all the 1-clique functions, the average value is remarkably close to 2^n, while for the $\oplus$-clique functions the average is significantly lower. The hypothesis of lower coefficient values indicating a choice of a FDD structure is borne out by these values.

6. CONCLUSIONS AND FUTURE WORK

In this work we propose the use of a preprocessing step to determine if a function can be better represented by a FDD or BDD structure. The goal is to remove the necessity of determining a decomposition-type list after the DD is built, thus reducing the time needed for heuristics to determine this and to possibly improve the performance of variable ordering heuristics. Previous work in [13] has indicated that there is much useful information present in a function's autocorrelation coefficients, and suggested some applications. We propose that the autocorrelation spectra of a function can also be used in determining the best choice of FDD or BDD structure.

Results suggest that in many cases we can pre-determine a good structure by examination of only the n first-order autocorrelation coefficients. However, when functions known to have a good FDD structure and poor BDD structure such as the $\oplus - cl^d_{n,3}$ functions as proposed in [9] are tested, preprocessing based on only the first-order coefficients does not identify the correct choice of FDD. However, the current heuristic for this decision is not only based on only a small subset of the autocorrelation coefficients, but is also based on simply identifying yes/no whether a majority of this subset consists of negative values. Clearly many refinements are possible.

This work has presented some investigations that illus-

trate that the hypothesis in [13] on which this theory is based does have validity in this application. However, a number of hurdles must be overcome. Of primary concern, any heuristics used must be based on a *subset* of the autocorrelation spectra; it is infeasible to compute and examine 2^n values for even small values of n. However, to date only the subset consisting of first-order coefficients has been examined, and so there are many other possibilities to consider. Additionally, techniques for approximating the autocorrelation spectra have been published [16], and these may be of use in this application. Future work must address how best to compute the most useful information from the autocorrelation coefficients, and how to make use of this information. An additional area of interest consists of identifying functions which result in some identifiable properties within the autocorrelation coefficients, as begun in [17], and how these properties may then be used in the context of this work.

7. REFERENCES

[1] S. Akers, "Binary Decision Diagrams," *IEEE Trans. on Comp.*, vol. C-27, no. 6, pp. 509–516, June 1978.

[2] C. Y. Lee, "Representation of Switching Circuits by Binary Decision Diagrams," *Bell System Technical Journal*, pp. 958–999, 1959.

[3] R. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. on Comp.*, vol. C-35, no. 8, pp. 677–691, Aug. 1986.

[4] R. Drechsler and D. Sieling, "Binary Decision Diagrams in Theory and Practice," *Int. Journal on Software Tools for Technology Transfer (STTT)*, pp. 112–136, May 2001.

[5] U. Kebschull, E. Schubert, and W. Rosenstiel, "Multilevel Logic Synthesis Based on Functional Decision Diagrams," in *Proceedings of the 3rd European Conference on Design Automation*, 1992, pp. 43 – 47.

[6] U. Kebschull and W. Rosenstiel, "Efficient Graph-based Computation and Manipulation of Functional Decision Diagrams," in *Proceedings of the 4th European Conference on Design Automation*, 1993, pp. 278–282.

[7] R. E. Bryant, "On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application to Integer Multiplication," *IEEE Trans. on Comp.*, vol. 40, no. 2, pp. 205–213, Feb. 1991.

[8] B. Becker, R. Drechsler, and R. Werchner, "On the Relation Between BDDs and FDDs," *Information and Computation*, vol. 123(2), pp. 185–197, 1995.

[9] B. Becker, R. Drechsler, and M. Theobald, "OKFDDs versus OBDDs and OFDDs," in *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, 1995, pp. 475–486.

[10] R. Drechsler and B. Becker, "Ordered Kronecker Functional Decision Diagrams – A Data Structure for Representation and Manipulation of Boolean Functions," *Trans. on CAD*, vol. 17, no. 10, pp. 965–973, October 1998.

[11] R. Drechsler, B. Becker, and A. Jahnke, "On Variable Ordering and Decomposition Type Choice in OKFDDs," *IEEE Trans. on Computers*, vol. 47, no. 12, pp. 1398–1403, Dec. 1998.

[12] M. Karpovsky, *Finite Orthogonal Series in the Design of Digital Devices*. John Wiley & Sons, 1976.

[13] J. E. Rice, "Autocorrelation Coefficients in the Representation and Classification of Switching Functions," Ph.D. dissertation, University of Victoria, 2003.

[14] A. Hett, "PUMA," 1995, decision diagram software, see http://ira.informatik.uni-freiburg.de/software/puma/pumamain.html.

[15] R. Drechsler and B. Becker, "Dynamic Minimization of OKFDDs," in *Proceedings of the International Conference on Computer Design: VLSI in Computers and Processors (ICCD)*, 1995, pp. 602–607.

[16] D. Wessels, "Efficient Approximation of Spectral and Autocorrelation Coefficients," in *Proceedings of the IEEE Region Ten Conference on Digital Signal Processing Applications (TENCON)*, 1996.

[17] J. E. Rice and R. Jansen, "Symmetrical, Dual and Linear Functions and Their Autocorrelation Coefficients," submitted to IWLS2005.