Split Multi-terminal Binary Decision Diagrams

Ilya Levin
Tel Aviv University, Ramat Aviv,
Tel Aviv, 69978, Israel
e-mail: ilia1@post.tau.ac.il

Osnat Keren
Bar Ilan University, Ramat Gan,
52900, Israel,
e-mail: kereno@eng.biu.ac.il

Abstract

A new data structure called Split Multi-terminal Binary Decision Diagrams (Split MTBDD) is introduced for representing Multi–Output logic Functions (MOF). Split MTBDDs are efficient for some functions where conventional BDDs are not. A Split MTBDD comprises interconnected MTBDD components, each associated with a "dichotomic fragment". The "dichotomy" reflects cognitive patterns introduced by the designer of a MOF specification. The paper describes a method of transforming an arbitrary MOF into a corresponding Split MTBDD. Experimental results indicate that Split MTBDDs are more compact than conventional MTBDDs for many benchmarks. Criteria for prediction of the Split MTBDD compactness are formulated and justified.

1. Introduction

Many problems in VLSI-CAD and other areas of computer science can be formulated in terms of Boolean functions. The central issue in providing computer-aided solutions to these problems is to find a compact representation for Boolean functions, where basic Boolean operations and the equivalence check can be efficiently performed. The requirements of compactness and manipulability are generally conflicting. Currently, Binary Decision Diagrams (BDDs) serve the most popular compromise between these conflicting requirements. In a large number of practical cases, a BDD representation of a Boolean function is exponential in the number of primary inputs. This fact limits complexity of the problems which can be solved using BDDs.

Representation of a system of Boolean functions (usually called a Multi-output function (MOF)) by a truth table, as well as representation by a set of disjoint cubes belongs to a *declarative* type of specification [1, 2]. An alternative, *procedural* representation of a MOF is a tree-like algorithmic specification, for example - a Multi Terminal BDD (MTBDD).

The simplest way to specify and/or to represent a Boolean function or a MOF is a truth table. The main advantage of the representation by a truth table is that such a representation is canonical. Moreover, it is ideal from the point of its completeness, non-contradictive character and manipulability, but impractical due to its exponential complexity.

A proper specification of a MOF has to be non-contradictory. It means that all product terms of the MOF, initiating different outputs, have to be disjoint. The majority of relevant studies are based on this property.

The initial point of this paper is an idea that an arbitrary MOF has an additional inherent feature which, if taken into account in synthesis, may improve the resulting overhead. The majorities of logic specifications of MOFs are developed by humans and, as a result, inherit some humans' thinking templates. Actually, the thinking templates usually have a tree-like structure and, fortunately, may be formalized. Fragments of a MOF, having the tree-like structure, will be called *dichotomic fragments.*

We say that a set of product terms forms a dichotomic fragment, if the set is straightforwardly mappable into an MTBDD. The mapping of the dichotomic fragment can be easily preformed by the Shannon expansion procedure. The dichotomic property guarantees that there exists a Shannon expansion process that will not bring additional product terms to the initial MOF. It means that the paths of the MTBDD are in the one-to-one correspondence with respective product terms in the specification.

We hypnotize that it is possible to directly formalize and implement the thinking templates comprising dichotomic fragments. We study cases where an MOF is represented by a conventional MTBDD. Our hypothesis is that the MOF can be more efficiently defined by a set of dichotomic fragments. As a result, the whole MOF would be considered a set of sub-functions, the logical sum of which would be equal to the output of the initial MOF.

To describe such a MOF, we introduce a new data structure– a Split MTBDD. The Split MTBDD consists of a number of component conventional MTBDDs having separate roots and non-shared nodes. These component MTBDDs correspond to our dichotomic fragments. Output vectors of each of the fragments are logically summed. We propose an algorithm for transforming a MOF into its Split MTBDD form. We check whether the structure of Split MTBDDs is more compact than the corresponding conventional MTBDD.

We perform a number of experiments with standard benchmarks and compare the obtained results with known MTBDD parallel decomposition method [3]. Based on the benchmark results, we study the efficiency of the Split MTBDD data structure and its correlations with different characteristics of the MOFs.

The paper is organized as following. Section 2 presents theoretical fundamentals of Split MTBDDs. Decomposition algorithms for transformation MOF into the Split MTBDD form described in Section 3. Experimental results is presented and discussed in Section 4. Conclusions are given in Section 5.

2. Split MTBDDs

Multi-Terminal Binary Decision Diagram (MTBDD) [4. 5. 6] is a rooted directed acyclic graph with two types of nodes – terminal and non-terminal. A non-terminal node is labeled with an input variable and has two successors. A terminal node is labeled with a variable of an output finite set.

A multiple-output function can be represented in the MTBDD form. To transform an arbitrary MOF into its MTBDD form, a well known Shannon expansion procedure can be applied. Efficiency of the representation is usually evaluated by a number of non-terminal nodes in the corresponding MTBDD [7].

The main problem of the MTBDD representation is a problem of complexity. There are efficient methods for Boolean functions decomposition [8, 9, 10], as well as methods for decomposition of BDDs [11, 12, 13]. Nevertheless, the problem of efficient representation of MOFs still remains actual and challengeable [14].

We extend the concept of MTBDD by introducing a new data structure –a Split MTBDD. We define the Split MTBDD by introducing a parallel connection of several conventional MTBDDs as it is shown in Figure 1:

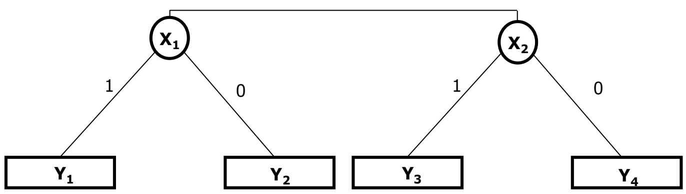

Figure 1. Parallel connection of two conventional MTBDDs.

The following example illustrates the construction of a Split MTBDD.

Example 2. Let a MOF is defined in the PLA form as follows:

x0	x1	x2	x3	y1	y2	y3	y4
1	0	*	1	0	0	1	0
1	0	1	*	0	1	1	0
1	1	*	1	1	1	0	0
1	1	1	*	1	0	0	0
0	0	*	1	1	0	1	0
0	0	1	*	1	0	1	1
0	1	*	1	0	0	1	1
0	1	1	*	0	1	0	1

It comprises one dichotomic fragment (defined by the variables x_0, x_1) and a number of non-dichotomic "tails". The corresponding MTBDD and Split MTBDD representations are shown in Figs. 2 and 3. Notice that we use decimal numbers of output vectors within the terminal nodes of both of BDDs.

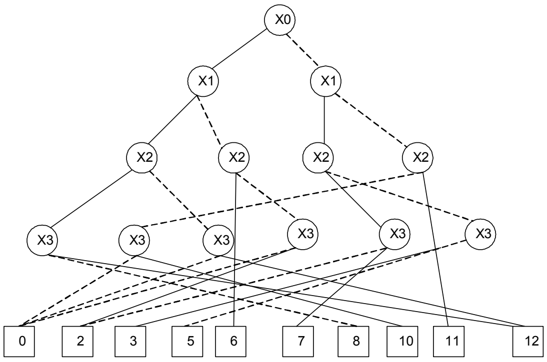

Figure 2. MTBDD for Example 2.

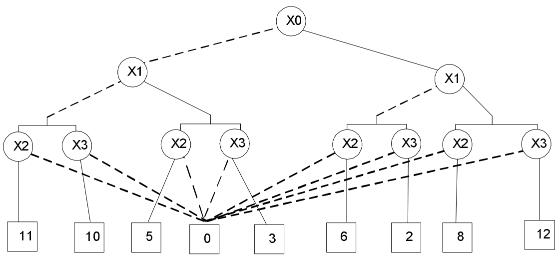

Figure 3. Split MTBDD for Example 2.

The MTBDD for our example has 13 non-terminal nodes (NTN) and 10 terminal nodes (TN). The Split MTBDD has 11 NTN and 8 TN.

3. Decomposition Algorithm

An algorithm for constructing a parallel network of component MTBDDs was described in [3]. It comprised a step of sequentially extracting component MTBDDs from a MOF and a step of constructing a network having a minimized total number of non-terminal nodes. As the main criterion for selecting the components, the algorithm uses a so-called "density factor", which is a percent of cells not comprising "don't-cares" (DC) in a corresponding PLA fragment.

In the present paper, we propose an algorithm for transforming a MOF into the Split MTBDD form.

According to the nature of a Split MTBDD being a set of interconnected dichotomic fragments, we propose a different criterion for selecting components of the Split MTBDD. Obviously, a dense fragment, which was preferred in the previous method, not always has the dichotomy property. Sometimes, the density and the dichotomy properties are contradictory.

The proposed new algorithm of constructing the Split MTBDD not only changes the main criterion of selecting the components. Unlike the "density-oriented" previous algorithm that uses heuristics for selecting the components, the new one has the exact nature. The proposed new algorithm gives exact estimation of a "rank of dichotomy" of an arbitrary MOF, which is further reflected by the number and the size of the dichotomic fragments in the corresponding Split MTBDD.

Both the "density-oriented" and the newly introduced "dichotomy-oriented" algorithms use one and the same general decomposition method. The general method is based on partitioning of the set of cubes of a MOF into a number of components. This partitioning is performed recursively. On each step of the recursive procedure, the corresponding MOF component is partitioned into two subsets: a subset of components having a common header and a remainder. Each of the components with the common header is implemented as a conventional MTBDD. The main concern of the general decomposition method is searching for optimal "common headers", for obtaining optimal resulting component MTBDDs.

As mentioned above, the proposed "dichotomy-oriented" algorithm incorporates the general decomposition method. The proposed algorithm begins its operation by detecting the dichotomy within every PLA fragment to be decomposed. The dichotomy is detected if at least one column without "don't cares" (DC) exists in the fragment. A block-scheme of the algorithm is shown in Figure. 3.

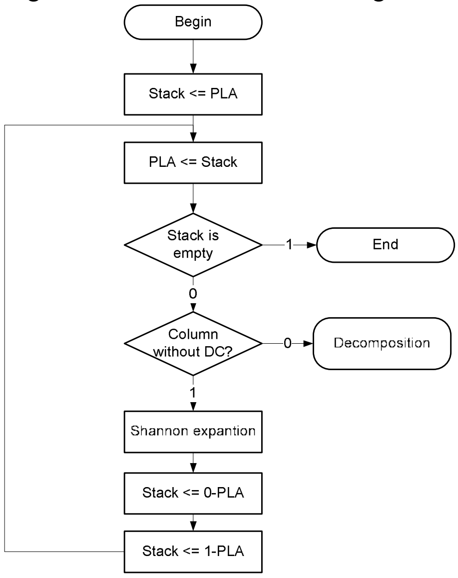

Figure 3. The block-scheme of the "dichotomy" oriented decomposition algorithm

The algorithm uses a stack for implementation of the Shannon expansion, if needed for a PLA fragment. The expansion is performed if at least one column without DC exists in the fragment. If all columns of the PLA comprise DC, the general decomposition method will be performed. Notice, that unlike the "density-oriented" algorithm, the new "dichotomy-oriented" algorithm a) comprises the Shannon expansion within the general decomposition procedure and b) repeats the expansion whenever possible. By this we guarantee that all dichotomic fragments of the initial MOF are separated and that an optimal "dichotomy oriented" Split MTBDD is constructed.

4. Experimental Results

In this section, we present experimental results obtained on a number of industrial benchmarks. Experiments have been performed by software implementing the propose algorithms. The results of the experiments are presented in Tables 1 and 2.

Table 1. Benchmark results of the decomposition

No	Benchmark	Inputs	Outputs	Products	Density	Nmt	Ndich	Ndens
1	dip	5	4	5	0.48	16	7	7
2	con1	7	2	9	0.3651	18	16	16
3	f2	4	4	12	0.75	14	21	21
4	xor5	5	1	16	1	9	9	46
5	dc1	4	7	9	0.7778	11	16	14
6	wim	4	7	9	0.5	13	9	9
7	dk27	9	9	10	0.3444	46	17	20
8	rd53	5	3	32	0.9	15	53	57
9	alu1	12	8	19	0.1798	1195	32	31
10	sqn	7	3	38	0.6917	61	98	106
11	squar5	5	8	32	1	30	30	67
12	misex1	8	7	32	0.4766	19	30	32
13	inc	7	9	34	0.7941	35	35	76
14	dc2	8	7	40	0.6656	128	77	88
15	z4	7	4	59	0.6102	76	146	146
16	root	8	5	57	0.6491	75	147	149
17	sqr6	6	12	50	0.6733	63	97	99
18	9sym	9	1	87	0.6667	39	265	274
19	adr4	8	5	75	0.5667	158	177	172
20	radd	8	5	75	0.5667	171	175	168
21	alu3	10	8	66	0.4227	397	162	163
22	alu2	10	8	68	0.3941	408	167	165
23	f51m	8	8	76	0.5263	255	163	163
24	5xp1	7	10	75	0.5638	127	138	139
25	rd73	7	3	141	0.8511	28	297	360
26	dist	8	5	121	0.7293	115	302	307
27	mlp4	8	8	128	0.7305	214	306	321
28	bw	5	28	65	0.7385	23	79	84
29	clip	9	5	167	0.5908	269	399	401
30	b12	15	9	431	0.286	238	184	191

Table 2. Algorithms runtime

Benchmark	Tmt	Tdich	Tdens
dip	0.16	0.7	7
con1	0.19	1.4	11
f2	0.15	1.7	15
xor5	0.18	0.1	9
dc1	0.14	1	12
wim	0.16	1	7
dk27	0.24	1.5	13
rd53	0.22	4	45
alu1	5.17	3.7	25
sqn	0.33	4.9	71
squar5	0.21	0.2	30
misex1	0.16	2.6	28
inc	0.23	0.2	35
dc2	0.53	5.7	75
z4	0.56	12	126
root	0.42	11	110
sqr6	0.37	7.9	79
9sym	1.06	23	182
adr4	0.97	16	151
radd	0.93	17	146
alu3	2.08	14	129
alu2	1.86	14	140
f51m	0.94	14	122
5xp1	0.6	13	106
rd73	0.75	121	321
dist	0.88	32	240
mlp4	1.17	62	228
bw	0.27	9.8	74
clip	2.05	66	333
b12	2.26	28	168

First five columns of Table 1 comprise titles of the benchmarks and their parameters. The 6-th column indicates the density of the corresponding benchmark. Columns 7, 8 and 9 indicate the number of nodes in the MTBDD and in two alternative Split MTBDD representations – column 8 relates to that obtained by the dichotomy oriented algorithm, and column 9 – to that obtained by the density oriented algorithm.

The results indicate that, approximately in half of cases, the proposed Split MTBDD gives more compact representation in comparison with the conventional MTBDD. Both the newly proposed "dichotomy oriented" algorithm and the "density oriented" algorithm give pretty close results. It means that both of them are suitable for constructing Split MTBDDs. The results indicate the high robustness of the proposed general algorithm, and confirm the rightness of the methodology.

When studying regularity in transforming a MOF into the Split MTBDD form by analyzing the benchmark results, a number of different parameters were used. Finally, the *density* of the benchmark was chosen as the most important parameter. To demonstrate the correlation of the results with the benchmark density, we introduce the following additional criteria for our data:

- *Normalized density* D_n : The normalized density is defined as $D_n = D / I$, where D - the density, I - the number of a MOF inputs;
- *Normalized closeness* Q : The normalized closeness Q reflects the relation between the numbers of non-terminal nodes in the Split MTBDD $Ndich$ and the number of non-terminal nodes in the conventional MTBDD Nmt, $Q = 2\left(Ndich - Nmt\right) / \left(Ndich + Nmt\right)$.

To obtain a better visualization, an auxiliary function for D_n is used: $D' = 10D_n - 1$. Values of D' and Q for our benchmarks (numbers on the horizontal axis) are presented in the graph shown in Figure 4.

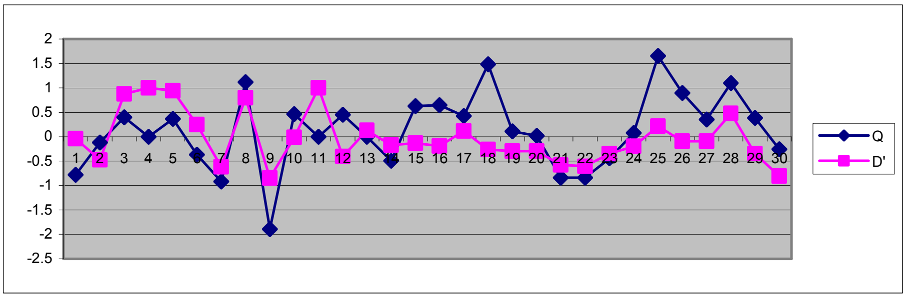

Figure 4. Benchmark results. Graph of the correlation between the density D' and the difference in number of non-terminal nodes of MTBDD and the Split MTBDD Q'.

The graph indicates a correlation between the density of the initial PLA (a benchmark in our case), and the compactness of the proposed Split MTBDD representation. Specifically, the smaller density corresponds to the higher efficiency of the proposed algorithm (the graph is drawn for the dichotomy oriented algorithm).

Table 2 compares runtimes of the proposed algorithms. Obviously, the transformation PLA – MTBDD performed by the standard Shannon expansion is very fast. Both of the algorithms for constructing Split MTBDDs are much slower. Between them, the dichotomy oriented algorithm is significantly faster, which may look surprising taking into account the fact that the dichotomy oriented algorithm includes the density oriented algorithm as its component. The high speed of the dichotomy oriented algorithm can be explained by the following. The dichotomy oriented algorithm includes the Shannon expansion, which extracts portions of the dichotomy from the MOF thus making is smaller, which is in turn decreases the run time of the algorithm.

5. Conclusions

We have proposed a new data structure for representing Boolean functions, called a Split Multi Terminal BDD. This data structure has a number of attractive properties. Split MTBDDs, in comparison with the conventional BDDs, are easier to construct and more compact for a wide class of functions. The main concern in constructing the Split MTBDDs is the dichotomy property of specifications that relates to a tree-like representation of logic functions.

We have developed a method for optimized transforming of an arbitrary multi-output function MOF into its Split MTBDD form. The method is based on partitioning the MOF into a set of dichotomic fragments.

The proposed algorithm was tested on a number of standard benchmarks. Results of the experiments were compared with the known "density-oriented" algorithm which did not care about the dichotomy property. The results of the experiments can be summarized as follows:

1. In comparison with the conventional MTBDD, the Split MTBDD allows reducing the number of non-terminal nodes required for a BDD-like representation of Boolean functions.
2. The proposed "dichotomy-oriented" algorithm and the known "density-oriented" algorithm provide close results in the total number of non-terminal nodes. This fact characterizes robustness and effectiveness of the proposed general approach.
3. The new dichotomy-oriented algorithm seems preferable due to its runtime parameters.
4. A correlation has been found between a number of nodes in the final BDD and a function of normalized density for the initial specification of a MOF. The correlation allows predicting, which representation form (Conventional or Split) is preferable for a specific MOF.

References

[1] I. Levin, D. Mioduser: A Multiple-Constructs Framework for Teaching Control Concepts, IEEE Transactions of Education, 39(4), 1996, 488-496.

[2] A. A. Zakrevskij: Common Logic Approach to Data Mining and Pattern Recognition. In book: E. Triantaphyllou, G. Felici, eds. Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques, Springer, 2006, pp. 1-44.

[3] I. Levin, O. Keren, V. Ostrovsky, G. Kolotov: Concurrent Decomposition of Multi-terminal BDDs. Proc. of 7th International Workshop on Boolean Problems, Freiberg, September 2006, pp. 165-169.

[4] I. Bahar, E. Frohm, C. Gaona, G. Hachtel, E. Macii, A. Pardo, F. Somenzi: Algebraic decision diagrams and their applications, Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design, 1993, pp. 188 – 191.

[5] E. M. Clarke, M. Fujita, P. C. McGeer, K. McMillan, and J. Yang: Multi-terminal binary decision diagrams: An efficient data structure for matrix representation, IWLS'93: International Workshop on Logic Synthesis, Lake Tahoe, CA, May, 1993, pp. 6a:1–15.

[6] C. Baier and E. Clarke: The Algebraic Mu-Calculus and MTBDDs. In 5th Workshop on Logic, Language, Information and Computation (WoLLIC'98), 1998, pp. 27–38.

[7] S. Yanushkevich, M. Miller, V. Shmerko, R. Stankovic: Decision Diagram Techniques for Micro- and Nanoelectronic Design Handbook, CRC, Taylot&Francis Group, 2005, 952 pages.

[8] R. L. Ashenhurst: The decomposition of switching functions, Bell Laboratories' Report, Vol. Vol. 1, 1953, pp.II-1-II-37.

[9] H. A. Curtis: A New Approach to the Design of Switching Circuits, 466-500. 1962, D. Van Nostrand Co., Inc., Princeton.

[10] S. Hassoun and T. Sasao, (editors): Logic Synthesis and Verification. The Springer International Series in Engineering and Computer Science, Vol. 654, 2002, 472 p.

[11] M. Perkowski and S. Grygiel: A Survey of Literature on Function Decomposition. Technical Report, 1995, Portland State University.

[12] A. Narayan, J. Jain, M. Fujita and A. Sangiovanni-Vincentelli: Partitioned ROBDDs-A Compact, Canonical and Efficiently Manipulable Representation for Boolean Functions, Proc. of the International Conference on Computer-Aided Design, 1996, pp. 547-554.

[13] B. Steinbach and C. Lang: Exploiting Functional Properties of Boolean Functions for Optimal Multi-Level Design by Bi-Decomposition. Artificial Intelligence Review, Kluwer Academic Publishers, 2003, pp. 319–360.

[14] R. Brayton: The future of logic synthesis and verification. In: Logic Synthesis and Verification, Kluwer Academic Publishers Norwell, MA, 2002, pp. 403 – 434.