Recent Advent in Spectral Techniques in Binary and Multiple-Valued Switching Theory

Mark G. Karpovsky
Dept. of Electlectrical and
Computer Engineering
Boston University
8 Saint Mary's Street
Boston, Ma 02215, USA

Radomir S. Stanković
Dept. of Computer Science
Faculty of Electronics
Beogradska 14
18 000 Niš
Yugoslavia

Claudio Moraga
Dept. of Computer Science and
Computer Engineering
Dortmund University
44221 Dortmund
Germany

Abstract

This paper presents reviews and briefly discuss recent development in spectral methods in switching and multiple-valued logic theory and the design of digital systems.

1 Introduction

In switching theory and multiple-valued (MV) logic, spectral techniques are used as alternative methods providing for efficient solutions of different tasks where the classical approaches appear inefficient or complex for applications. In this paper, we review and discuss some recent research work in this area.

2 Group Theoretic Approach to Spectral Techniques

In this approach, discrete functions are considered as a mapping $f : G \rightarrow P$, where G is a finite not necessarily Abelian group, and P is a field that may be the complex field C or a finite (Galois) field $GF(p)$. The set of all functions on G into P is denoted by $P(G)$, and the structure of a linear vector space is assumed for $P(G)$.

Switching functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$, and MV functions $f : \{0, \ldots, p-1\}^n \rightarrow \{0, \ldots, p-1\}$, with n variables, are considered as particular examples, when P is $GF(2)$ and $GF(p)$, and G is the direct product of n support groups of $GF(2)$ and $GF(p)$, respectively. We denote these domain groups by C_2^n and C_p^n, assuming the addition in $GF(2)$ and $GF(p)$ componentwise as the group operations. Alternatively, switching and MV functions are considered as subsets of complex-valued functions in $C(C_2^n)$ and $C(C_p^n)$, in which case the logic values for arguments and function values, conveniently represented by 0,1 and $0, \ldots, p-1$, with 0 assigned to the unity in G, are considered as the corresponding integers.

In $P(G)$, Fourier analysis is defined in terms of unitary irreducible group representations of G over P, which in the case of Abelian groups reduce to group characters, providing that some restrictions between the cardinality of G and characteristic of P ensuring the existence of a Fourier transform are satisfied.

The group theoretic approach permits an easy way to derive relationships among the most often used transforms in this area, the Walsh, arithmetic, and Reed-Muller transforms. Generalizations to MV functions and related transforms are straightforward [3].

The discrete Walsh transform is the Fourier transform in $C(C_2^n)$, since is defined with respect to the discrete Walsh functions which are group characters of C_2 over C, in the same way as in the Fourier transform, the exponential functions e^{jwx}, $w, x \in R$, are the group characters of the locally compact Abelian group of the real numbers R. The Walsh functions are product of elements of the incomplete in $C(C_2^n)$ set of n Rademacher functions, which are switching variables in $(0, 1) \rightarrow (1, -1)$ coding. Thus, in this coding, x_i maps into $1 - 2x_i$, $x_i \in \{0, 1\}$. Due to that, if the Walsh functions are represented as product of first order polynomials in terms of switching variables, then after a simple calculation, the Walsh series expansion, in terms of Walsh coefficients, converts into the arithmetic expression, in terms of arithmetic transform coefficients. The calculation consist of determination of products of switching variables by expanding the parenthesis, expressing the Walsh coefficients in terms of function values, and reassignment of coefficient to the products. Reed-Muller

Table 1. Number of coefficients for $f(x) = sin(x)$.

Walsh	Arithmetic	Arithmetic-Haar	Fourier
65538	59930	55481	18144

expressions follow by replacing the field C by $GF(2)$, which means, by recalculating the coefficients modulo 2 and replacing the additions and subtractions by modulo 2 addition (EXOR). Relationships to the discrete Haar transform are straightforward, if the discrete Haar functions are expressed in terms of switching variables.

In this setting, the use of the non-Abelian quaternion group Q_2, see for example [15], can be considered as the recoding of triplets of switching variables $(x_i x_j x_k)$ in C_2^3 by a single variable in Q_2. Thus, $C(C_2^{3r})$ is replaced by $C(Q_2^r)$. The relationship to the Fourier transform on Q_2, which is defined in terms of unitary irreducible representations of Q_2 over C consisting of four one-dimensional and a single two dimensional representation, is straightforward, if the group representations are expressed in terms of switching variables. In the same way, an extension to the arithmetic-Haar transform in $C(Q_2^r)$ is achieved [**?**].

The following example shows that such replacement of the domain group may be useful in practice.

Example 1 *Consider a function $f(x) = sin(x)$, where $f(x)$ and x are represented as 16-bit binary numbers. This function can be considered as a multiple-output switching function with 16 inputs x_i and 16 outputs f_i and can be represented by an integer equivalent function $f_Z = \sum_{i=1}^{16} f_i$. Table 1 shows the number of coefficients in the Walsh, the arithmetic, the arithmetic-Haar expressions on the quaternion groups, and the Fourier expression on the quaternion groups for f. In the case of Fourier expressions, there are 5248 real and 5120 purely imaginary coefficients. The rest of 7776 coefficients are complex-valued, which means require twice more space to be stored. However, even in this case, the Fourier expression appears the most compact in the number of non-zero coefficients count.*

2.1 FFT on non-Abelian Groups

Even tough applications of FFT on non-Abelian groups have been suggested in [3], [4], and also used somewhere else, see for example, [5], [15], and references therein, the potential advantages of spectral methods on non-Abelian groups seems not yet to have been properly exploited. In [**?**], the following question is asked: "The ultimate purpose must be to find out whether this group (the quaternion Q_2) may be as significant for logic synthesis as it seems to be for filtering and other signal processing tasks".

In [13], and [14], some partial answers to this question are provided by the way of a series of experiments. It is shown, among the other things, that for functions for more than 10 variables, FFT on quaternion groups permits compromising between space and time complexities of FFT. For example, if $n = 14$, FFT on the quaternion groups is 10 times faster than on the dyadic groups at the price of three times more space.

Example 2 *Table 2 compares the time (t) and space (m) requirements for FFT on finite dyadic groups and the quaternion groups. For $n = 8, 9, 10, 14$, we use the domain groups C_2^8, C_2^9, C_2^{10}, C_2^{14}, and $C_4Q_2^2$, Q_2^3, $C_2Q_2^3$, $C_4Q_2^4$, respectively. The test functions are taken from $mcnc$ benchmarks.*

Table 2. Complexity of FFT.

f	n	t	m	t	m
adr4	8	60.90	46.39	125.80	100.27
misex1	8	33.20	35.14	73.70	89.17
rd84	8	116.10	46.39	133.90	101.39
mul4	8	63.40	45.27	129.70	99.29
9sym	9	379.80	86.39	324.60	193.30
apex4	9	464.00	86.39	336.00	196.96
clip	9	466.00	86.39	339.00	197.38
adr5	10	759.00	166.39	787.00	381.89
mul5	10	788.00	164.14	833.00	379.93
sao2	10	1531.00	166.39	763.00	375.99
adr7	14	233360.00	1926.39	78360.00	5334.39
misex3	14	801480.00	1926.39	75060.00	5331.19
mul7	14	123200.00	1618.28	65320.00	5165.25

3 Spectral Interpretation of DDs

Due to spectral interpretation of DDs, different classes of DDs can be uniformly regarded as particular examples of Spectral transform DDs (STDDs) [**?**], [16]. Spectral interpretation of DDs shows that assignment of a given function f to a DD through a decomposition by expansion rules defining the nodes in the DD is equivalent with performing a spectral transform of f. Constant nodes show the spectral coefficients, and products of labels at the edges determine the basis functions with respect to which the spectral transform is defined. The basic DDs are those defined with respect to the identity transform. They are denoted as Binary DDs (BDDs) Multi-terminal binary DDs (MTBDDs) [2], Multiple-place DDs (MDDs) and Multi-terminal DDs (MTDDs) [11], [**?**], depending on the domain set for variables and the range for function values.

Each non-terminal node in a STDD is related with two co-factors of the spectrum S_f. In the case of basic DDs, these are co-factors of f. For a node at the i-th level, the co-factors are determined by an assignment of the values for the first $(i-1)$ variables. The co-factors are represented by subtrees rooted at the nodes where point the outgoing edges of the considered node. Since, the products of labels at the edges represent basis functions, when we determine f from a STDD, we perform a transform inverse to that used in definition of the STDD, which is specified by the labels at the edges. We start from the constant nodes and follow the edges by performing a composition of subfunctions represented at the nodes multiplied with the labels at the edges. Thus, we perform a direct spectral transform to represent f by a DD, and conversely, the inverse transform to determine f from the DD. Fig. 2 illustrates the spectral interpretation of DDs. This interpretation permits the following remark, clarifying relationships between some DDs, as well as the way and the analytic expressions in form of which various DDs, as for example those considered in [1], [6], [**?**], represent discrete functions.

Remark 1 *A DD represents at the same time f and the spectrum S_f with respect to the transform used in definition of this DD. Thus, a DD is STDD(f) and the MTDD(S_f).*

Example 3 *For a given switching function f, BDD(f) is FDD(S_f), and conversely, BDD(S_f) is FDD(f), where FDD stands for Functional DDs [7] denoted also as Positive Polarity Reed-Muller DDs (PPRMDDs) [10].*

Example 4 *Edge-valued binary DDs (EVBDDs) [6], Factored EVBDDs (FEVBDDs) [**?**], Binary moment DDs (BMDs) [1], their edge-valued version $*$BMD, Arithmetic spectral transform DDs (ACDDs) [16], are graphic representations for arithmetic expressions in $C(C_2^n)$.*

Spectral interpretation corrects and generalizes a statement that in DDs the number of outgoing edges per nodes is necessarily equal to the number of different values the variables assigned to the nodes can take. It should be formulated as follows.

Remark 2 *Assume that for a given n-variable function f, the domain group G of order g is the direct product of n subgroups G_i, of orders g_i, $i = 1, \ldots, n$. Then, the number of outgoing edges of nodes at the i-th level is equal to the cardinality γ_i of the dual object of G_i.*

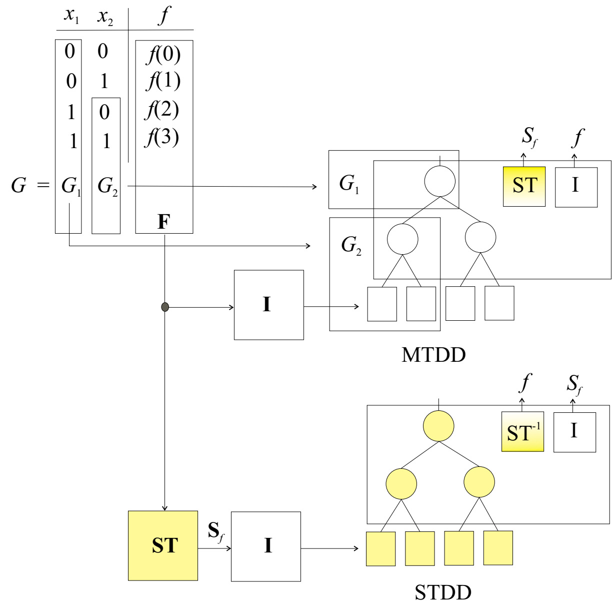

Figure 1. Spectral interpretation of DD.

It follows, that the maximum number of nodes per levels, usually denoted as the width of DD, is determined by the cardinality of the dual object Γ for G. For Abelian groups Γ expresses the structure of a group isomporphic to G. Therefore, the use of non-Abelian groups as domain groups for DDs permit a simultaneous reduction of both the width and the number of levels, denoted as the depth, of a DD. In circuit synthesis from DDs, these characteristics, the depth and the width, relate to the propagation delay and the area of circuit.

Example 5 *Table 3 and Table 5 compare the size (s), and the width (w) of Shared BDDs (SBDDs) and Fourier DDs (FNADDs) on quaternion groups for adders for multipliers. We also show the number of non-terminal (ncn), the constant nodes (cn), and the percent of used nodes in the DD from the total of nodes in the corresponding decision trees.*

Table 4 and Table 6 compare the area ($a = sw$) and percent of used nodes with respect to the toal of nodes in the decision trees of Shared BDDs (SBDDs) and Fourier DDs (FNADDs) on quaternion groups for adders for multipliers.

The discussed feature of non-Abelian groups becomes especially important when we want to reduce the

Table 3. SBDDs and FNADDs for adders

	SBDD				FNADD			
n	ntn	cn	s	w	ntn	cv	s	s
2	190	2	8	21	4	7	11	2
3	55	2	57	20	6	7	13	4
4	101	2	103	30	14	14	28	7
5	224	2	226	62	18	16	34	7
6	475	2	477	126	21	12	33	7

Table 4. Area and percent of used nodes SBDDs and FNADDs for adders

		SBDD		FNADD	
n	cubes	a	%	a	%
2	11	168	22.00	22	52.38
3	31	1140	42.86	52	31.70
4	75	3090	19.84	196	33.73
5	167	14012	10.98	238	8.31
6	355	60102	5.81	231	4.00

Table 5. SBDDs and FNADDs for multipliers

	SBDD				FNADD			
n	ntn	cn	size	width	ntn	cv	size	width
2	17	2	19	5	4	10	14	2
3	61	2	63	15	9	20	29	4
4	157	2	159	39	24	42	66	11
5	471	2	473	114	37	50	87	17
6	786	2	788	192	45	49	94	22

Table 6. Area and percent of used nodes in SBDDs and FNADDs for multipliers

		SBDD		FNADD	
n	cubes	a	%	a	%
2	7	95	14.28	28	9.52
3	32	945	11.27	116	9.75
4	128	6201	7.51	726	13.25
5	488	53912	5.54	1479	4.51
6	939	151296	2.34	2068	2.68

complexity of DD representations by using nodes with increased number of outgoing edges. In group-theoretic approach, the increased functionality of nodes is interpreted as the use of larger subgroups G_i in decomposition of the domain group G, in which case the non-Abelian groups offer advantages, since it is always $\gamma_i \leq g_i$. It should be noted, related to this property, that non-Abelian groups introduce matrix-valued constant nodes in DDs. Each of these nodes can be represented by an ordinary number valued DD by concatenating rows or columns of the matrices in constant nodes. Thus, DDs on non-Abelian groups permit two-level optimization of DDs. First, we chose a suitable decomposition of G, where some of G_i may be non-Abelian groups, and then, we chose the most compact DD representations for each matrix-valued constant node.

3.1 DD-Methods for calculation of spectral transforms

Spectral interpretation of DDs, establishes and explains relationships between FFT and DD-methods for the calculation of spectral transforms. In both, FFT and DD-methods, the calculation of a transform of a function on $G = G_1 \cdots G_n$ is performed though a series of n transforms of on G_i. In DD-methods, these transforms are performed by traversing MTBDD(f) and processing nodes and cross points level by level, by performing at the i-th level, the operations described by the transform matrix on G_i, $\mathbf{T}_i$. Componentwise operations over vectors in FFT, are replaced by the operations over subtrees representing subfunctions related to a node. Efficiency of DD-methods, and ability to process large functions originates in the reduction performed in transforming DT into DD

1. In DD-methods, the processing of constant subvectors in the vector of function values is simplified and reduced to the processing of a constant node.
2. There is no repeated processing of equal subvectors, since the isomorphic subtrees are eliminated in the reduction of the DD.

The basic operations in FFT are conveniently described by some basic matrices used in definition of a transform. In Kronecker product representable transforms, these are factors in the Kronecker product for the transform matrix $\mathbf{T}$. Table 7 shows the basic matrices for the most often used transforms in $C(C_2^n)$. The matrices $\mathbf{I}(1)$ and $\mathbf{W}(1)$ are used in the matrix definition of the Haar transform, although it is not Kronecker product representable [3]. To calculate the spectrum of a function f given by the MTBDD(f), we perform at each node and

Table 7. Basic matrices.

Identity	Reed-Muller
$\mathbf{I}(1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\mathbf{R}(1) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$
Arithmetic	**Walsh**
$\mathbf{A}(1) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$	$\mathbf{W}(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

Table 8. Calculation times.

f	In	Out	RM	Walsh	AR	Haar
alu4	14	8	400	1590	640	2.26
apex4	9	19	270	370	160	1.69
misex3	14	14	310	1610	380	2.12
5xp1	7	10	50	20	10	0.11
sao2	10	4	50	30	40	0.20

the cross point the operations described by $\mathbf{T}_i$. Due to the properties of the Haar matrix, a further simplification is possible in calculation of the Haar spectrum. It is enough to perform the calculations described by $\mathbf{W}(1)$ over the first two values of the subfunctions related to the node. Conversely, the inverse transform is calculated by performing operations described by the inverses of the basic transform matrices. Again, the savings are possible in the case of the Haar transform. It is enough to process the leftmost node at the each level in the DT. Generalizations of this algorithm to functions in Fibonacci topologies and related circuit synthesis from Fibonacci DDs are given in [**?**].

Example 6 *Table* **??** *shows the CPU times for calculation of different spectral transforms by using DD-methods for some benchmark functions. The important feature is that the algorithms are executable on a simple hardware. In this experiment, the calculations were performed on a 133MHz Pentium PC with 32MBytes of RAM. Savings in the case of the Haar transform, achieved by the exploitation of properties of the Haar matrix, are obvious.*

This interpretation shows that in DD-methods, we perform FFT over another data structure, instead over the vectors, since the basic matrices determine the basic operations in the corresponding FFT methods [12].

The same interpretation permits to define different sets of basic functions used in spectral techniques, by assigning the basic transforms to the nodes of MTDDs, and determining the labels at the edges from columns of the inverse matrices. This is the same way in which we determine the expansion rules in different DDs. In particular, Fourier DDs are determined by using the expansion rules defined by the Fourier transform, which permits combination of Abelian and non-Abelian groups as subgropus in decomposition of the domain group G into a direct product of subgroups G_i.

4 Closing Remarks

The complexity of problems in switching and MV theory and logic design give rise for further application of spectral techniques, since they may provide for simple and elegant analytic solutions where the traditional approaches reduce to the brute force search methods.

Acknowledgements

Special thanks are due to Prof. Jaakko T. Astola of Tampere Int. Center for Signal Processing (TICSP), Tampere, Finland for providing an opportunity for the authors to meet at TICSP and for comments and suggestions in several discussions which were very enlightening and helpful for the work leading to this report.

References

[1] Bryant, R.E., Chen, Y.-A., “Verification of arithmetic functions with binary moment decision diagrams”, Research Report CMU-CS-94-160, May 31, 1994.

[2] Clarke, E, M., M.C., Millan, K.L., Zhao, X., Fujita, M., “Spectral transforms for extremely large Boolean functions”, in Kebschull, U., Schubert, E., Rosenstiel, W., Eds., *Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expression in circuit Design*, Hamburg, Germany, September 16-17, 1993, 86-90. Workshop Reed-Muller’93, 86-90.

[3] Karpovsky, M.G., *Finite Orthogonal Series in the Design of Digital Devices*, John Wiley, 1976.

[4] Karpovsky, M.G., ” Fast Fourier transform over a finite non-Abelian group”, *IEEE Trans. on Computers*, Vol. C-26, 1977, 1028-1031.

[5] Karpovsky, M.G., Trachtenberg, E.A., ”Fourier transforms over finite groups for error detection and error correction in computation channels”, *Inf. and Control*, 1979, 40, 335-358.

[6] Lai, Y.-T., Pedram, M., Vrudhula, S.B.K., “EVBDD-based algorithms for integer linear programming, spectral transformation, and functional decomposition”, *IEEE Trans. on CAD*, Vol. 13, No. 8, 1994, 959-975.

[7] Kebschull, U., Schubert, E., Rosenstiel, W., "Multilevel logic synthesis based on functional decision diagrams", *EDAC 92*, 1992, 43-47.

[8] Moraga, C., *Advances in Spectral Techniques*, Berichte zur angewandten Informatik, Dortmund, 1998.2, ISSN 0946-2341.

[9] Moraga, C., Heider, R., "Tutorial review on applications of the Walsh transform in switching theory", *Proc. First Int. Workshop on Transforms and Filter Banks*, TICSP Series # 1, June 1998, 494-512.

[10] Sasao, T., "Representations of logic functions by using EXOR operators", in: [11], 29-54.

[11] Sasao, T., Fujita, M., (eds.), *Representations of Discrete Functions*, Kluwer Academic Publishers, 1996.

[12] Stanković, R.S., Falkowski, B.J., "FFT and decision diagrams based methods for calculation of spectral transforms", *Proc. IEEE Int. Conf. on Informatics, Communications and Signal Processing*, Singapore 1997, Vol. 1, 241-245.

[13] Stanković, R.S., Milenović, D., "Some remarks on calculation complexity of Fourier transforms on finite groups", *Proc. 14th European Meeting on Cybernetics and Systems Research, CSMR'98*, April 15-17, 1998, Vienna, Austria, 59-64.

[14] Stanković, R.S., Milenović, D., "Some remarks on Fourier transforms on finite non-Abelian groups", *Proc. First Int. Conf. on Information, Communications and Signal Processing*, Singapore, September 9-12, 1997.

[15] Stanković, R.S., Moraga, C., Astola, J.T., *Readings in Fourier Analysis on Finite Non-Abelian Groups*, TICSP Series #5, TICSP, Finland, 1999.

[16] Stanković, R.S., Sasao, T., Moraga, C., "Spectral transform decision diagrams" in: [11], 55-92.