FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 20, no. 3, December 2007, 295-308

Remarks on Applications of Arithmetic Expressions for Efficient Implementation of Elementary Functions

Radomir S. Stanković and Jaakko T. Astola

Abstract: It has been recently shown in [1], that elementary mathematical functions (as trigonometric, logarithmic, square root, gaussian, sigmoid, etc.) are compactly represented by the Arithmetic transform expressions and related Binary Moment Diagrams (BMDs). The complexity of the representations is estimated through the number of non-zero coefficients in arithmetic expressions and the number of nodes in BMDs.

In this paper, we show that further optimization can be achieved when the method in [1] is combined with Fixed-polarity Arithmetic expressions (FPRAs). In addition, besides complexity measures used in [1], we also compared the number of bits and 1-bits required to represent arithmetic transform coefficients in zero polarity and optimal polarity arithmetic expressions. This is a complexity measure relevant for the alternative implementations of elementary functions suggested in [1]. Experimental results confirm that exploiting of FPARs may provide for considerable reduction in terms of the complexity measures considered.

Keywords: Elementary functions, Arithmetic expressions, Fixed-polarity Arithmetic expressions, decision diagrams,

1 Introduction

Implementation of elementary mathematical functions, as trigonometric, logarithmic, square root, sigmoid, gaussian, etc. in hardware, is an important task in many applications. It is usually accepted that for efficient realizations, compact representations of such functions are required whatever the criteria of efficiency of the

Manuscript received August 18, 2007.

R. Stanković is with Dept. of Computer Science, Faculty of Electronics, 18 000 Niš, Serbia (e-mail: rstankovic@bankerinter.net). J. Astola is with Institute of Signal Processing, Tampere University of Technology, FIN-33101, Tampere, Finland, (e-mail: Jaakko.Astola@tut.fi).

realizations. Depending on these criteria, various definitions of compactness of the representations are assumed.

The representation of elementary functions was recently considered in [1], where it has been shown that the arithmetic transform represents many elementary functions compactly. The same applies to the sizes of Binary Moment Diagrams (BMDs), since they represent functions in terms of arithmetic expressions, when compared to Multi-Terminal Binary Decision Diagrams (MTBDDs), representing functions in terms of generalized Sum-of-Product forms [2].

It is conjectured in [1] that other decision diagrams which represent functions in terms of arithmetic expressions, (see, for example, [2, 3] and references therein) can also be used to compactly represent certain important classes of elementary functions. In particular, links between the arithmetic coefficients and weights at the edges in Edge-valued Binary Decision Diagrams (EVBDDs) [4], shown in [5], motivated exploiting of EVBDDs for design of numeric function generators [6].

This paper is a continuation of research in this area in the following directions.

The interesting results presented in [1] exploit a basic feature of spectral methods that a proper selection of a spectral transform may provide compact representation of a class of functions. We point out that besides this, further manipulation with the basis functions used in the selected spectral transform, may provide additional improvement in the compactness of the representations. In particular, shift and reordering of basis functions in the arithmetic transform, which can be alternatively interpreted as Fixed-polarity arithmetic expressions (FPARs) [7], can be used to further reduce complexity of representations of elementary functions.

By following the method used in [1], the complexity of representation of elementary functions is first estimated in terms of the number of coefficients in zero-polarity and optimal polarity arithmetic expressions. We also consider the number of distinct function values and the coefficients.

However, it has been stated in [1]

Since BMDs represent elementary functions compactly, BMDs are promising for verification of hardware for elementary functions, and for the alternative implementation of embedded RAM on FPGA for the function tables.

For such applications, the number of bits and 1-bits required to represent function values or arithmetic coefficients is an appropriate complexity measure. Therefore, we also compared the number of bits and 1-bits required to represent function values and arithmetic coefficients.

Finally, we considered related decision diagrams. Instead of BMDs that are discussed in [1], we compared the sizes of MTBDDs and Arithmetic Spectral Transform Decision Diagrams (ACDDs) [8] for the following reasons.

BMDs take advantages of the reduction rules (generalized ZBDD reduction rules) adopted to the underlying arithmetic transform that is actually used in the definition of them. This advantage is a feature of BMDs derived indirectly from the usage of the arithmetic transform viewed as the integer equivalent of the Reed-Muller transform that is used in formulation of the ZBDD reduction rules. The term generalized here is intended to point out that we are working with integers and real numbers instead of logic values 0 and 1.

ACDDs do not exploit these reduction rules, but the generalized BDD reduction rules, the same rules as in MTBDDs. The reduction rules are applied to elements of the function vectors in the case of MTBDDs and vectors of zero-polarity and optimal polarity arithmetic spectra for ACDDs. Since after a change of labels at the edges, ACDDs converts into MTBDDs of the arithmetic spectra, ACDDs illustrate the effect of the application of the arithmetic spectra rather directly.

However, the price is that ACDDs do not always have a smaller size than MTB-DDs for elementary functions. This follows from the property that FPARs reduce the number of non-zero coefficients, which are related to the number of paths in decision diagrams. It is known that reduction of the number of paths does not necessarily imply the reduction of the number of nodes in a decision diagram [9].

2 Arithmetic Representations of Elementary Functions

By following classical principles, explained for example in detail in [1], we can convert n-bit precision real valued functions into n-input m-output switching functions.

The multiple-output functions can be converted into integer functions by considering m-bit binary vectors as integers.

In this paper, we discuss representations of elementary functions, converted into integer functions by using n-bit fixed-point representation for function values, by arithmetic transform coefficients. We extend the considerations in [1] by exploiting Fixed-polarity arithmetic (FPAR) expressions. That permits more compact representations than arithmetic expressions restricted to zero-polarity and related decision diagrams [1]. We also use the number of bits and 1-bits to represent arithmetic coefficients as another complexity measure well suited for different realizations of elementary functions.

2.1 Arithmetic transform

For an n-variable function defined by the function vector $\mathbf{F} = [f(0), \ldots, f(2^n - 1)]^T$, the arithmetic spectrum $S_f = [S_f(0), \ldots, S_f(2^n - 1)]^T$ is defined as

$$\mathbf{S}_f = \mathbf{A}(n)\mathbf{F},$$

where the $(2^n \times 2^n)$ transform matrix $\mathbf{A}(n)$ is defined as

$$\mathbf{A}(n) = \bigotimes_{i=1}^{n} \mathbf{A}(1), \quad \mathbf{A}(1) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \tag{1}$$

and $\otimes$ denotes the Kronecker product.

2.2 Arithmetic expressions

The arithmetic expression is the functional expression in terms of elementary products of binary-valued variables, whose coefficients are the arithmetic spectral coefficients. In matrix notation, the arithmetic expression for an n-variable function is defined as

$$f(x_1,\ldots,x_n) = \left(\bigotimes_{i=1}^{n} \begin{bmatrix} 1 & x_i \end{bmatrix} \right) \left(\bigotimes_{i=1}^{n} \mathbf{A}(1) \right) \mathbf{F}. \tag{2}$$

Example 1 *Table 2 shows a two-oputput function $f = (f_0, f_1)$ of $n = 3$ variables. The function vector $\mathbf{F}$ with output considered as binary representations of integers, and the arithmetic spectra for zero-polarity $\mathbf{A}_0$ and optimal polarity $\mathbf{A}_{opt}$ arithmetic expressions are*

$$\begin{aligned} \boldsymbol{F} &= [0,1,2,2,2,1,2,1]^T, \\ \boldsymbol{A}_0 &= [0,1,2,-2,2,-2,-2,2]^T, \\ \boldsymbol{A}_{opt} &= [1,1,0,0,0,0,0,-2]^T. \end{aligned}$$

The optimal polarity is $H = (1,1,1)$. The number of non-zero function values and arithmetic coefficients is 7, 7, and 3. The number of distinct non-zero function values and arithmetic coefficients are 2, 3, and 2, respectively. The number of 1-bits/bits to represent these coefficients are $7/11$, $7/13$, and $3/5$. The number of bits is determined by counting bits starting from the most significant non-zero bit, and 1-bits are simply encountered. The sign bit is not taken into considerations.

There are different approaches to determine FPARs efficiently in terms of space and time, see for example [10] and references therein. These algorithms are analogues to the corresponding algorithms for determination of Fixed-polarity Reed-Muller expressions (FPRMs) [11], and in the case of elementary functions with the precision of 16 bits, exact algorithms for the determination of optimal FPARs can be easily applied even on simple hardware. Determination of FPARs by using disjoint cubes to specify functions processed has been considered in [12].

Table 1. Function f in Example 2.

$x_1x_2x_3$	f_0, f_1	f
000	00	0
001	01	1
010	10	2
011	10	2
100	10	2
101	01	1
110	10	2
111	01	1

The applications of arithmetic expressions in circuit description and design date back to fifties [13–15], with a continuous interest in the subject, see for instance, [16–19], up to the recent applications discussed in a number of papers devoted exclusively to this subject the special issue of the journal *Avtomatika i Telemekhanika*, No. 6, 2004, see also related discussions in [20].

2.3 Fixed-polarity arithmetic expressions

The optimization of arithmetic expressions in the number of non-zero coefficients count can be performed by selecting different polarities for variables x_i, i.e., the usage of positive and negative literals, but not both for the same variable. In this way, Fixed-polarity arithmetic expressions (FPARs) are defined, see for example, [7].

In matrix notation, FPARs are defined as

$$f(x_1,\ldots,x_n) = \left(\bigotimes_{i=1}^{n} \begin{bmatrix} 1 & x_i^{h_i} \end{bmatrix}\right)\left(\bigotimes_{i=1}^{n} \mathbf{A}^{h_i}\right)\mathbf{F}, \tag{3}$$

where

$$x_i^{h_i} = \begin{cases} x_i, & h_i = 0, \\ \overline{x}_i, & h_i = 1, \end{cases}$$

and

$$\boldsymbol{A}^{h_i}(1) = \begin{cases} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, & h_i = 0 \\ \\ \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}, & h_i = 1, \end{cases}$$

The polarity vector $H = (h_1,\ldots,h_n)$ uniquely specifies each FPRA.

Example 2 *Table 2 shows a two-oputput function* $f = (f_0, f_1)$ *of* $n = 3$ *variables. The function vector* **F** *with output considered as binary representations of integers, and the arithmetic spectra for zero-polarity* $\mathbf{A}_0$ *and optimal polarity* $\mathbf{A}_{opt}$ *arithmetic expressions are*

$$\begin{aligned} \boldsymbol{F} &= [0,1,2,2,2,1,2,1]^T, \\ \boldsymbol{A}_0 &= [0,1,2,-2,2,-2,-2,2]^T, \\ \boldsymbol{A}_{opt} &= [1,1,0,0,0,0,0,-2]^T. \end{aligned}$$

The optimal polarity is $H = (1,1,1)$. *The number of non-zero function values and arithmetic coefficients is* 7, 7, *and* 3. *The number of distinct non-zero function values and arithmetic coefficients are* 2, 3, *and* 2, *respectively. The number of* 1-*bits/bits to represent these coefficients are* $7/11$, $7/13$, *and* $3/5$. *The number of bits is determined by counting bits starting from the most significant non-zero bit, and* 1-*bits are simply encountered. The sign bit is not taken into considerations.*

Table 2. Function f in Example 2.

$x_1x_2x_3$	f_0, f_1	f
000	00	0
001	01	1
010	10	2
011	10	2
100	10	2
101	01	1
110	10	2
111	01	1

There are different approaches to determine FPARs efficiently in terms of space and time, see for example [10] and references therein. These algorithms are analogues to the corresponding algorithms for determination of Fixed-polarity Reed-Muller expressions (FPRMs) [11], and in the case of elementary functions with the precision of 16 bits, exact algorithms for the determination of optimal FPARs can be easily applied even on simple hardware. Determination of FPARs by using disjoint cubes to specify functions processed has been considered in [12].

2.4 Arithmetic transform decision diagrams

Multi Terminal Binary Decision Diagrams (MTBDDs) are a generalization of Binary Decision Diagrams (BDDs) derived by allowing integers as the values of constant nodes, see for instance [21]. They can represent multi-output switching functions when m-bit output binary vectors are considered as binary representations of integers.

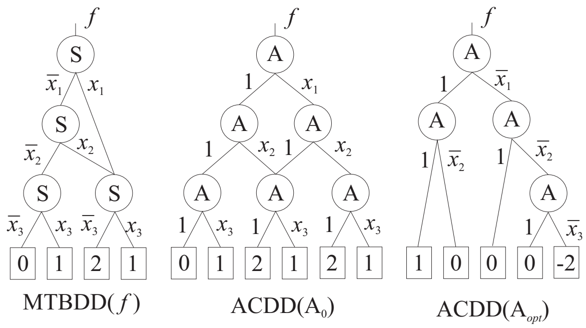

Fig. 1. MTBDD, ACDDs for the zero-polarity and optimal polarity expressions.

Arithmetic Transform Decision Diagrams (ACDDs) are another generalization where instead of functions values the constant nodes represent arithmetic coefficients [8]. Correspondingly, labels at the edges are modified to correspond to the arithmetic analogues of positive Davio expansion rule $f = 1 \cdot f_0 + x_i(-f_0 + f_1)$, where f_0 and f_1 are co-factors of f for $x_i = 0$ and 1, respectively. Thus, ACDDs are graphical representations of positive polarity arithmetic expressions [8]. The integer counterpart of the negative Davio expansion rule is defined as $f = 1 \cdot f_1 + (1 - x)(f_0 - f_1)$. When constant nodes in ACDDs represent coefficients in FPARs, the labels at the edges are the integer counterparts of either positive or negative Davio expansion rules. As noticed above, the difference between BMDs and ACDDs is in the reduction rules [8].

Example 3 *Fig. 1 shows the MTBDD, ACDD for the zero-polarity and optimal polarity arithmetic expressions for the function f in the Example 2. The numbers of non-zero constant nodes in these diagrams are* 4, 6, *and* 4*. The numbers of pats from the root node to the non-zero constant nodes are* 4, 7, *and* 2.

3 Complexity Measures and Experimental Results

When considering representations by functional (spectral or other) expressions, the usually accepted complexity measure is the number of non-zero coefficients and the number of distinct coefficients (which is equal to the number of constant nodes in decision diagrams).

In this respect, FPARs which have a reduced number of non-zero coefficients may be useful to further reduce complexity of the arithmetic expressions (zero-polarity expressions), which proved useful in representation of elementary mathematical functions [1].

However, in circuit realizations, it is important to consider also the number of bits and 1-bits to represent function values and spectral coefficients. It may happen that for some functions, smaller number of non-zero coefficients, but of larger values, may require greater number of bits, thus, it has larger complexity than other functional expressions although possibly with a larger number of coefficients.

In the representation of functions by decision diagrams, relevant complexity measures are parameters of the diagrams, as the number of non-terminal nodes, constant nodes, number of paths, etc.

In this section, we discuss some experimental results in estimation of complexity of representations of elementary functions converted into 8-bit input 8-bit output switching functions by using the method presented in [1]. In these tables $poly-c1$ and $poly-c2$ are examples of two randomly generated polynomials of order four and the functions Entropy, Sigmoid and Gaussian are specified below.

Entropy	$-x\log_2 x-(1-x)\log_2(1-x)$,
Sigmoid	$\frac{1}{1+e^{-4x}}$,
Gaussian	$\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$.

3.1 Number of arithmetic coefficients

Table 3 compares the number of non-zero coefficients in zero-polarity and optimal polarity arithmetic expressions for elementary functions. We can see that FPARs always produce smaller number of coefficients, compared to the number of function values and coefficients in zero-polarity arithmetic expressions. There are few expressions with the same minimum number of non-zero coefficients. In this table, we show the last determined minimum polarity H in an exhaustive search for optimal polarity.

Table 4 shows the number of distinct values in the function vectors and vectors of zero-polarity and optimal polarity arithmetic expressions. There are examples were the number of distinct coefficients is smaller for the zero-polarity than for the optimal polarity FPAR, but certainly not the total number of coefficients.

3.2 Number of bits and 1-bits

The number of bits to represent a coefficient is determined as the number of bits counting from the most significant non-zero bit to the last significant bit. The number of 1-bits is the number of non-zero bits out of the total of bits.

Table 5 shows the number of bits and 1-bits required to represent coefficients in zero-polarity and optimal polarity arithmetic expressions for elementary functions. The shown values are for the same polarities as in Table 3.

We see, that optimal polarity FPARs require quite fewer bits and also 1-bits to represent the coefficients.

Table 3. Number of non-zero values and coefficients

Function	function values	Arithmetic expressions		
		zero	optimal	H
$1/\sqrt{(x)}$	254	249	104	11111111
$1/x$	256	80	43	11011100
$2^x - 1$	254	210	142	01100010
$\cos(\pi x)$	256	163	153	11100011
entropy	256	255	192	11000000
gaussian	256	211	184	10100000
$\ln(x)$	255	207	129	11111100
$poly - c1$	255	222	217	00001000
$poly - c2$	255	252	232	00101110
x^2	240	135	102	00000111
x^3	215	174	138	00000011
x^4	193	135	122	00000111
sigmoid	256	187	169	01101110
$\sin(\pi x)$	256	143	141	00000100
$\sqrt{-\ln(x)}$	256	256	155	01101101
$\sqrt{x}$	256	252	145	11110111
$\tan(\pi x)$	256	236	195	10010000
$\tan^2(\pi x) + 1$	216	107	97	10001000
av.	246	193	147	

3.3 Number of nodes

Coefficients in a functional expression correspond to paths in the related decision diagrams viewed as graphic representations of these expressions. In general, the reduced number of paths do not necessarily imply the reduced number of nodes [9]. However, the experiments performed show that in ACDDs for elementary functions defined with respect to optimal polarity expressions, the number of nodes is also always reduced compared to MTBDDs except for the square root. In six examples, ACDDs corresponding to the zero-polarity arithmetic expressions have fewest nodes.

Table 7 shows in detail parameters of MTBDDs and ACDDs for the zero-polarity and optimal polarity expressions. There, the number of non-terminal nodes (ntn), constant nodes (cn), number of paths from the root node to the zero valued constant node (0-paths), and non-zero paths (c-paths), as well as the width of the diagram defined as the maximum number of nodes per level (w) are shown.

FPARs reduce the number of c-paths and therefore related decision diagrams can be even more efficient in applications requiring traversing diagrams than di-

Table 4. Number of distinct function values and coefficients

Function	function value	Arithmetic expressions zero	optimal
$1/\sqrt(x)$	59	108	19
$1/x$	41	102	30
$2^x - 1$	212	30	21
$\cos(\pi x)$	230	33	35
entropy	106	57	42
gaussian	121	40	40
$\ln(x)$	256	29	21
$poly - c1$	155	98	100
$poly - c2$	161	128	126
x^2	192	11	11
x^3	157	21	19
x^4	135	23	20
sigmoid	180	25	32
$\sin(\pi x)$	115	33	32
$\sqrt{-\ln(x)}$	169	90	36
$\sqrt{x}$	175	198	198
$\tan(\pi x)$	255	58	41
$\tan^2(\pi x) + 1$	76	88	57
av.	155	65	48

Table 5. Number of bits and 1-bits

Function	function value	Arithmetic expressions zero	optimal
$1/\sqrt(x)$	945/2009	717/1237	125/189
$2^x - 1$	938/1658	305/420	173/252
$1/x$	451/753	942/1690	131/188
$\cos(\pi x)$	1180/1880	222/337	215/324
entropy	1234/1956	468/769	305/457
gaussian	1152/1898	315/457	267/417
$\ln(x)$	933/1696	275/399	151/227
$poly - c1$	1001/1741	644/1319	630/1230
$poly - c2$	1006/1765	1242/1898	732/1535
x^2	834/1466	50/102	50/102
x^3	692/1215	222/319	171/231
x^4	596/1029	161/240	153/ 213
sigmoid	1024/1702	284/350	229/336
$\sin(\pi x)$	1180/1880	201/290	198/286
$\sqrt{-\ln(x)}$	956/1730	517/876	221/364
$\sqrt{x}$	1148/1963	517/876	178/245
$\tan(\pi x)$	896/1618	370/559	264/391
$\tan^2(\pi x) + 1$	916/1584	490/825	355/550
av.	949/1641	441/720	252/418

Table 6. Number of nodes in MTBDD and ACDDs

Function	MTBDD	ACDD	
		zero	optimal
$1/\sqrt{(x)}$	165	339	143
$1/x$	111	312	134
2^x-1	441	203	180
$\cos(\pi x)$	456	206	206
entropy	339	270	237
gaussian	368	236	237
$\ln(x)$	511	214	164
$poly-c1$	410	340	343
$poly-c2$	416	378	378
x^2	393	92	92
x^3	337	162	156
x^4	287	144	151
sigmoid	399	206	215
$\sin(\pi x)$	350	199	196
$\sqrt{-\ln(x)}$	379	320	214
$\sqrt{x}$	411	284	178
$\tan(\pi x)$	510	278	233
$\tan^2(\pi x)+1$	261	333	268
av.	363	251	206

agrams based on zero-polarity arithmetic expressions. Application of ZBDD reduction rules to various diagrams that are graphical representations of FPARs will further reduce the number of nodes. For instance, it is clear that BMDs based on FPARs might further reduce complexity of the representations compared to BMDs using zero-polarity expressions.

4 Closing Remarks

In spectral techniques, given functions are represented as linear combinations of some predefined basis functions and there are three natural levels for the optimization of these representations. When ordered hierarchically by the level of freedom, these optimization possibilities are

1. Selection of the underlying algebraic structure, and then choosing a suitable basis within the structure selected.
2. When the algebraic structure is fixed, we still can select among many different sets of basis functions.
3. In the basis, i.e., in a selected transform, we can perform the optimization by some reordering of basis functions, or other similar manipulations with basis functions, like shift or permutation of their elements.

Table 7. Number of nodes in MTBDD and ACDDs for zero-polarity and optimal polarity expressions.

Function	MTBDD					ACDD-zero					ACDD-optimal				
	ntn	cn	0	c	w	ntn	cn	0	c	w	ntn	cn	0	c	w
$1/\sqrt(x)$	106	59	1	106	3	231	108	6	248	105	124	19	77	99	35
$1/x$	70	41	0	71	20	210	102	0	256	89	104	30	59	77	30
2^x-1	229	212	1	229	102	173	30	35	210	58	159	21	77	137	55
$\cos(\pi x)$	235	230	0	236	110	173	33	71	148	58	171	35	77	150	56
entropy	233	106	0	234	112	213	57	1	255	8	195	42	46	188	73
gaussian	247	121	0	248	120	196	40	42	206	73	197	40	61	177	71
$\ln(x)$	255	256	1	255	128	185	29	41	198	63	143	21	85	127	46
$poly-c1$	255	155	1	255	128	242	98	32	222	115	243	100	36	216	116
$poly-c2$	255	161	2	254	128	250	128	4	251	123	252	126	24	231	125
x^2	201	192	1	201	86	81	11	73	49	23	81	11	73	49	23
x^3	180	157	3	178	75	141	21	36	172	46	137	19	67	136	47
x^4	152	135	1	152	63	124	20	43	133	38	128	23	62	118	40
sigmoid	219	180	0	220	94	181	25	61	182	62	183	32	69	167	62
$\sin(\pi x)$	235	115	0	236	110	166	33	89	141	56	164	32	88	140	55
$\sqrt{-\ln(x)}$	210	169	0	211	83	230	90	0	256	103	178	36	68	154	61
$\sqrt{x}$	219	192	0	220	92	223	61	4	252	96	157	21	81	138	54
$\tan(\pi x)$	255	255	2	254	128	220	58	37	211	93	192	4	71	167	66
$\tan^2(\pi x)+1$	255	255	2	254	128	220	58	37	211	93	192	41	71	167	66
av.	211	166	114	211	95	192	55	34	200	72	166	36	66	146	60

The second option has been exploited in [1], where it is shown that for representation of elementary mathematical functions, selection of the arithmetic transform provides compact representations. As measures of the complexity, there have been used the number of nodes in MTBDDs and BMDs, number of non-zero arithmetic coefficients, and number of distinct function values and arithmetic coefficients.

In this paper, we examine the third option by suggesting the usage of Fixed-polarity arithmetic expressions (FPARs) as a modification the basis functions in the arithmetic transform by peculiar shifting and reordering of them. In addition, besides the complexity measures mentioned, we also considered the number of bits and 1-bits required to represent zero-polarity and optimal polarity arithmetic coefficients. This complexity measure is especially important in circuit realizations, since relates to the number and complexity of interconnections.

Experimental results shows that FPARs and related decision diagrams can provide for further reductions of the representations compared to these already provided by the arithmetic transform (zero-polarity arithmetic expressions) of elementary mathematical functions.

Acknowledgments

The authors are grateful to the Reviewers whose comments improved the presentation in the paper.

This work was supported by the Academy of Finland, Finnish Center of Excellence Programme, Grant No. 213462.

References

[1] T. Sasao and S. Nagayama, "Representations of elementary functions using binary moment diagrams," in *Proc. Int. Symp. on Multiple-Valued Logic*, Singapore, May 17–20, 2006.

[2] R. Stanković and J. Astola, *Spectral Interpretation of Decision Diagrams*. Berlin: Springer, 2003.

[3] C. Moraga, T. Sasao, and R. Stanković, "A unified appraoch to edge-valued and artimetic transform decision daigrams," *Avtomatika i Telemekhanika*, no. 1, pp. 140–153, 2002, english version in *Avtomation and Remote Control*, Vol. 63, No. 1, 2002, 125-138.

[4] Y. Lai, M. Pedram, and S. Vrudhula, "Evbdd-based algorithms for integer linear programming, spectral transformation, and functional decomposition," *IEEE Trans. CAD*, vol. 13, no. 8, pp. 959–975, 1994.

[5] R. Stanković, "Some remarks about spectral transform interpretation of MTBDDs and EVBDDs," in *ASP-DAC'95*, Makuhari Messe, Chiba, Japan, Aug. 29–Sept.1, 1995, pp. 385–390.

[6] S. Nagayama, T. Sasao, and J. Butler, "Numerical function generators using edge-valued binary decision diagrams," in *ASPDAC-2007*, Yokohama, Japan, Jan. 25, 2007, pp. 535–540.

[7] V. Malyugin, *Paralleled Calculations by Means of Arithmetic Polynomials*. Moscow: Physical and Mathematical Publishing Company, Russian Academy of Sciences, 1997, (in Russian).

[8] R. Stanković, T. Sasao, and C. Moraga, "Spectral transform decision diagrams," in *Representations of Discrete Functions*, T. Sasao and M. Fujita, Eds. Kluwer Academic Publishers, 1996, pp. 55–92.

[9] E. Dubrova and D. Miller, "On disjunctive covers and robdd size," in *Proc. Pacific Rim Conference on Communication, Computers and Signal Processing*, Victoria, B.C., Canada, Aug. 23–25, 1999, pp. 162–164.

[10] R. Stanković, "Arithmetic transform ternary decision diagrams for exact minimization of fixed polarity arithmetic expressions," *Multiple-Valued Logic and Soft Computing*, vol. 10, no. 3, pp. 287–307, 2004.

[11] T. Sasao and D. Debnath, "An exact minimization algorithm for generalized reed-muller expansions," in *Proc. IEEE Asia-Pacific Conference on Circuits and Systems*, Taipei, Taiwan, Dec. 5–8, 1994, pp. 460–465.

[12] D. Janković, R. Stankovi´ c, and C. Moraga, "Arithmetic expressions optimization using dual polarity property," *Serbian Journal of Electrical Engineering*, vol. 1, no. 1, pp. 71–80, 2003.

[13] The Annals of the Computation Laboratory of Harvard University, Volume XXVII, *Synthesis of Electronic Computing and Control Circuits*, Cambridge, Massachusetts, USA, 1951.

[14] Y. Komamiya, “Theory of relay networks for the transformation be- tween the decimal and binary system,” *Bull. of E.T.L.*, vol. 15, no. 8, pp. 188–197, Aug. 1951.

[15] Translation from English under supervision by B.I. Shestakov, *Synthesis of Electronic Calculation and Control Networks*. Moscow: Izv. Inostranoi Literaturi, 1954.

[16] G. Kukharev, V. Shmerko, and S. Yanushkevich, *Technique of Binary Data Parallel Processing for VLSI*. Minsk, Belarus: Vysheyshaja shcola, 1991.

[17] V. D. Malyugin, “Representation of boolean functions by arithmetical polynomials,” *Avtomatika i Telemekhanika*, vol. 43, Part 1, no. 4, pp. 484–93, 1982, english trnaslation *Automation and Remote Control*, Vol.43, No. 4, Part 1, 1982, 496-504.

[18] ——, “Realization of boolean function corteges by means of linear arithmetical polynomials,” *Avtomatika i Telemekhanika*, no. 2, pp. 114–122, 1984, english translation *Automation and Remote Control*, Vol. 45, No. 2, Part 1, 1984, 239-245.

[19] J. Muzio, “Stuck fault sensitivity of reed-muller and arithmetic coefficients,” in *Theory and Applications of Spectral Techniques*, 2nd ed., C. Moraga, Ed., Dortmund, Germany, 1989, pp. 36–45.

[20] S. Yanushkevich, D. Miller, V. Shmerko, and R. Stanković, *Decision Diagram Techniques for Micro- and Nanoelectronic Design Handbook*. CRC Press, Taylor & Francis, 2006.

[21] T. Sasao and M. Fujita, *Representations of Discrete Functions*. NY: Kluwer Academic Publishers, 1996.