

Proceedings Int. Conf. on Boolean Problems– Freiberg,
Germany, 1998, pp. 101-108

Development of Zakrevskij's Minimization Strategy towards Arithmetical Polynomial Domain

G. Dueck¹, G. Holowinski², K. Malecki², V. Shmerko³, S. Yanushkevich⁴

¹ St. Francis Xavier Univ., Dept. of Math. & Comp. Sci., Antigonish, B2G 2W5, CANADA, e-mail: gdueck@stfx.ca ² Inst. of Comp. Sci. & Inf. Systems, Technical Univ., Szczecin, POLAND, ^{3,4} Inst. of Comp. Sci. & Inf. Systems, Technical Univ., Zolnierska Str. 49, 71210 Szczecin, POLAND, Fax: (+4891) 48 76 439, e-mail: shmerko@ii.tuniv.szczecin.pl / State Univ. of Informatics & Radioelectronics, Minsk, BELARUS

ABSTRACT

In this paper we study the possibility of using the called staircase strategy, originally presented by Zakrevskij, for minimization Boolean function in Arithmetical Polynomials forms. The results on developing staircase minimization strategy to find a quasi-optimal Arithmetical Polynomial forms for both completely and incompletely specified Boolean functions is presented. The experimental results for benchmarks and the comparison with well-known minimization strategies are discussed.

1. INTRODUCTION

1.1. Arithmetical Logic is an *inter-disciplinary* field which employs the methods of logic, polynomial algebra, matrix algebra and digital signal processing. The potential of Arithmetical Logic lies, in our view, in the efficient implementation of arithmetical circuits.

The arithmetical operations over Boolean variables are applied in pseudo-Boolean and integer programming. Arithmetical Polynomials (APs) also find applications in various problems of CAD [Falk97], [Davi78]. Decision Diagrams (DDs) have proved to be a very convenient data structure for many discrete functions. DDs permit manipulations and calculations of large discrete functions efficiently in terms of space and time [DeMi94], [Clar96], [Vrud96], [Shin97], [Falk97]. For example, it was shown in [Stan96] that Arithmetic Transform Decision Diagrams require $O(n)$ nodes while another types of DD require $O(2^n)$ nodes. The main advantage of Arithmetical Logic is that it allows us to represent and manipulate Boolean and Multiple Valued Logic (MVL) functions by DDs at the *word level* or in the *integer domain* which is often more convenient and efficient [Vrud96], [Stan96]. Authors of paper [Stan96] showed (earlier this result was presented by Malyugin [Malu82,84]) that systems of Boolean

²This work was supported in part by State Committee for Scientific Researches, POLAND

³Support in part as *James Chair Professor* from the St. Francis Xavier Univ., Dept. of Math. Comp. & Inf. Systems, CANADA, and *Found of Fundamental Researches* of National Academy of Sciences, BELARUS

⁴ Support in part from the St. Francis Xavier Univ., CANADA, and State Committee for Scientific Researches, POLAND

functions can be efficiently represent as AP (for example, a 2 bit multiplier is described as the system of 4 Boolean functions the AP is $4x_1y_1+2(x_0y_1+x_1y_0)+x_0y_0$). Probabilistic verification of a pair of Boolean functions is another promising application of APs [Jain96]. For other useful applications see [Wild78], [Fran95].

In this paper we concentrate our attention on minimization of Boolean functions in AP domain.

1.2. Minimization strategies. The following strategies to minimize logic functions in AP domain should be noted.

(A) *Davio'-Deschamps'-Thayse'* strategy based on the idea that to each logic function can be associated with a pseudo-logic function whose value at any vertex is the weight (number of summed terms) of the logic function Taylor expansion at the same vertex. Well known classical theorems of differential analysis allow to obtain a formal expression of this pseudo-logic function. Any pseudo-logic minimization methods (for example, *Hammer-Rudeanu* methods) may then be used to derive the optimal Taylor expansion [Davi78].

(B) *Moraga'* strategy is based on the fundamentals of digital signal processing. In particular, it was shown that AP is one of the simplest arithmetical forms synthesized via changing the transform basis [Mora78].

(C) *Malyugin'* strategy is based on constructing corteges of the linear APs using so-called *algebra of APs* for Boolean functions.

(D) *Perkowski'-Falkowski'* strategy is based on so-called *generalized Adding transform* which is an inverse of Arithmetic transform for each polarity. The Generalized Adding transform is also useful to find a minimal generalized RM expansion for the IS Boolean functions, (see, for instance, [Falk90]).

(E) *Stankovic'-Sasao'-Moraga'* strategy [Stan96] is oriented towards *spectral DDs*.

(F) *Shmerko'-Yanushkevich'* approach combines the spectral approach by *Moraga* and *Malyugin'* algebra of corteges of the linear APs [Anto95], [Boch98], [Yanu95,96,98], [Shme97].

We use the so called staircase strategy originally presented by Zakrevskij [Zakr95,97] for minimizing Incompletely Specified (IS) Boolean functions in Reed-Muller domain and generalized for MVL functions [Zakr96], [Shme97], [Yanu98]. It was shown that the staircase strategy is efficient to obtain a minimal RM form for weakly specified MVL functions (more than 70% of *don't cares*). Our main goal is to answer the next questions:

(i) Is it possible to use the staircase strategy to minimize Boolean functions in the AP domain? If so, under which conditions will this strategy be efficient?

(ii) What are the advantages (disadvantages) of AP representation of Boolean functions (systems of functions, IS functions) ?

We use benchmarks to compare the proposed algorithm with other algorithms.

2. ARITHMETICAL POLYNOMIAL FORM

The 0-polarity AP of Boolean function $f(X)=f(x_1,x_2,\dots,x_n)$ of n variables is expressed as:

$$P(X) = \sum_{j=0}^{2^n-1} p^{(j)} (x_1)^{j_1} (x_2)^{j_2} \dots (x_n)^{j_n}, \quad x_i^{j_i} = \begin{cases} 1, & j_i = 0, \\ 0, & j_i = 1, \end{cases}$$

where $p^{(j)}$ are integer coefficients of AP $P(X)$; $P_c(X) \in (0,1)$; $i=1, \dots, n$; $j=j_1 j_2 \dots j_n$ is binary representation of j . A *literal* (L) of an input variable x_i is signed $x_i^{S_i}$ for $x_i \in S_i$; $x_i^{S_i} = 0$ for $x_i \notin S_i$ and $x_i^{S_i} = 1$, where $S_i \subseteq P_c(X)$. Logic function and AP are mutually related as shown in Fig. 1. A *product* which include literals $x_1^{S_1}, x_2^{S_2}, \dots, x_k^{S_k}$, $k \leq n$, be a *term product*, or *product* (T).

The principal advantage of the APs is their ability to present not only a single Boolean function $f(X)$, but also any system of m Boolean functions $f_1(X), f_2(X), \dots, f_m(X)$. If the system is represented by a matrix of truth vectors $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m$, then we can represent it by the *generalized truth vector* \mathbf{X}_D , consisting of the decimal codes of rows of this matrix. Then the *generalized AP* of the system of functions is written as

$$D(X) = \sum_{j=0}^{2^n-1} d^{(j)} (x_1)^{j_1} (x_2)^{j_2} \dots (x_n)^{j_n},$$

where $d^{(j)}$ are integer coefficients of the generalized AP $D(X)$. This expression can be linearized, i.e. represented by a *linear AP* (no products of variables) using the method from [Maly82,84], [Anto95].

Consider the system with $m=2$ IS Boolean functions with $n=5$ variables, specified by 5 values of each function (see *Example*), written by their IS truth vectors (matrix \mathbf{X}) or by the generalized IS truth vector \mathbf{X}_D . According to the minimization algorithms considered below, it results in the system of Boolean functions represented by an AP: $D(X) = x_4 x_3 + 3x_2 + 3x_1 x_3$.

3. STRATEGY OF THE GZ-AP ALGORITHM

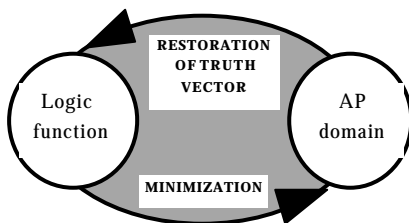


Fig.1. Mutual mapping of truth table of a logic function and its AP

Example. The matrix representation of Boolean function $f(X)$ and truth vectors \mathbf{X} for the system of $m=2$ IS Boolean functions

$$D = \begin{matrix} & x_0 & x_1 & x_2 & x_3 & x_4 \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}, & \mathbf{X} = \begin{matrix} \mathbf{X}_2 & \mathbf{X}_1 \\ \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, & \mathbf{X}_D = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 0 \\ 3 \end{bmatrix} \end{matrix}$$

This strategy is based on the *staircase heuristic* minimization algorithm for logic function in RM domain [Zakr95,97]. The main idea of the staircase strategy is *do not care about don't cares*. The first step of this strategy is to extend columns of the Definition matrix D by adding possible products of literals L as candidates for the resulting quasi-optimal canonical expression. The GZ-AP consists of a *heuristic search* of combinations of vectors from matrix D to find the best combination which is equal to the truth vector \mathbf{X} . The best solution includes the minimum number of literals.

In RM domain, the principle *don't care about don't cares* seems to be efficient for weakly specified logic functions [Zakr95,97], [Shme97]. Given the obtained quasi-minimal RM expansion, we can always evaluate *don't cares* for any combination of variables (Fig.1).

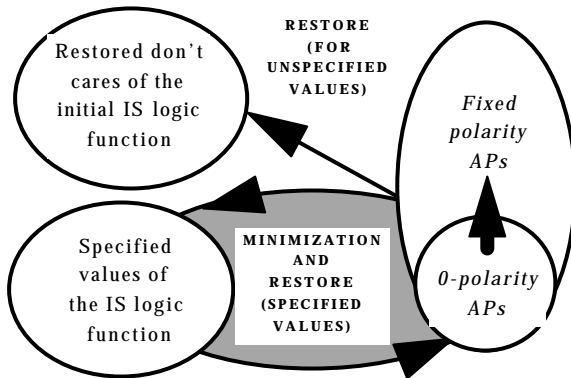


Fig.2. An affect of the staircase minimization strategy of IS logic function for AP domain

Two aspects are distinctive about the staircase strategy for minimization of logic functions in AP domain:

(i) The evaluation of *don't cares* from the quasi-optimal can lead to values outside of the acceptable range for a system of m functions; for example, given $D(X)=x_4x_3+3x_2+3x_1x_3$ then $D(01110)=6$, which is not valid for a two-variable system,

(ii) 0-polarity ΔD (no complemented variables) is not always obtained; for such

the case, we have to find a fixed polarity AP (the same variable may not appear both complemented and uncomplemented) (Fig.2).

4. EXPERIMENTS

All experiments were made on PC Pentium 166MMX working under Unix-Linux OS. The MCNC, PSU (set of benchmarks from the Portland State Univ., USA) and randomly generated weakly specified functions with hundreds variables were used. All measurements of the algorithms were made without taking into account the AP coefficient values. We implement the experiments consisting of the following stages.

4.1. Comparison of GZ-AP algorithm and *Perkowski' - Falkowski' strategies.* We have examined the GZ-AP algorithm as well as *Falkowski'* [Fal95] and *Perkowski - Song's* (EXORCISM-MV3) algorithms for the same MCNC benchmark sets. The evaluated parameters were time t , the number of literals L , and product terms T in the resulting quasi-optimal APs. Table 1 represents a fragment of the experimental results.

Table 1

The minimization results (the number of terms - T , literals - L and time - t) for the GZ-AP algorithm (0-polarity and fixed-polarity APs), algorithm [Fal95], and EXORCISM algorithm (in RM domain)

Name	IN	OUT	GZ-AP 0-polarity		GZ-AP fixed-polarity		Algorithm [Fal95]		EXORCISM-MV3	
			T/L	t [s]	T/L	t [s]	T/L	t [s]	T/L	t [s]
misex1	8	7	no solution		27/121	3.40	20/68	0.1	17/82	0.2
rd53	5	3	11/30	0.00	11/30	0.04	20/45	0.1	14/57	0.4
rd73	7	3	127/448	0.03	127/448	0.80	63/189	0.1	35/188	4.2
squar5	5	8	23/56	0.00	23/56	0.06	23/56	0.1	19/87	0.8
xor5	5	1	31/80	0.00	16/33	0.06	5/5	0.01	5/10	0.1
clip	9	5	240/1212	0.38	240/1212	62.72	206/995	0.2	63/172	7.4

The obtained results show that the average values of terms T and literals L is larger for AP domain than their analogues in RM domain *Falkowski*' [Falk95] and *Perkowski-Song*'s algorithms [Shme97].

4.2. GZ-AP algorithm for 0- and fixed-polarity APs. The GZ-AP algorithm was studied for its possibilities to find optimal AP of 0-polarity (Table 2).

The experiment have shown that for a number of benchmark functions there is no a solution of 0-polarity AP (*misex1* in Table 1, and all benchmarks in Table 3). The numbers of terms T and literals L in AP domain are larger that their analogues in RM domain. Note, that we apply the GZ-AP algorithm for many-output functions (systems of functions), using the generalized AP forms allowed due to AP domain features. For example, for the system with 4 Boolean functions with 4 variables (benchmark *f2*) GZ-AP found one AP

$$D(X)=8x_3+x_2+2x_1+4x_0-8x_0x_3-x_0x_2-4x_0x_1.$$

For benchmark *newtplat* (2 functions, 10 variables) the following a quasi-minimal AP:

$$D(X)=2x_6-x_2-x_0+2x_0x_2-x_0x_2x_8$$

was found.

These APs are not linear, but they can be transformed to linear ones (no products of variables). However, the coefficients will be larger in this case.

4.3. GZ-AP algorithm for weakly specified Boolean functions. In our study

we used two types of test data:

(i) PSU benchmark set which represent 8-variable Boolean functions specified for about 10 % of possible 256 values,

(ii) randomly generated super-large 1-output functions (up to 900 variables).

Experiment 1. The results of testing the GZ-AP algorithm are illustrated in Table 4. We compare the GZ-AP and GZ-RM algorithms for weakly specified functions (PSU benchmarks with 8 inputs and 1 output). Each of the examples results in the synthesized 0-polarity and fixed polarity non-linear APs. For instance,

$$P(X) = x_4 \bar{x}_5 - x_1 \bar{x}_5 (x_1 - \bar{x}_0) \text{ for } ch30f0,$$

$$P(X) = \bar{x}_1 x_4 - \bar{x}_1 x_4 (\bar{x}_1 - \bar{x}_0) \text{ for } ch70f3,$$

$$P(X) = \bar{x}_1 - x_0 x_2 (x_0 - \bar{x}_1) \text{ for } kdd1,$$

$$P(X) = x_0 x_2 - x_0 x_1 x_2 \text{ for } kdd2.$$

Experiment 2. Table 5 illustrates the results on testing GZ-AP algorithm on very large examples $lbn \times m$ [Shme97] to evaluate the efficiency of the proposed strategy.

The experimental results allow us to conclude:

- (i) in most cases, non-linear 0-polarity APs were obtained as results of minimizing IS Boolean functions,
- (ii) for fixed-polarity forms, more effort is required to construct the extended matrix, but a quasi-optimal solution is found faster from this matrix, and
- (iii) the main advantage of this algorithm is that it can be applied to a many-output function (system of functions).

Table 2

The minimization results of the GZ-AP algorithm for Boolean functions (0-polarity APs)

Name	IN	OUT	T/L	t [s]
adr2	4	3	13/26	0.00
adr4	8	5	67/293	0.04
dc2	8	7	183/665	0.08
temp	4	3	15/32	0.01
f2	4	4	7/10	0.00
z4ml	7	4	126/443	0.03
rd84	8	4	251/100	0.13
log8mod_	8	5	108/374	0.05
mlp4	8	8	80/354	0.05
newtpla1	10	2	5/8	0.01
newtpla2	10	4	511/230	0.68
radd	8	5	235/958	0.13
root	8	5	254/101	0.14
sqn	7	3	77/248	0.01
sqr6	6	12	45/132	0.00
sqrt8	8	4	245/989	0.14
majority	5	1	11/38	0.00
newill	8	1	65/289	0.04
sam	5	1	19/57	0.00
exam3	4	1	6/23	0.03
z9sym	9	1	393/180	0.34

Table 3

The minimization results of GZ-AP algorithm for fixed-polarity APs of Boolean functions

Name	IN	OUT	T/L	t [s]
adr2	4	3	13/26	0.00
dc1	4	7	10/49	0.03
m1	6	12	30/179	0.44
newcpla1	9	16	158/119	51.25
newcpla2	7	10	37/212	0.73
newtpla2	10	4	256/102	185.20
newtag	8	1	24/139	6.51
exam3	4	1	6/23	0.03

Table 4

The results of the GZ-AP algorithm for 0- and fixed-polarity APs of Boolean functions with 90% *don't cares* for some PSU benchmarks

Name ^{*)}	0-polarity		fixed-polarity	
	<i>T/L</i>	<i>t</i> [s]	<i>T/L</i>	<i>t</i> [s]
add2	16/3	0.00	7/23	0.65
	8			
ch15f0	21/4	0.01	11/3	0.66
	4		6	
ch76f0	11/2	0.01	3/9	0.46
	3			
ch77f0	7/15	0.00	3/10	0.41
ch22f0	6/13	0.01	3/7	0.41
ch30f0	6/15	0.00	2/6	0.41
ch47f0	15/2	0.00	6/16	0.62
	9			
ch52f4	13/2	0.00	5/15	0.58
	8			
ch70f3	6/14	0.00	2/6	0.48
ch74f1	16/3	0.01	6/20	0.60
	3			
ch83f2	20/4	0.00	14/4	63.83
	2		6	
ch8f0	11/2	0.00	5/14	0.56
	0			
contains	14/3	0.01	9/34	0.47
4 ones	4			
interval1	20/4	0.00	10/3	69.40
	1		3	
interval2	16/3	0.01	11/2	0.59
	5		8	
kdd1	12/2	0.01	2/5	0.50
	1			
kdd10	12/2	0.01	8/35	0.60
	6			
kdd2	2/5	0.01	2/5	0.36
kdd3	14/2	0.01	5/11	0.52
	8			
kdd4	10/1	0.01	1/1	0.46
	9			
kdd5	11/2	0.01	4/13	0.51
	7			
kdd6	11/2	0.01	7/18	0.54
	4			
kdd7	7/18	0.01	3/10	0.41
majority	15/4	0.01	5/19	0.40
gate	1			
mux8	9/22	0.01	5/12	0.42
pal	23/5	0.01	12/3	0.65
	0		6	
pal_dbl_	15/3	0.01	11/2	0.46
output	3		9	
pal_outp	17/3	0.01	10/3	0.62
ut	8		2	
rnd_m1	14/3	0.01	9/27	0.58
	7			

Note: *) 8-input, 1-output benchmarks

Table 5

The minimization results for the GZ-AP and GZ-RM algorithms for randomly generated n -variable Boolean functions of m specified values (0-polarity APs)

Name ^{*)}	n	m	GZ-AP		GZ-RM	
			<i>T/L</i>	<i>t</i> [s]	<i>T/L</i>	<i>t</i> [s]
lb100x200	100	200	180/711	6.34	47/176	0.45
lb100x300	100	300	236/119	11.67	84/333	1.00
			7			
lb100x400	100	400	374/176	20.22	112/483	2.10
			8			
lb300x200	300	200	178/637	61.19	41/152	3.63
lb300x300	300	300	279/112	123.89	59/252	8.16
			4			
lb300x400	300	400	377/162	199.03	87/372	14.83
			9			
lb500x200	500	200	176/616	191.62	37/138	8.21
lb500x300	500	300	275/107	341.32	59/234	17.78
			3			
lb500x400	500	400	375/157	553.61	81/333	38.34
			3			
lb700x200	700	200	181/614	396.78	30/102	18.14
lb700x300	700	300	280/106	767.38	47/176	42.49
			8			
lb700x400	700	400	382/156	1321.0	57/227	68.77
			1	7		
lb900x200	900	200	177/593	652.56	27/91	26.85
lb900x300	900	300	277/103	1219.4	40/142	54.59
			7	1		
lb900x400	900	400	375/153	2010.8	56/211	109.94
			0	6		

Note: *) n -input, 1-output benchmarks

5. CLOSING REMARKS

We have shown that the staircase strategy can be applied successfully to the AP domain. Note, that the considered minimization strategy can be applied to minimize MVL functions too.

Our experiments allows us to conclude that:

(i) For AP domain, the heuristic strategy does not always resulted in a 0-polarity AP; in this case, a fixed polarity AP was found.

(ii) The offered strategy finds a quasi-optimal AP for both a single IS function and a system of logic functions. However, the evaluation of *don't cares* from a quasi-

optimal solution can lead to values outside of the acceptable range for a system of functions.

We see our future work as (i) improve the efficiency of the algorithm by investigating new strategies, for example, [Duec96], (ii) develop an algorithm to find solutions in the class of *linear APs*, since they are easier to evaluate [Anto95], and (iii) apply methods from Information Theory to Arithmetical Logic [Cheu98].

ACKNOWLEDGMENTS

We are thankful to *Prof. D. Bochmann, Prof. C. Moraga (Germany), Prof. S. Rudeanu (Romania)* and *Prof. A. Zakrevskij (Belarus)* for discussions on problems of Arithmetic Logic and their advises.

REFERENCES

- Antonenko V., Ivanov A., Shmerko V. (1995), Linear Arithmetical Forms of k -valued Logic Functions and their Realization on Systolic Arrays, *Automation & Remote Control (USA)*, vol.56, no.3, Pt.2, pp. 419-432
- Bochmann D., Stankovic R., Tomic Z., Shmerko V., Yanushkevich S. (1998), Logic Differential and Integral Calculus: History, Achievements, and Trends, *will be published in Automation & Remote Control (USA)*
- Cheushev V., Shmerko V., Simovici D., Yanushkevich S (1998), Information Estimations for Logical Functions, *Proc. IEEE 28th Int. Symp. on Multiple-Valued Logic, Japan*
- Clarke E.M., Fujita M., Zhao X. (1996), Multi-Terminal Binary Decision Diagrams and Hybrid Decision Diagrams, *In Book: T. Sasao and M. Fujita (Eds.), Representations of Discrete Functions, Kluwer Academic Pub.*, pp. 93-108
- Davio M., Deschamps P., Thayse A. (1978), Discrete and Switching Functions, *Georgi Publishing Co. & N.J.- McGraw-Hill, Inc.*
- DeMicheli G. (1994), Synthesis and Optimization of Digital Circuits, *McGraw Hill, N.Y.*
- Dueck G., Buttler J. (1996), A Heat Quench Algorithm for the Minimization of Multiple-Valued Programmable Logic Arrays, *Computers Elect. Engng.*, vol.22, no. 2, pp. 103-107
- Falkowski B. J., Chang Chip-Hong. (1995), An Exact Minimizer of Fixed Polarity Reed-Muller Expansions, *Int. J. Electronics*, 1995. vol. 79. no. 4. pp. 389-409
- Falkowski B., Perkowski M. (1990), A Family of All Essential Radix-2 Addition/Subtraction Multi-Polarity Transforms: Algorithms and Interpretations in Boolean Domain, *Proc. 23rd IEEE Int. Symp. on Circuits & Systems, New Orleans, USA*, pp. 2913-2916
- Falkowski B., Shmerko V., Yanushkevich S (1997), Arithmetical Logic - Its Status and Achievements, *Proc. Int. Conf. on Applications of Computer Systems, Szczecin, Poland*, pp. 208-223
- Franke D. (1995), Arithmetic Modeling - a Link Between Discrete-Time Systems and Binary Process Control, *Appl. Math. & Comp. Sci.*, vol.5, no.3, pp. 511-521
- Jain J.(1996), Arithmetic Transform of Boolean Functions, *In Book: T. Sasao and M. Fujita (Eds.), Representations of Discrete Functions, Kluwer Academic Publ.*, pp.133-161
- Malyugin V. (1982), Representation of Boolean Functions by Arithmetical Polynomials, *Automation & Remote Control (USA)*, vol.43, no.4, Pt. 1, pp.496-504
- Malyugin V. (1984), Realization of Boolean Function's Corteges by Means of Linear Arithmetical Polynomials, *Automation & Remote Control (USA)*, vol.45, no.2, Pt. 1, pp. 239-245
- Moraga C. (1978), Complex Spectral Logic, *Proc. 8th IEEE Int. Symp. on Multiple Valued Logic*, pp. 149-156
- Shin-Ichi Minato (1997), Arithmetic Boolean Expression Manipulator Using BDDs, *Formal Methods in System Design (Netherlands)*, vol.10, pp. 221-242
- Shmerko V., Holowinski G., Song N., Dill K., Ganguly K., Safranek R., Perkowski M. (1997), High-Quality Minimization of Multi-Valued Input Binary Output Exclusive-Or Sum of Product Expressions for

- Strongly Unspecified Multi-Output Functions, *Proc. Int. Conf. on Applications of Computer Systems, Szczecin, Poland*, pp. 248-255
- Stankovic R., Sasao T., Moraga C. (1996), Spectral Transform Decision Diagrams, *In Book: Representations of Discrete Functions, T. Sasao and M. Fujita (Eds.), Kluwer Academic Publ.*, pp. 55-92
- Vrudhula S.B.K., Pedram M., Lai Y.Te (1996), EDGE Valued Binary Decision Diagrams, *In Book: Sasao T. and Fujita M.(Eds.), Representations of Discrete Functions, Kluwer Academic Pub.*, pp. 107-132
- Wilde D.J. (1978), Globally Optimal Design, *John Wiley & Sons, N.Y.*
- Yanushkevich S. (1995), Arithmetical Canonical Expansions of Boolean and MVL Functions as Generalized RM Series, *Proc. IFIP 10.5 Int. Workshop on Applications of the Reed-Muller Expansions in Circuit Design, Japan*, pp. 300-307
- Yanushkevich S. (1996), Analogues of Boolean Differences and Differentials in Arithmetical Logic, *Proc. Workshop on Boolean Problems, Freiberg, Germany*, pp. 115-120
- Yanushkevich S. (1998), Logic Differential Calculus in Multi -Valued Logic Design, *Techn. University of Szczecin Academic Pub., Szczecin, Poland*
- Zakrevskij A. (1995), Minimum Polynomial Implementation of Systems of Incompletely Specified Boolean Functions, *Proc. IFIP WG 10.5 Workshop on Applications of the RM Expansions in Circuit Design, Japan*, pp. 250-256
- Zakrevskij A., Jaroszewicz S., Yanushkevich S. (1996), Minimization of RM Expansions for Systems of Incompletely Specified MVL Functions, *Proc. Int. Symp. on Methods and Models in Automation & Robotics, Poland*, pp. 1085-1090
- Zakrevskij A. (1997), Search Space Reducing: a Super-Fast Algorithm for Minimum AND/EXOR Implementation of Systems of Weakly Specified Boolean Functions, *Proc. Int. Conf. on Pattern Recognition & Information Processing, Belarus*, pp. 327-331