

Krzysztof MAŁECKI

ZACHODNIOPOMORSKI UNIWERSYTET TECHNOLOGICZNY

## Wyznaczanie symetrii argumentów funkcji logicznych

Dr inż. Krzysztof MAŁECKI

Adiunkt na Wydziale Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego, kierownik Zakładu Układów Cyfrowych i Telekomunikacji. Zainteresowania naukowe związane są z teoretycznymi aspektami projektowania układów oraz z systemami informatycznymi. W ostatnim czasie autor zajmuje się także wykorzystaniem automatów komórkowych w logistyce miejskiej.



e-mail: kmalecki@wi.ps.pl.

### Streszczenie

W praktyce istotnym problemem jest zmniejszenie wejść układu logicznego np. zaprojektowanego dla potrzeb sterowania. Rozpoznanie symetrii argumentów funkcji logicznych pozwala na zmniejszenie liczby wejść takiego układu. W niniejszym artykule autor przedstawia wyniki badań nad algorytmem poszukiwania symetrii zmiennych dla form wielomianowych. Proponowane rozwiązanie dotyczy nie tylko układów logicznych ale także tych zjawisk, które można zapisać w postaci funkcji lub systemu funkcji logicznych.

**Słowa kluczowe:** wielomiany arytmetyczne, symetria funkcji.

### Recognition of variable symmetry of logic function

#### Abstract

In practice, an important problem is the reduction of logic circuit inputs e.g. for conceptual control design. Symmetry identification of logic function arguments permits to reduce the circuit input number. This article contains a theoretical introduction, in the form of a number of definitions concerning arithmetic polynomials and different kind of symmetry of functions. Then the author presents an innovative way to identify the symmetry of arguments (developed algorithms are described in Part Three of this article) and gives a part of many research results on an algorithm for exploration of symmetry variables of different arithmetical forms (Tab. 2, 3, 4). The great advantage of this approach is the ability to search for symmetry variables for the whole system of logic functions, as illustrated by results presented in Tab. 3. Studies have been confronted with the results obtained in the class of RM polynomials (Tab. 4). At the end of this article there are the conclusions of the research proving the validity of the arithmetic form application in the process of identifying the symmetry arguments of logic function system. The suggested solution can be applied not only to the logic circuits but also to those phenomena, which can be put in the form of logic functions or logic function system.

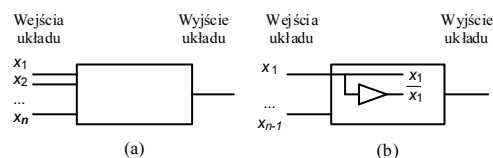
**Keywords:** arithmetical polynomials, symmetry of logic functions.

### 1. Wstęp

Symetria funkcji pozwala na zmniejszenie liczby wejść układu logicznego (rys. 1). Metody wykrywania symetrii funkcji boolowskich zostały opracowane w [1]. Właściwości symetrii zastosowano również w syntezy diagramów decyzyjnych (ang. *Decision Diagrams - DD*) [2]. Wykrywanie symetrii zmiennych stanowi niezbędną część algorytmów minimalizacji funkcji logicznych (np. Espresso [3] i innych [4]), stosowanych obecnie przy projektowaniu układów FPGA. Etap wykrywania symetrii pozwala skrócić czas minimalizacji poprzez wyeliminowanie zmiennych symetrycznych w kolejnych krokach algorytmu.

Rozpoznawanie symetryczności zmiennych, badanej funkcji logicznej, przedstawionych w postaci tablicy prawdy jest problemem NP-trudnym, gdyż zadane funkcje są, w większości przypadków funkcjami zupełnymi. Oznacza to, że dla liczby  $n$  zmiennych wektor prawdy funkcji boolowskiej składa się z  $2^n$  elementów. Aby zmniejszyć rozmiar naszej funkcji należałoby przepro-

wadzić proces minimalizacji i przedstawić zadaną funkcję w postaci np. wielomianu arytmetycznego. Niniejszy artykuł poświęcony jest badaniom mającym za zadanie sprawdzenie możliwości wykorzystania wielomianów arytmetycznych do rozpoznawania symetrii całkowitej oraz częściowej.



Rys. 1. Układ logiczny bez symetrii wejść (a) i z symetrią zmiennych (b)

Fig. 1. Logic circuit without input symmetry (a) and with symmetry of variables (b)

## 2. Podstawy teoretyczne

Formą arytmetyczną funkcji boolowskiej  $f(X)=f(x_1, x_2, \dots, x_n)$   $n$  zmiennych jest zapis postaci:

$$P(X) = \sum_{i=0}^{2^n-1} p^{(i)}(x_1)^{i_1} (x_2)^{i_2} \dots (x_n)^{i_n}, \quad x_i^j = \begin{cases} 1 & \text{dla } i_j = 0 \\ 0 & \text{dla } i_j = 1 \end{cases} \quad (1)$$

gdzie  $p^{(i)}$  są współczynnikami wielomianu  $P(X)$ , liczbami całkowitymi takimi, że  $P(X) \in (0,1)$  [5],  $i=i_1 i_2 \dots i_n$  - binarna reprezentacja  $i$ .

Zaletą wielomianów arytmetycznych jest możliwość przedstawienia nie tylko pojedynczych funkcji logicznych ale także systemów  $m$  funkcji logicznych  $f_1(X), f_2(X), \dots, f_m(X)$ .

Definicja 1. Podziałem  $\rho$  zbioru  $n$  zmiennych  $X$  nazywamy [6] ciąg  $(\rho_1, \rho_2, \dots, \rho_s)$ ,  $s \leq n$  podzbiorów zbioru  $X$ ,  $\rho_j \subseteq X$  takich, że:

$$\bigcap_{j=1}^s \rho_j = X, \quad \forall i, j \in \{1, \dots, s\} \rho_i \cap \rho_j = \emptyset. \quad (2)$$

Przykład. Dla zbioru zmiennych  $X = (x_1, x_2, x_3, x_4, x_5)$  przykładowymi podziałami są:  $((x_1, x_2), (x_3, x_4, x_5))$ ,  $((x_1, x_2), (x_3), (x_4, x_5))$ .

Definicja 2. Permutacją  $\Pi$  podziału  $\rho$  nazywamy działanie polegające na dowolnym przestawieniu zmiennych w każdym z podzbiorów  $\rho$  [6]:

$$\Pi(X) = (\Pi(\rho_1), \Pi(\rho_2), \dots, \Pi(\rho_s)). \quad (3)$$

Dysponując definicją podziału i permutacji  $\rho$  można wprowadzić pojęcie funkcji  $\rho$ -symetrycznej.

Definicja 3. Funkcją  $\rho$ -symetryczną nazywamy funkcję logiczną  $f(X) = f(\rho_1, \rho_2, \dots, \rho_s)$ , dla której dla dwóch dowolnych różnych permutacji  $\rho$  wartość funkcji nie ulega zmianie:

$$\forall \Pi_1, \Pi_2 f(\Pi_1(X)) = f(\Pi_2(X)). \quad (4)$$

Tab. 1. Tablica prawdy systemu dwóch niezupełnych funkcji logicznych  
Tab. 1. The truth table of system of two incompletely defined logic functions

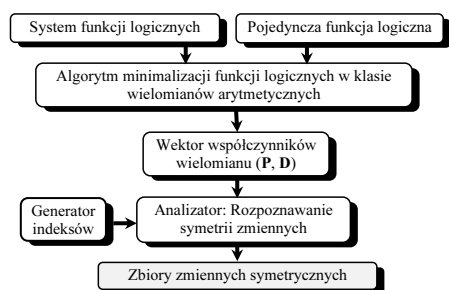
$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$f_1$	$f_0$
0	0	0	1	1	0	1
1	0	0	0	0	0	0
1	1	1	0	1	1	1
1	0	0	1	0	0	0
1	1	0	1	0	1	1

Definicja 4. Funkcją całkowicie symetryczną nazywamy funkcję  $\rho$ -symetryczną, dla której podział  $\rho$  jest zdefiniowany jako pojedynczy podzbiór obejmujący wszystkie zmienne  $((x_1, x_2, \dots, x_n))$ :

$$f(X) \text{ jest c.sym.} \Leftrightarrow \forall \Pi_1(X), \Pi_2(X) f(\Pi_1(X)) = f(\Pi_2(X)).$$

### 3. Struktura i zasada działania pakietu programowego do rozpoznawania symetrii

W niniejszym podrozdziale został zamodelowany system do rozpoznawania symetryczności. Jego struktura została przedstawiona na rysunku 2. Poszczególnymi składowymi są: algorytm minimalizacji funkcji logicznych w klasie wielomianów arytmetycznych [7], generator indeksów oraz analizator sprawdzający symetryczność.



Rys. 2. Struktura pakietu programowego do rozpoznawania symetryczności  
Fig. 2. The structure of software for symmetry recognition

W związku z tym, że opracowany system jest wykorzystywany do rozpoznawania symetryczności w postaci form arytmetycznych, analizowane mogą być zarówno pojedyncze funkcje logiczne jak i systemy funkcji logicznych.

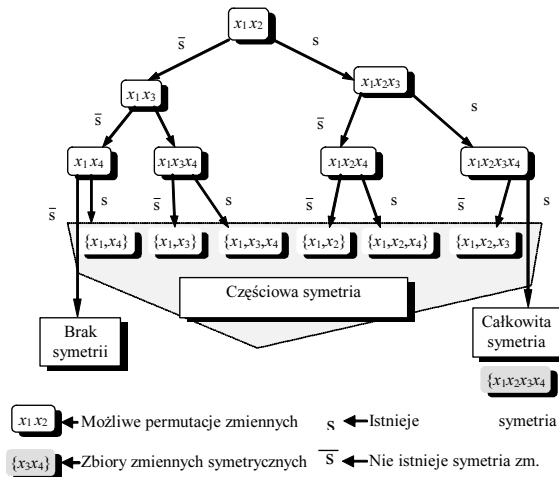
Wynikiem minimalizacji funkcji logicznych jest pojedynczy wielomian arytmetyczny w polaryzacji zerowej zapisany w postaci wektora współczynników (dla pojedynczej funkcji logicznej - P oraz dla systemu funkcji logicznych - D). Następnie, dane te (w formie pojedynczych produktów występujących w końcowej formie rozwiązania) i dodatkowo indeks opisujący dany produkt są przekazywane do analizatora, gdzie wykrywana jest symetria dla poszczególnych zmiennych. Rozwiązanie jest wyświetlane w postaci zbiorów zmiennych symetrycznych.

Zasada działania generatora indeksów. Generator indeksów działa na zasadzie przeszukiwania drzewa wszystkich możliwych permutacji. Rysunek 3 przedstawia, na przykładzie uproszczonego drzewa, zasadę działania generatora indeksów dla przypadku poszukiwania występowania symetrii względem zmiennej  $x_1$ .

Generator indeksów argumentów funkcji całkowicie symetrycznych. Program generuje wszystkie możliwe kombinacje indeksów dla argumentów całkowicie symetrycznych funkcji boolowskich. Oznacza to, że generowane są grupy indeksów dla poszczególnych grup argumentów mających ten sam współczynnik wielomianu arytmetycznego.

Dla funkcji boolowskiej  $n$  zmiennych generator podaje na swoim wyjściu indeksy następujących grup argumentów funkcji: (a) wszystkich możliwych argumentów  $x_1; x_2; \dots; x_n$ , (b) wszystkich możliwych par  $x_1-x_2; x_1-x_3; \dots; x_1-x_n; x_2-x_3; x_2-x_4; \dots; x_2-x_n; \dots; x_{n-1}-x_n$ , (c) wszystkich możliwych trójek  $x_1-x_2-x_3; x_1-x_2-x_4; \dots; x_2-x_3-x_4; x_2-x_3-x_5; \dots; x_{n-2}-x_{n-1}-x_n$ , (d) itd., aż do wszystkich możliwych kombinacji  $n-1$  elementów, np.  $x_1-x_2-\dots-x_{n-1}$ .

Poszczególne grupy indeksów przekazywane na wyjściu generatora trafiają do modułu, który sprawdza, czy wartości elementów wektora prawdy o indeksach uzyskanych przy pomocy generatora są dla rozpatrywanej funkcji boolowskiej takie same. Jeżeli dla którejś z grup indeksów elementy wektora prawdy nie będą przyjmowały tej samej wartości, to oznacza, że funkcja nie jest całkowicie symetryczna.



Rys. 3. Uproszczony algorytm w postaci drzewa, poszukiwania zmiennych symetrycznych dla funkcji boolowskiej czterech zmiennych ( $n=4$ )  
Fig. 3. The simplified algorithm in a tree form, looking for symmetry of variables for boolean function of four variables ( $n=4$ )

Generator indeksów argumentów funkcji częściowo symetrycznych. Program generuje wszystkie możliwe kombinacje indeksów argumentów funkcji boolowskich w celu wykrycia istniejących dla nich symetrii.

Zasada działania generatora indeksów argumentów funkcji częściowo symetrycznych	
<b>Dane wejściowe</b>	Wektor współczynników wielomianu arytmetycznego opisującego pojedynczą funkcję logiczną lub system funkcji logicznych
<b>Dane wyjściowe</b>	Zbiory zmiennych symetrycznych
<b>Krok 1:</b>	Pobierz dwa pierwsze argumenty, dla których ma być rozpatrzone częściowa symetryczność.
<b>Krok 2:</b>	Rozpoznawanie symetrii dla pobranych argumentów.
<b>Krok 3:</b>	Jeżeli rozpoznawanie zostało przeprowadzone dla wszystkich zmiennych zadanej funkcji logicznej, to idź do 5. Gdy symetryczność zmiennych nie została rozpoznana, to weź kolejną parę argumentów badanej funkcji logicznej i idź do 2. Gdy symetryczność zmiennych została rozpoznana, to idź do 4.
<b>Krok 4:</b>	Dobierz kolejny argument i idź do 2.
<b>Krok 5:</b>	Wyświetl zbiory zmiennych symetrycznych.

Rys. 4. Algorytm opisujący zasadę działania generatora indeksów argumentów (zmiennych) dla funkcji częściowo symetrycznych  
Fig. 4. The algorithm of variable index generation for partially symmetric functions

Opisywany algorytm jest wykonywany dla wszystkich par argumentów. W przypadku, gdy dla wygenerowanych indeksów argumentów moduł porównujący wartości wektora prawdy rozpatrywanej funkcji potwierdzi symetryczność zmiennych, to do takiej pary argumentów dobiera się po kolei - pojedynczo - wszystkie z pozostałych argumentów w celu wykrycia zjawiska symetrii trzech argumentów, dla których generator szuka odpowiednich indeksów. Gdy symetria zostanie wykryta, to dobiera się kolejny z możliwych argumentów funkcji w celu wykrycia symetrii poczwórnej, itd. Możliwym końcowym stanem jest wykrycie całkowitej symetrii lub jej całkowitego braku.

### 4. Badania eksperymentalne

Poniższe badania eksperymentalne zostały wykonane na standardowych funkcjach porównawczych MCNC.

Celem przeprowadzonych badań eksperymentalnych było: (a) wyznaczenie pewnych charakterystyk opracowanego systemu, (b) potwierdzenie i udowodnienie metody rozpoznawania symetrii zmiennych dla zespołów funkcji boolowskich, (c) porównanie uzyskanych wyników z wynikami uzyskanymi w klasie wielomianów Reeda-Mullera.

Tab. 2. Fragment wyników minimalizacji dla pojedynczych funkcji boolowskich (liczba produktów -  $T$ , literalów -  $L$ , czas generatora -  $t_{gen}$  i czas całkowity: czas minimalizacji i czas generatora -  $t_c$ )

Tab. 2. The part of minimization results for single-output boolean functions (number of products -  $T$ , literals -  $L$ , time of generator -  $t_{gen}$  and total time: time of minimization and time of generator -  $t_c$ )

Funkcja	WE	T/L	Rozpoznana symetria	$t_{gen}$ [s]	$t_c$ [s]
bw11	5	24/68	$x_1-x_3-x_5; x_2-x_4$	0,00	0,00
bw13	5	7/25	$x_2-x_4; x_3-x_5$	0,00	0,00
bw2	5	4/13	$x_4-x_5$	0,01	0,01
5x01	7	18/81	$x_3-x_4$	0,00	0,01
5x10	7	7/34	$x_1-x_7; x_2-x_3-x_4; x_5-x_6$	0,00	0,00
5x5	7	17/51	$x_1-x_4; x_5-x_6$	0,00	0,00
5x6	7	9/21	$x_5-x_6-x_7$	0,00	0,00
z41	7	40/189	$x_1-x_4-x_7; x_2-x_5; x_3-x_6$	0,00	0,01
z42	7	55/236	$x_1-x_4-x_7; x_2-x_5; x_3-x_6$	0,00	0,01
f53	8	27/98	$x_1-x_2$	0,00	0,01
f55	8	9/21	$x_1-x_2-x_3-x_4$	0,01	0,01
f57	8	3/4	$x_1-x_2-x_3-x_4-x_5-x_6; x_7-x_8$	0,00	0,00
newtag	8	24/116	$x_1-x_3; x_5-x_6; x_7-x_8$	0,00	0,02
sao21	10	380/1868	$x_6-x_{10}$	0,03	0,85
sao22	10	768/4096	$x_6-x_{10}$	0,06	2,11
sao23	10	577/3057	nie wykryto symetrii	0,03	1,73
sao24	10	936/4536	$x_6-x_{10}$	0,06	2,67
Sym10	10	848/4660	$x_1-x_2-x_3-x_4-x_5-x_6-x_7-x_8-x_9-x_{10}$	0,36	2,46

Należy zaznaczyć, że wszystkie funkcje porównawcze (tab. 2) wykorzystane do przeprowadzenia badań są funkcjami zupełnymi. Dlatego też, wyniki uzyskane dla funkcji powyżej 8 zmiennych są wynikami mało zadowalającymi, jeżeli chodzi o stopień minimalizacji. Dzieje się tak dlatego, że algorytm GZ-AP został opracowany z myślą o minimalizacji funkcji niezupełnych. Zwróćmy uwagę na wyniki dotyczące rozpoznania symetrii argumentów dla poszczególnych funkcji. Prawie wszystkie wymienione w powyższej tabeli funkcje są funkcjami częściowo symetrycznymi, za wyjątkiem funkcji *sym10*, która jest całkowicie symetryczna i za wyjątkiem funkcji *sao23*, dla której nie została rozpoznana żadna symetria.

Sprawdźmy teraz efektywność opracowanego pakietu w przypadku rozpoznawania symetrii dla zespołów funkcji boolowskich. Okazuje się, że nie zawsze system funkcji w danej funkcji porównawczej ma rozpoznawalną symetrię (funkcji *f2* w tabeli 3). Rozważając cały zespół nie została rozpoznana żadna symetria. Jednakże funkcje *f2\_1* i *f2\_2* oraz *f2\_3* i *f2\_4* mają wspólne symetrie, co zostało uwidocznione w tabeli 3.

Tab. 3. Fragment wyników minimalizacji dla zespołów funkcji boolowskich  
Tab. 3. The part of result minimization for system of boolean functions

Funkcja	WE	WY	T/L	Rozpoznana symetria	$t_{gen}$ [s]	$t_c$ [s]
F2	4	4	9/20	-	0,00	0,01
F2_1,2	4	2	7/18	$x_1-x_4$	0,00	0,03
F2_3,4	4	2	7/18	$x_2-x_3$	0,00	0,00
Bw26_27	5	2	28/75	$x_2-x_4$	0,00	0,00
Rd53	5	3	11/30	$x_1-x_2-x_3-x_4-x_5$	0,00	0,00
Z4	7	4	64/250	$x_1-x_4-x_7; x_2-x_5; x_3-x_6$	0,00	0,02
F5	8	7	94/400	-	0,00	0,04
Rd84	8	3	255/1024	$x_1-x_2-x_3-x_4-x_5-x_6-x_7-x_8$	0,03	0,15
Sao2	10	3	1020/5116	$x_6-x_{10}$	0,05	3,04

Sprawdźmy, czy rzeczywiście rozpoznawanie symetrii argumentów funkcji logicznych przedstawionych w postaci form arytmetycznych daje nam konkretne korzyści (tabela 4).

W tabeli 4 zostały przedstawione trzy wielowjściowe funkcje testujące. W przypadku rozpoznawania symetrii dla poszczególnych pojedynczych składowych występujących w rozpatrywanych systemach, wyniki są takie same. Oznacza to, że korzystanie z logiki arytmetycznej w celu przeprowadzenia procesu rozpoznawania symetrii jest możliwe a uzyskiwane wyniki są poprawne.

Tab. 4. Fragment badań eksperymentalnych porównania wyników uzyskanych w procesie rozpoznawania symetrii argumentów w klasie wielomianów arytmetycznych i w klasie wielomianów RM

Tab. 4. The part of experimental research for comparison of symmetry recognition by means of arithmetical polynomials and RM polynomials

Funkcja	Numer w systemie	Rozpoznana symetria			
		Wielomiany arytmetyczne		Wielomiany RM	
		Pojedyncze funkcje	System funkcji	Pojedyncze funkcje	System funkcji
f2	f2_1	$x_1-x_4$	$x_1-x_4$	$x_1-x_4$	Brak
	f2_2	$x_1-x_4$	$x_1-x_4$	$x_1-x_4$	
	f2_3	$x_2-x_3$	$x_2-x_3$	$x_2-x_3$	
	f2_4	$x_2-x_3$	$x_2-x_3$	$x_2-x_3$	
rd53	rd53_1	$x_1-x_2-x_3-x_4-x_5$		$x_1-x_2-x_3-x_4-x_5$	Brak
	rd53_2	$x_1-x_2-x_3-x_4-x_5$	$x_1-x_2-x_3-x_4-x_5$	$x_1-x_2-x_3-x_4-x_5$	
	rd53_3	$x_1-x_2-x_3-x_4-x_5$		$x_1-x_2-x_3-x_4-x_5$	
rd73	rd73_1	$x_1-x_2-x_3-x_4-x_5-x_6-x_7$		$x_1-x_2-x_3-x_4-x_5-x_6-x_7$	Brak
	rd73_2	$x_1-x_2-x_3-x_4-x_5-x_6-x_7$	$x_1-x_2-x_3-x_4-x_5-x_6-x_7$	$x_1-x_2-x_3-x_4-x_5-x_6-x_7$	
	rd73_3	$x_1-x_2-x_3-x_4-x_5-x_6-x_7$		$x_1-x_2-x_3-x_4-x_5-x_6-x_7$	

W klasie wielomianów RM niemożliwe było rozpoznanie symetrii dla całego systemu funkcji logicznych. Natomiast w klasie wielomianów arytmetycznych jest to możliwe ale zaznaczyć należy, że nie zawsze istnieje symetria dla wielowjściowej funkcji logicznej. W takim przypadku należy podzielić rozpatrywaną funkcję na mniejsze części i dla nich przeprowadzić proces ponownie. W szczególnym przypadku sprowadzi się to do rozbicia zespołu funkcji na pojedyncze funkcje logiczne.

## 5. Podsumowanie

Autor w niniejszym artykule zaprezentował pakiet programowy do rozpoznawania argumentów funkcji logicznych w postaci form arytmetycznych. Przeprowadzone doświadczenia potwierdziły możliwość rozpoznawania symetrii argumentów pojedynczych funkcji logicznych przedstawionych w postaci wielomianów arytmetycznych oraz możliwości zastosowania logiki arytmetycznej do rozpoznawania symetrii dla zespołów funkcji logicznych.

## 6. Literatura

- [1] Chien-Chang T., Marek-Sadowska M.: Generalized Reed-Muller Forms as a Tool to Detect Symmetries. IEEE Trans. on Computers, no.1, vol. C-45, pp. 33-40, 1996.
- [2] Moller D., Molitor J., Drechsler R.: Symmetry Based Variable Ordering for ROBDDs. Proc. IFIP Workshop on Logic and Architecture Synthesis, pp. 47-53, 1994.
- [3] Brayton R.K., Hatchel G.D., McMullen C.T., Sangiovanni-Vincentelli A.L.: Logic Minimization Algorithms for VLSI Synthesis. Kluwer Academic Publishers, Boston, 1984.
- [4] Łuba T., Jasiński K., Zbierchowski B.: Specjalizowane układy cyfrowe w strukturach PLD i FPGA. Wydawnictwa Komunikacji i Łączności, Warszawa, 1997.
- [5] Falkowski B., Hołowiński G., Małecki K.: Effective Minimization of Logic Functions in RM Domain. Proc. Int. Conf. Applications of Computer Systems, Szczecin, Poland, pp. 248-255, 1997.
- [6] Davio M.J., Deschamps P., Thayse A.: Discrete and Switching Functions. McGraw-Hill Int. Book Co., 1978.
- [7] Małecki K.: Nowy algorytm do minimalizacji systemów funkcji boolowskich w klasie wielomianów arytmetycznych. XX Międzynarodowe sympozjum naukowe studentów i młodych pracowników nauki, Zielona Góra, Polska, pp. 319-323, 1998.