

Dariusz KANIA¹, Waldemar GRABIEC²

¹ POLITECHNIKA ŚLĄSKA, INSTYTUT ELEKTRONIKI

² WOJSKOWA AKADEMIA TECHNICZNA, WYDZIAŁ ELEKTRONIKI

Synteza logiczna przeznaczona dla struktur CPLD z elementami XOR

Dr hab. inż. Dariusz KANIA

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej. Pracę doktorską obronił w 1995, habilitacyjną w 2004r. Jest adiunktem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe koncentrują się wokół programowalnych układów i systemów cyfrowych.



e-mail: Dariusz.Kania@polsl.pl

Mgr inż. Waldemar GRABIEC

Ukończył studia o specjalności telekomunikacja na Wydziale Elektroniki Wojskowej Akademii Technicznej. Aktualnie pracuje na stanowisku starszego wykładowcy w Instytucie Telekomunikacji Wydziału Elektroniki WAT. Jego zainteresowania naukowe dotyczą współczesnych systemów telekomunikacyjnych (głównie systemów transmisyjnych i optotelekomunikacji) oraz techniki cyfrowej w zakresie układów logiki programowalnej.



e-mail: waldemar.grabiec@wel.wat.edu.pl

Streszczenie

W artykule przedstawiono koncepcję syntezy logicznej dla matrycowych struktur CPLD. Zaproponowane rozwiązanie oparte jest na tzw. dekompozycji kolumnowej, natomiast jego ideą przewodnią jest wykorzystanie elementu XOR występującego w blokach logicznych typu PAL większości oferowanych struktur CPLD. Istotą zaproponowanego modelu dekompozycji jest problem poszukiwania dopełnień wzorców kolumn matrycy podziałów pozwalających wykorzystać elementy XOR.

Słowa kluczowe: synteza logiczna, dekompozycja, odwzorowanie technologiczne.

Logic synthesis dedicated for CPLDs with XOR gates

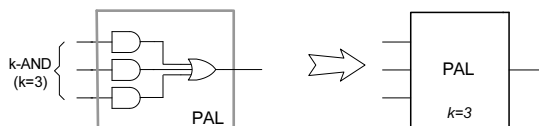
Abstract

This paper presents conception of logical synthesis for CPLDs. Proposed solution bases on column decomposition. The main idea of presented logic synthesis is based on utilization of XOR gates in CPLDs. In proposed conception of logic synthesis we seek of the complement column patterns in partition matrix. This is main idea for utilization XOR gates in PAL-based logical blocks.

Keywords: logic synthesis, decomposition, technology mapping.

1. Wprowadzenie

Większość obecnie dostępnych na rynku struktur CPLD wykorzystuje architekturę charakterystyczną dla układów PAL. Rdzeniem tych struktur jest blok logiczny typu PAL zawierający pewną liczbę iloczynów k (najczęściej $k = 3 \div 8$) dołączonych na stałe do wejść bramki sumy logicznej (rys. 1).



Rys. 1. Struktura bloku logicznego typu PAL zawierającego k -iloczynów
Fig. 1. Structure of PAL-based logic block consisting of k terms

Jednym z głównych problemów syntezy logicznej dedykowanej dla układów CPLD typu PAL jest efektywne wykorzystanie iloczynów. Bloki logiczne struktur CPLD oprócz iloczynów często zawierają dodatkowe elementy, m.in. konfigurowalne przerzutniki, wyjściowe buforów trójstanowe, bramki XOR itp. Elementy te, chociaż mają różnorodne przeznaczenie, mogą z powodzeniem być uwzględniane w procesie syntezy logicznej. W syntezie przeznaczonej dla matrycowych struktur CPLD można wykorzystać elementy dekompozycji, odgrywające kluczową rolę w procesie

syntezy układów realizowanych w strukturach FPGA typu tablicowego (ang. *LUT-based FPGA*). W przypadku struktur matrycowych istota dekompozycji polega na dopasowaniu projektowanego układu do wewnętrznej struktury układu CPLD [1].

Celem artykułu jest zaprezentowanie pomysłu dekompozycji, umożliwiającej wykorzystanie elementu XOR powszechnie występującego w blokach logicznych wielu oferowanych struktur CPLD. Opracowany model dekompozycji jest rozszerzeniem tzw. dekompozycji kolumnowej, którego podstawą jest klasyczny model dekompozycji Curtisa [1, 2, 3]. Zaproponowaną koncepcję syntezy logicznej porównano, na prostym przykładzie, z klasyczną metodą syntezy i metodą syntezy wykorzystującą dekompozycję Curtisa, ukierunkowaną na efektywne wykorzystywanie bloków logicznych typu PAL [1].

2. Klasyczna metoda realizacji funkcji w strukturach typu PAL

Klasyczna metoda realizacji funkcji $f : B^n \rightarrow B^m$ w strukturach CPLD typu PAL związana jest z realizacją zminimalizowanych funkcji $f_i : B^n \rightarrow B$ ($i=1,2,\dots,m$) w postaci sieci składających się z k -iloczynowych bloków typu PAL [1].

Przykład 1

Rozważmy realizację funkcji $y=f(a,b,c,d,e)$ opisanej za pomocą siatki Karnauga (rys. 2a) wykorzystując bloki logiczne typu PAL zawierające trzy iloczyny. W wyniku minimalizacji za pomocą programu ESPRESSO uzyskujemy postać zawierającą 13 implikantów (rys. 2b).

		cde							
		000	001	011	010	110	111	101	100
ab	00	0	1	0	1	0	1	1	1
	01	1	0	1	0	1	0	0	0
	11	0	1	0	1	0	1	1	1
	10	1	0	1	1	1	1	0	1

		.i 5				
		.o 1	.ilb a b c d e	.ob y	.p 13	
		0010- 1				
		1-100 1				
		-0111 1				
		1-111 1				
		01000 1				
		01011 1				
		01110 1				
		00-01 1				
		11-01 1				
		-0010 1				
		1-010 1				
		10-0 1				
		10-1- 1				
		.e				

Rys. 2. Siatka Karnauga funkcji (a) wynik minimalizacji $y.pla$ (b)
Fig. 2. The Karnaugh map of function (a) and result of minimization $y.pla$ (b)

Niech Δ_{f_i} oznacza liczbę implikantów, dla których funkcja $f_i: B^n \rightarrow B$ przyjmuje wartość 1. Niech δ_{f_i} będzie liczbą bloków PAL potrzebnych do realizacji funkcji f_i . Liczbę iloczynów zawartych w bloku typu PAL oznaczmy literą k . W sytuacji kiedy $\Delta_{f_i} > k$ liczba niezbędnych do realizacji funkcji

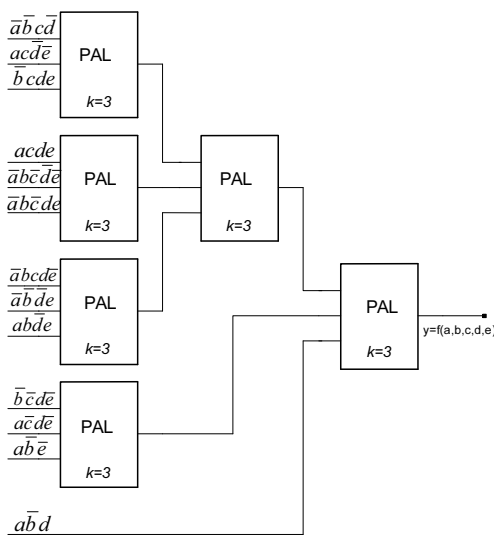
$$k\text{-iloczynowych bloków wynosi } \delta_{f_i} = \left\lceil \frac{\Delta_{f_i} - k}{k - 1} \right\rceil + 1, \text{ gdzie zapis } \lceil x \rceil$$

oznacza najmniejszą liczbę naturalną nie mniejszą od x . W rozważanym przypadku powstaje struktura składająca się z ξ_{f_i} warstw bloków logicznych typu PAL ($\xi_{f_i} = \lceil \lg_k \Delta_{f_i} \rceil$). Dla funkcji przedstawionej na rys. 2 poszczególne wartości wynoszą

odpowiednio: $\Delta_{f_1} = 13, \delta_{f_1} = \left\lceil \frac{\Delta_{f_1} - k}{k - 1} \right\rceil + 1 = \left\lceil \frac{13 - 3}{3 - 1} \right\rceil + 1 = 6$ oraz

$$\xi_{f_1} = \lceil \lg_k \Delta_{f_1} \rceil = \lceil \lg_3 13 \rceil = 3.$$

Realizację funkcji $y=f(a,b,c,d,e)$ metodą klasyczną w oparciu o bloki logiczne typu PAL zawierające 3-iloczyny przedstawiono na rys. 3.



Rys. 3. Klasyczna realizacja funkcji $f: B^5 \rightarrow B$ wykorzystująca bloki logiczne PAL

Fig. 3. Classical implementation of the function $f: B^5 \rightarrow B$ based on PAL-logic blocks

3. Metoda realizacji funkcji wykorzystująca dekompozycję kolumnową

Istota dekompozycji funkcji sprowadza się zwykle do odpowiedniego podziału projektowanego układu na podukłady o zadanej liczbie wejść i wyjść (realizacja układu cyfrowego w strukturach FPGA typu tablicowego). Okazuje się jednak, że dekompozycja może być również wykorzystywana do podziału projektu na części, realizowane w poszczególnych blokach logicznych typu PAL [1]. Głównym ograniczeniem bloków logicznych typu PAL jest liczba wielowejściowych iloczynów. Fakt ten sprawia, że istota dekompozycji przeznaczonej dla struktur typu PAL sprowadza się do minimalizacji liczby wykorzystywanych iloczynów, pośrednio prowadząc do minimalizacji liczby użytych bloków logicznych typu PAL lub dopasowania projektowanego układu do struktury bloków logicznych typu PAL.

Funkcja $y = f(i_1, \dots, i_2, i_1) = f(\mathbf{X}_2, \mathbf{X}_1)$ podlega dekompozycji tzn. $f(\mathbf{X}_2, \mathbf{X}_1) = F[g_1(\mathbf{X}_1), g_2(\mathbf{X}_1), \dots, g_p(\mathbf{X}_1), \mathbf{X}_2]$ wtedy i tylko wtedy, gdy złożoność kolumnowa maczyzy podziałów

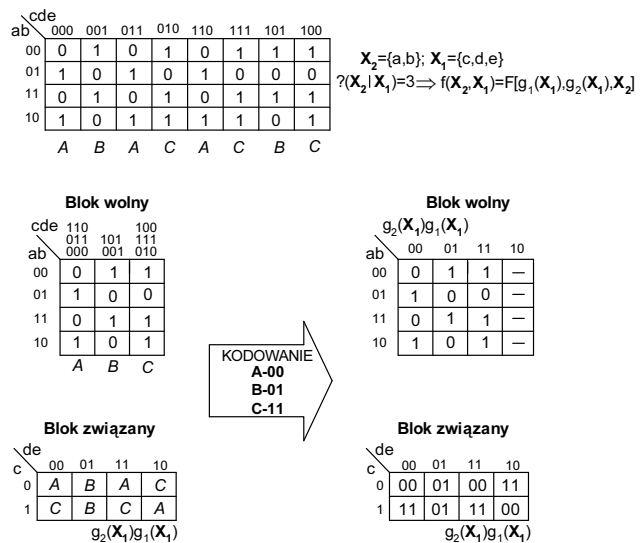
$v(\mathbf{X}_2 | \mathbf{X}_1) \leq 2^p$ [2, 3]. Zbiory \mathbf{X}_1 i \mathbf{X}_2 nazywane są odpowiednio zbiorom **związanym** i **wolnym**, przy czym $\mathbf{X}_1 \cup \mathbf{X}_2 = \{i_1, \dots, i_2, i_1\}$ oraz $\mathbf{X}_1 \cap \mathbf{X}_2 = \emptyset$. Czyli:

$$v(\mathbf{X}_2 | \mathbf{X}_1) \leq 2^p \Leftrightarrow f(\mathbf{X}_2, \mathbf{X}_1) = F[g_1(\mathbf{X}_1), g_2(\mathbf{X}_1), \dots, g_p(\mathbf{X}_1), \mathbf{X}_2]$$

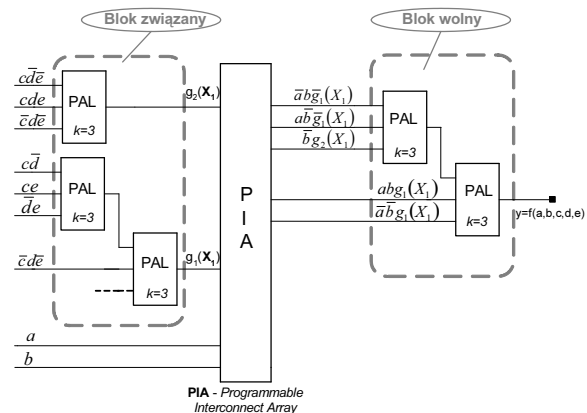
Analizując podział układu będący konsekwencją dekompozycji Curtisa należy zauważyć, że związany jest on z ekspansją całkowitej liczby wyjść. W przypadku układów CPLD prowadzi to do wykorzystania dodatkowo co najmniej p -bloków logicznych typu PAL niezbędnych do realizacji bloku związanego. Fakt ten sprawia, iż zastosowanie dekompozycji może być opłacalne tylko wtedy, gdy w klasycznym podejściu wykorzystanie sprzężeń zwrotnych prowadzi do użycia większej liczby bloków logicznych typu PAL.

Przykład 2

Rozpatrzmy realizację funkcji z przykładu 1. Kolumny siatki Karnaugh tworzą 3 wzorce (kolumny oznaczone literami A, B, C). Stąd złożoność kolumnowa $v(\mathbf{X}_2 | \mathbf{X}_1) = 3$. Z twierdzenia Curtisa wynika, iż do rozróżnienia wzorców kolumn musimy użyć dwóch bitów $g_1(\mathbf{X}_1), g_2(\mathbf{X}_1)$. Siatki Karnaugh opisujące powstające po dekompozycji podukłady czyli bloki: związany i wolny wraz ze strukturą ostateczną układu przedstawiono na rys. 4.

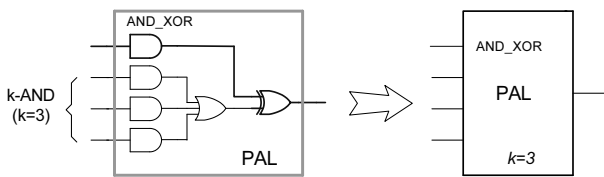


Rys. 4. Realizacja funkcji bazująca na dekompozycji funkcjonalnej
Fig. 4. Implementation of the function based on functional decomposition



4. Metoda dekompozycji ukierunkowana na wykorzystanie bramki XOR

Bloki logiczne większości struktur CPLD zawierają bramkę XOR (rys. 5). Zwykle ten element wykorzystuje się do wyboru aktywności poziomu wyjściowego, wyboru sposobu realizacji funkcji z warunków działania lub niedziałania, modyfikacji typu przerzutnika itp. Okazuje się, że możliwe jest ukierunkowanie całego procesu syntezy układów cyfrowych realizowanych w strukturach CPLD na wykorzystanie elementu XOR. Prowadzi to w wielu sytuacjach do efektywniejszych rozwiązań w stosunku do rozwiązań uzyskiwanych metodą klasyczną i metod zaimplementowanych w narzędziach komercyjnych.

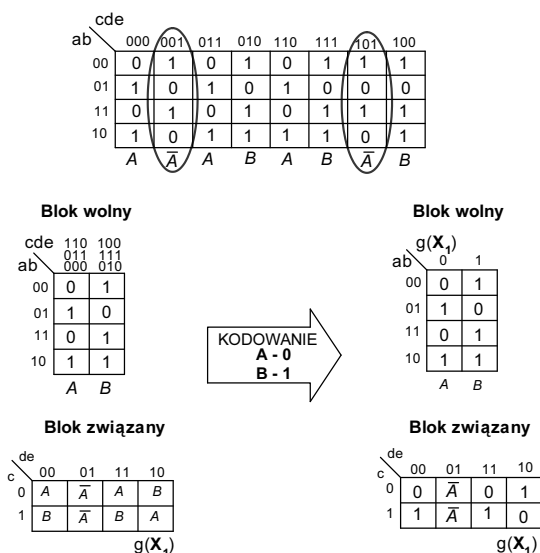


Rys. 5. Struktura bloku logicznego typu PAL zawierającego bramkę XOR
Fig. 5. Structure of PAL-based logic block consisting of XOR gate

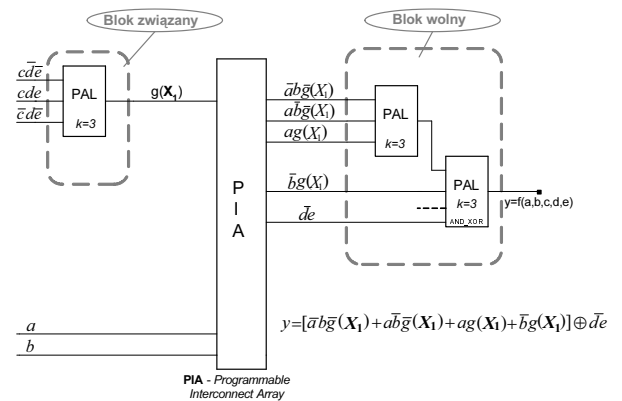
Przykład 3

Rozpatrzmy realizację funkcji z przykładu 1 i 2 uwzględniając obecność w bloku logicznym elementu XOR. Siatka Karnaugh (rys. 6) rozpatrywanej funkcji zawiera trzy typy wzorców kolumn (oznacz.: A, \bar{A}, B), przy czym wzorzec \bar{A} (kolumny zakreślone) stanowi dopełnienie wzorca A [1]. Poszukiwanie złożoności kolumnowej siatki Karnaugh z uwzględnieniem relacji dopełnienia kolumn można wykonywać kolorując wierzchołki grafu niezgodności i dopełnień kolumn [1].

Wzorzec \bar{A} występuje dla dwóch kolumn skojarzonych z wyrażeniem $\bar{d}\bar{e}$. W tej sytuacji możliwe jest rozpatrywanie rozkładu wzorców kolumn tak jakby zawierały one dwa rodzaje kolumn A i B natomiast wzorzec \bar{A} można uzyskać poprzez zanegowanie uzyskanego wyrażenia, wykorzystując do tego celu element XOR. Siatki Karnaugh obrazujące poszczególne etapy syntezy przedstawiono na rys. 6. Realizację funkcji po dekompozycji uwzględniającej w syntezie istnienie bramki XOR przedstawiono na rys. 7.



Rys. 6. Kolejne etapy syntezy ukierunkowane na wykorzystanie bramki XOR
Fig. 6. Implementation of the function based on using XOR gate method



Rys. 7. Implementacja wykorzystująca bramkę XOR
Fig. 7. Implementation with XOR gate

Wnioski końcowe

Zbiornicze zestawienie wyników dla analizowanego przykładu zawarto w tabeli 1. Jak widać wynik syntezy proponowaną metodą wykorzystującą bramkę XOR jest znacznie lepszy pod względem oszczędności układowej w porównaniu z metodą klasyczną. Oczywiście trudno na podstawie jednego prostego przykładu wyciągać daleko idące wnioski. Można jednak z proponowaną metodą wiązać duże nadzieje, ponieważ:

- Wyniki bardzo licznych eksperymentów przeprowadzonych dla popularnych układów testowych pokazały, że metody oparte na dekompozycji opracowane dla struktur CPLD dają znacznie lepsze rezultaty w porównaniu z metodą klasyczną [1],
- Proponowana metoda wykorzystująca bramkę XOR stanowi uzupełnienie dekompozycji kolumnowej i tym samym prowadzi do nie gorszych rozwiązań niż strategię syntezy wykorzystującą dekompozycję kolumnową [1],
- Proponowana metoda dekompozycji stanowi naturalne uzupełnienie strategii syntezy zaimplementowanych w systemie PALDec [1].

Tab. 1. Wyniki końcowe
Tab. 1. Final results

METODA	LICZBA BLOKÓW PAL (K=3)	LICZBA WARSTW LOGICZNYCH
Klasyczna	6	3
Oparta na dekompozycji	5	4
Dekompozycja + XOR	3	3

5. Podsumowanie

W artykule przedstawiono koncepcję syntezy logicznej wykorzystującą bramki XOR powszechnie występujące w strukturach CPLD. Zaproponowana metoda dekompozycji stanowi rozszerzenie tzw. dekompozycji kolumnowej. Jej istota polega na wyszukiwaniu dopełnień wzorców kolumn matrycy podziałów. Natomiast w procesie poszukiwania wzorców kolumn można wykorzystać algorytm kolorowania wierzchołków grafu zgodności i dopełnień kolumn [1].

6. Literatura

- [1] D. Kania: Synteza logiczna przeznaczona dla matrycowych struktur logicznych typu PAL. Zeszyty Naukowe Politechniki Śląskiej, Gliwice 2004
- [2] R.L. Ashenurst: The decomposition of switching functions, Proceedings of an International Symposium on the Theory of Switching, April 1957
- [3] H.A. Curtis: The Design of switching Circuits, D.van Nostrand Company Inc., Princeton, New Jersey, Toronto, New York 1962