Paweł MORAWIECKI [1], **Mariusz RAWSKI** [2]
[1] WYŻSZA SZKOŁA HANDLOWA W KIELCACH, ZAKŁAD INFORMATYKI
[2] POLITECHNIKA WARSZAWSKA, ZAKŁAD PODSTAW TELEKOMUNIKACJI

Utilizing Common Information in Disjoint Decomposition of Multioutput Boolean Functions

Mgr inż. Paweł MORAWIECKI

Otrzymał stopień inżyniera w 2003 roku a rok później stopień magistra na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. Obecnie jest asystentem w Zakładzie Informatyki w Wyższej Szkole Handlowej w Kielcach. Jego zainteresowania naukowe koncentrują się wokół syntezy logicznej układów cyfrowych, algorytmów i struktur danych, obliczeń kwantowych i logiki rewersyjnej.

e-mail: pawelm@wsh-kielce.edu.pl

Dr inż. Mariusz RAWSKI

Otrzymał stopień magistra inżyniera na Wydziale Elektroniki Politechniki Warszawskiej w 1995 roku. Stopień doktora otrzymał na tym samym wydziale w 2000 roku. Obecnie jest adiunktem na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. Jego zainteresowania naukowe obejmują syntezę logiczną układów cyfrowych, narzędzia CAD dla syntezy i optymalizacji logicznej, projektowanie systemów cyfrowych z wykorzystaniem struktur programowalnych PLD.

e-mail: rawski@tele.pw.edu.pl

Abstract

In the article it is described a new method of using a disjoint decomposition as a part of a functional decomposition. The functional decomposition has important applications in many fields of modern engineering and science (FPGA synthesis, information systems, neural networks and many others). The presented algorithm is dedicated to multioutput boolean functions. The concept is based on dividing the complex function into single output functions and then utilizing common information existing in these functions. To test the algorithm, the prototype tool was implemented and the results are presented in the paper.

Keywords: logic synthesis, functional decomposition.

Wykorzystanie wspólnej informacji w dekompozycji rozłącznej wielowyjściowych funkcji boolowskich

Streszczenie

W artykule zostanie przedstawiona nowa metoda wykorzystania dekompozycji rozłącznej jako elementu dekompozycji funkcjonalnej. Dekompozycja funkcjonalna ma zastosowania w wielu dziedzinach elektroniki, informatyki czy telekomunikacji (np. synteza układów FPGA, systemy informacyjne, sieci neuronowe, synteza filtrów cyfrowych). Zaproponowany algorytm dedykowany jest wielowyjściowym funkcjom boolowskim. Działanie algorytmu bazuje na dekompozycji równoległej i wykorzystaniu wspólnej informacji tkwiącej w dekomponowanych podfunkcjach.

Słowa kluczowe: synteza logiczna, dekompozycja funkcjonalna.

1. Introduction

Functional decomposition consists of breaking down a complex system of discrete functions into a network of smaller and relatively independent cooperating subsystems in such a way that the original system's behaviour is preserved, while some constraints are satisfied and some objectives are optimised. The motivation for using functional decomposition in system analysis and design is to reduce the complexity of the problem by divide-and-conquer paradigm and to find an appropriate network of coherent subsystems: a system is decomposed into a set of smaller subsystems, such that each of them is easier to analyze, understand or synthesize.

The efficient functional decomposition process should generate possibly few subsystems which would preserve the logic of the original system. This paper presents the method which can reduce the number of subsystems (subfunctions) in the functional decomposition process.

One of the leading academic system dedicated to the functional decomposition is 'Demain' [1]. The system has implemented a so called 'balanced decomposition'. It is a combination of two procedures: serial decomposition and paralell decomposition. Using both procedures leads to the very good results [2]. However, the decompostion strategy used in Demain brings some limitations. Once the paralell decomposition is applied, two new subfunctions are being decomposed separetely, till the end of the whole decomposition process. Because of this, the common information between two subfunctions is lost and can not be utilized. What we mean by common information here is that some inputs are the same in subfunctions. Additionally, for a given input variables partition the subfunctions may give the similar incompatibility graphs. More details are explained in next sections.

The idea to keep the possibility of utilizing common information is as follows. The function is divided into single outoput subfunctions. Then some of them are chosen to construct G block. Once the G block is constructed, all the single output functions can take part in the next step of the decomposition process. Therefore, functions are not taken apart as in Demain decomposition strategy. Before we introduce more details for this idea, it is needed to present basic definitions and notions regarding functional decomposition.

2. Basic Definitions

Functional decomposition relies on partitioning a switching function into a network of two smaller and independent co-operating sub-functions, in such a way that the original system's behaviour is preserved (Fig. 1).

The set X of function's input variable is partitioned into two subsets: *free variables* U and *bound variables* V, such that $U \cup V = X$.

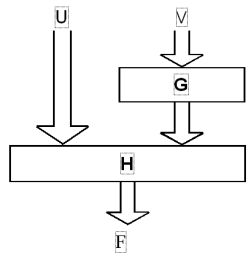

Fig. 1. Schematic representation of the serial decomposition
Rys. 1. Schemat dekompozycji szeregowej

The existence of functional decomposition can be formulated using many different notions (decomposition charts [3], blanket algebra theorems [4], binary decision diagrams [5]). The presented method can be clearly described using decomposition charts and graphs.

Decomposition chart is a matrix $2^{|U|}$ x $2^{|V|}$. The chart has as many columns as minterms induced by bound-set variables and as many rows as minterms induced induced by free-set variables. Each cell in the matrix represents a certain value of f for the corresponding minterm of input variables. Each column (row) is labelled by a minterm of bound-set (free-set) variables.

The next step to verify the existence of serial decomposition is to construct an **incompatibility graph**. Each graph vertex corresponds to the one column from the decomposition chart. Two vertices are connected by an edge only if the vertices (columns) are incompatible. Columns are incompatible when they have different values at least in a one row.

The incompatibility graph is coloured with graph colouring procedure. Each vertex is assigned such a colour, that no two vertices connected by an edge have the same colour assigned. The minimal number of colours is called a **graph chromatic number**.

The next step is to encode the colours. Each colour is assigned the unique binary code. The number of bits used to encode the colours is equal to the number of outputs from the G block. The introduced notions allow to formulate the following theorem.

Theorem 1:

Let *F* be a boolean function and *U* and *V* be sub-sets of its input variables. Let *Γ* be the incompatibility graph based on a given variables partitions. Let *k* be the chromatic number of *Γ* graph and *v* the number of variables in *V* set.

If $\lceil \log_2 k \rceil < v$ then function *F* has a serial decomposition for given *U* and *V* sets.

This theorem is a direct consequence of describing the serial decompositon by decomposition charts and incompatibility graphs. More details can be found in [11].

3. Utilizing common information in decomposition of multioutput functions

The presented algorithm consists of four basic steps:

- dividing the multioutput function into single output functions
- choosing variables partition
- constructing the common G block by merging the incompatibility graphs
- verifying decomposition by applying Theorem 1

Example:

Let us consider two subfunctions F_1 and F_2 described by Table 1 and Table 2. Let us assume the functions have to be decomposed into smaller functions with 3 inputs and 1 output.

Tab. 1. Truth table of function F_1
Tab. 1. Tablica prawdy dla funkcji F_1

a	b	c	e	g	y
0	0	0	0	0	1
0	1	0	0	0	0
0	1	0	0	1	1
1	0	0	0	1	0
0	0	0	1	0	1
1	0	0	1	0	0
1	1	0	1	1	1
0	1	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
0	1	1	1	1	1

Tab. 2. Truth table of function F_2
Tab. 2. Tablica prawdy dla funkcji F_2

c	d	e	f	g	Y
0	0	0	0	0	1
0	0	0	1	0	0
0	0	0	1	1	1
0	1	0	0	1	0
0	0	1	0	0	1
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	1	0	0
1	1	0	0	1	1
1	1	1	0	0	1
1	1	0	0	0	1
1	0	0	1	1	0
1	0	1	1	1	1

For the functions described by Tables 1 and 2, the following variables partitions have been chosen:

U={a,b}, V={c,e,g} for the first function (Table 1)
U={d,f}, V={c,e,g} for the second function (Table 2)

Next step is to construct decomposition charts and incompatibility graphs for both subfunctions.

Tab. 3. Decomposition chart for the function from Tab. 1
Tab. 3. Tabela dekompozycyjna dla funkcji z tablicy Tab. 1

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

Tab. 4. Decomposition chart for the function from Tab. 2
Tab. 4. Tabela dekompozycyjna dla funkcji z tablicy Tab. 2

	ceg							
df	000	001	010	011	100	101	110	111
00	1	-	1	-	-	-	-	-
01	0	1	-	-	0	0	1	-
10	-	0	0	-	1	1	-	1
11	-	-	-	1	-	-	-	-

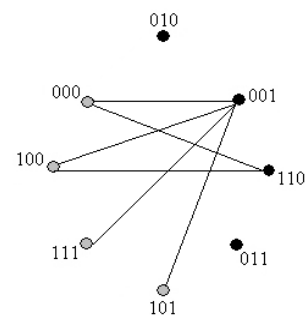

Fig. 2. Incompatibility graph for the decomposition chart (Tab. 3)
Rys. 2. Graf niezgodności dla tablicy dekompozycyjnej (Tab. 3)

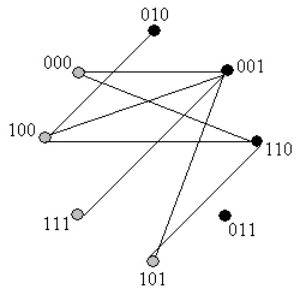

Fig. 3. Incompatibility graph for the decomposition chart (Tab. 4)
Rys. 3. Graf niezgodności dla tablicy dekompozycyjnej (Tab. 4)

It can be noticed that both graphs have the chromatic number equals 2 so the colours could be encoded on one bit. These two functions could be decomposed separately and the result would be four logic cells, each cell with 3 inputs and 1 output. However, the functions have some common information and it could be utilized to produce better results. Variables c,e and g are common so the construction of one G block (G function) is possible. Additionally, the incompatibility graph are very similar.

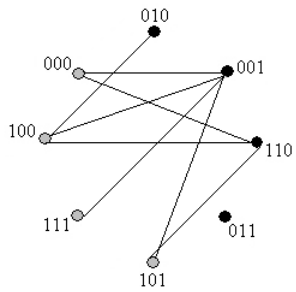

Fig. 4. Incompatibility graph constructed by merging graphs in Fig. 2 and Fig. 3
Rys. 4. Graf niezgodności powstały przez nałożenie grafów z rysunków 2 i 3

The merged graph (Fig. 4) has a chromatic number equals 2 so only one bit is needed to encode the graph. Therefore, these two functions can be decomposed into three logic cells. Figure 5 shows the decomposition scheme.

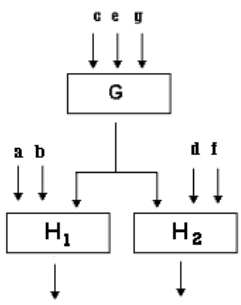

Fig. 5. Decomposition with common G block
Rys. 5. Dekompozycja ze wspólnym blokiem G

The presented algorithm was implemented in a prototype application and some experiments were performed. The functions from the standard benchmark set [6] were decomposed for XC2000 architecture (4/1 or 3/2 logic cells). The application (called DCG – decomposition with common G block) is compared with two academic systems SIS [7] and DEMAIN. Table 5 shows the comparison results. The fewer number of logic cells means the better decomposition quality.

DEMAIN and DCG obtained comparable results and they were better than SIS results.

Tab. 5. Algorithms comparison
Tab. 5. Porównanie algorytmów

Function	Size		DEMAIN	SIS	DCG
	Inputs	Outputs			
Clip	9	5	27	32	27
Sao2	10	4	37	50	41
Plan	13	25	360	333	357
Rd73	7	3	10	13	10
E64	65	65	86	385	86
Z4	7	4	7	9	7
∑			**527**	**822**	**528**

4. Conclusions

The proposed algorithm is dedicated to the multioutput boolean functions. It uses the concept of creating the incompatibility graphs for each single output function and then merging them into one graph. This approach allows to utilize common information between subfunctions through the whole decomposition process. The additional advantage of the method is that computing the incompatibility graphs for single output functions and merging them is less memory and time consuming than the computation for the whole function. This feature was proved in authors' earlier papers [8, 9].

To build an efficient decomposition system, more problems have to be taken into consideration. Before the designed algorithm is applied, one has to decide which subfunctions are taken to create a common G block and which variables should be chosen. This is a difficult task and more research is needed to create the efficient strategy.

5. References

[1] M. Nowicka, "Zrównoważona metoda odwzorowania technologicznego dla układów FPGA. Praca doktorska", Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska, 1999
[2] M. Nowicka, M. Rawski, T.Łuba, „FPGA-based decomposition of boolean functions. Algorithms and implementation. Proceeding of the 6th International Conference on Advanced Computer Systems, 1999
[3] R.L. Ashenhurst. The decomposition of switching functions. Internation Symposium on the Theory of Switching Functions, April 1959
[4] J. A. Brzozowski and T. Łuba, "Decomposition of Boolean Functions Specified by Cubes", Journal of Multiple-Valued Logic and Soft Computing, Vol. 9, Old City Publishing, Inc., Philadelphia, 2003, pp. 377–417.
[5] Chang S.C., Marek-Sadowska M., Hwang T.T., „Technology Mapping for TLU FPGAs Based on Decomposition of Binary Decision Diagrams", IEEE Trans. on CAD, Vol. 15, No.10, pp. 1226-1236, 1996
[6] Collaborative Benchmarking Laboratory, Department of Computer Science at North Carolina State University, http://www.cbl.ncsu.edu: 16080/benchmarks/
[7] Sentovich E.: SIS: A system for Sequential Circuits Synthesis. Electronics Research Laboratory Memorandum, No. VCB/ERLM92/41, University of California, Berkeley, 1992
[8] P. Morawiecki, M. Rawski, „Input Variable Partition Method in Functional Decomposition based on Shannon Expansion", Konferencja Reprogramowalne Układy Cyfrowe, RUC Szczecin 2007
[9] M. Rawski, P. Morawiecki, H. Selvaraj, "Decomposition of Combinational Circuits Described by Large Truth Tables", Proceedings of Eighteenth International Conference on Systems Engineering, Coventry, United Kingdom, September 5-7 2006, pp. 401 - 406.
[10] T. Luba, H. Selvaraj, M. Nowicka, A. Kraśniewski, "Balanced multilevel decomposition and its applications in FPGA-based synthesis", In: Logic and Architecture Synthesis (G.Saucier, A.Mignotte ed.), Chapman&Hall, 1995.
[11] A. Chojnacki, Effective and Efficient Circuit Sythesis for LUT FPGAs Based on Functional Decomposition and Information Relationship Measures. Praca doktorska, Technishe Universiteit Eindhoven, 2004

Artykul recenzowany