The International Workshop
on Discrete-Event System Design, DESDes'01,
June 27÷29, 2001; Przytok near Zielona Gora, Poland

BLOCK SYNTHESIS OF COMBINATIONAL CIRCUITS IN THE BASIS OF PLA AND LIBRARY GATES

Pyotr BIBILO, Natalia KIRIENKO

Institute of Engineering Cybernetics of National Academy of Sciences of Belarus,
Surganov str. 6, 220012 Minsk, BELARUS, <*Bibilo, Kir*>*@newman.bas-net.by*

Abstract. *Circuit realization in one PLA may be unacceptable because of the large number of terms in SOP, therefore a problem of block synthesis is considered in this paper. This problem is to realize a multi-level form of Boolean function system by some blocks, where each block is PLA of smaller size. A problem of block synthesis in gate array library basis is discussed in this paper too. The results of experimental research of influence of previous partitioning of Boolean function systems on circuit complexity in PLA and gate array library basis are represented in this paper.*

Key Words. *Synthesis of Combinational Circuits, Partitioning, PLA, Gate Array Library.*

1. INTRODUCTION

There are different ways of implementation of the control logic of custom digital VLSI circuits. The most important ways are the realization of two-level AND/OR circuits in Programmable Logic Arrays (PLA) basis [9] and the realization of multi-level circuits in library gates basis [7]. Each of them has its advantage and disadvantage. The advantage of PLA-circuits is simplicity of layout design, testing and modification, because circuits are regular. There are effective methods and programs of the PLA-area minimization [9,4]. The disadvantage of two-level PLA-circuit is the large chip area compared with the area required for multi-level library gates circuit. But the synthesis of a multi-level circuit is a very difficult task [5], moreover, such circuits are harder for testing and topological design than PLA-circuits. The implementation of a circuit in one PLA may be unacceptable because of the large size of the PLA. Therefore a problem of block synthesis is considered in this paper. The problem is to realize a multi-level form of Boolean functions system by some blocks, every of which being a PLA of smaller size. The problem of block synthesis in the library gates (LG) basis is considered in this paper too. The results of experimental research of influence of preliminary partitioning of Boolean function systems on circuits complexity in the PLA and LG basis are given.

2. REPRESENTATION OF BOOLEAN FUNCTIONS AND THE BASIS OF SYNTHESIS

It is well known, that the formal (mathematical) model of functioning of multi-output combinational circuit is a system of completely defined Boolean functions [9]. Let a combinational circuit have n inputs and m outputs. One of the forms of Boolean function system representation is the sum-of-product (SOP) system. Let us denote by $D_f(n,k,m)$ the system of Boolean functions $\boldsymbol{f}(\boldsymbol{x})=(f^1(\boldsymbol{x}),\ldots,f^m(\boldsymbol{x}))$, $\boldsymbol{x}=(x_1,\ldots,x_n)$, specified on k common elementary products of Boolean variables $x_1,\ldots,x_n$. Let us represent $D_f(n,k,m)$ by a pair of matrices – the ternary $k\times n$ matrix $\boldsymbol{T}^{\boldsymbol{x}}$ and the Boolean $k\times m$ matrix $\boldsymbol{B}^{\boldsymbol{f}}$. A pair of the appropriate rows of $\boldsymbol{T}^{\boldsymbol{x}}$ and $\boldsymbol{B}^{\boldsymbol{f}}$ represents accordingly the product and the subset of the functions it belongs to their SOPs. The representation of a system of Boolean functions in the form of SOP system will be called two-level representation. We refer to specification of a system of Boolean functions in the form of a system of parenthesised algebraic expression in the basis of logic AND, OR, NOT operators as the multi-level representation.

A programmable logic array is a classical two-level structure for realization of SOP-system of Boolean functions. A system of SOP $D_f(n,k,m)$ can be realized on $PLA(n,m,k)$, which has not less than n input pins, m output pins and k intermediate lines. Elementary products are realized on intermediate lines of matrix AND of PLA, sum-of-products are realized in matrix OR of PLA. The PLA structure is adequate to the system of SOPs $D_f(n,k,m)$. The commutation points between input pins and intermediate lines in AND matrix correspond to fixed (0,1) elements of $\boldsymbol{T}^{\boldsymbol{x}}$, the commutation points between output pins and intermediate lines in OR matrix correspond to elements 1 of $\boldsymbol{B}^{\boldsymbol{f}}$.

The library gates used as basis elements for synthesis of combinational logic circuits are the elements from the logic gate library K1574 [3]. Each element of such a library is characterized by the number of basis gates needed for its location in the chip. There are various elements in the gate library K1574: inverters, multiplexers, buffer elements, gates AND, OR, NAND, NOR, XOR etc.

3. ALGORITHM FOR PARTITIONING MULTI-LEVEL REPRESENTATION OF A SYSTEM OF BOOLEAN FUNCTIONS

Let us consider the multi-level algebraic form of a Boolean function system in AND/OR/NOT basis. Each intermediate or output function is given by a separate SOP. Let non-overlapping SOP subsets $R_1,\ldots,R_h$ form a partition of SOP set $D=\{D^1,\ldots,D^m\}$. $R_1,\ldots,R_h$ are the blocks of the partition with the following parameters: n_i is the number of input variables, m_i is the number of output variables, k_i is the number of products in two-level form of SOP of functions of the block.

Let the restrictions (n^*, m^*, k^*) on block complexity be given: n^* is the maximal number of input variables, m^* is the maximal number of output variables, k^* is the maximal number of products in two-level representation of SOP of functions of the block.

Transformation of multi-level representation to two-level one and determination of the parameter k_i is connected with solving problems of intermediate variables elimination [6] and joint minimization of a system of Boolean functions in SOP form [4, 5, 9].

The problem of partitioning multi-level representation of a system of Boolean functions is to find a partition $R_1,\ldots,R_h$ which fulfils (n^*, m^*, k^*)-restriction and has a minimum number of blocks h.

The main idea of the technique for solving this problem consists in the following. The blocks (or SOP subsystems) are building up step-by-step. A SOP with the maximal number of external variables is chosen out as starting for formation of the next subsystem. Then, SOPs that are most closely connected to this subsystem (on common variables) are added to this subsystem, the elimination of intermediate variables and joint minimization of functions in SOP subsystem are performed. The resulting subsystem is checked for the fulfilment of (*n*, m*, k**)-restriction. If (*n*, m*, k**)-restriction isn't violated, then the next SOP is added to this subsystem, otherwise the constructing the next subsystem begins. As a result, each SOP is in one of the subsystems. This algorithm was described in detail in [2].

4. BLOCK METHOD FOR SYNTHESIS IN PLA BASIS

The block method for synthesis in PLA basis has two procedures. The first procedure is the partitioning multi-level representation of the system of Boolean functions into blocks with (*n*, m*, k**)-restricted parameters. The second procedure is realization of each block by one PLA.

Let the area of a circuit consisting of some (possibly interconnected) PLAs be equal to the sum of the areas of these PLAs. The conditional area of one PLA(*n*,*m*,*k*), evaluated in conditional units (bits), is determined by the formula

$$S_{PLA}^{log} = (2n + m)k \text{ (bit)}. \quad (1)$$

At the level of particular layout used in silicon compiler SCAS [1], the real PLA area is determined by the formula

$$S_{PLA}^{top} = S_{AND,OR} + S_{BND}, \quad (2)$$

where $S_{AND,OR}$ is the area of information matrixes *AND, OR* determined by the formula

$$S_{AND,OR} = 2\left\lceil\frac{k}{8}\right\rceil\left(9\left\lceil\frac{n}{2}\right\rceil + 10\left\lceil\frac{m}{4}\right\rceil\right), \quad (3)$$

S_{BND} is the area of PLA-boundary (load transistors, buffers etc.), determined by the formula

$$S_{BND} = 34{,}992\left\lceil\frac{k}{8}\right\rceil + 85\left\lceil\frac{n}{2}\right\rceil + 49{,}104\left\lceil\frac{m}{4}\right\rceil + \left(\left\lceil\frac{m}{4}\right\rceil - 1\right)\cdot 12{,}276 + 60{,}423. \quad (4)$$

The value of S_{PLA}^{top} determined by formulas (3), (4) is the number of real layout cells that the PLA layout is composed of [1].

5. BLOCK METHOD FOR SYNTHESIS IN THE GATE ARRAY LIBRARY BASIS

The block method for synthesis in gate array library basis has two procedures. The first procedure is the partitioning of multi-level representation of the system of Boolean functions into blocks with (*n*, m*, k**)-restricted parameters. The second procedure is realization of technology mapping each block into the logic gate library. This procedure includes synthesis in the gate array library using a basic method "Cover".

The basic method "Cover" is a process of covering Boolean expressions in AND/OR/NOT basis by elements from the gate library. Previously each multi-place AND(OR) operator of the system is replaced by superposition of two-place AND(OR) operators, respectively. Then the Boolean network is building for each expression, where each node corresponds to two-place operator AND(OR) or one-place operator NOT. The problem of covering Boolean network is to find subnetworks in it, which are functionally equivalent to library elements. The basic method "Cover" was described in detail in [3]. It is experimentally confirmed to be better than the method represented in [7].

6. EXPERIMENTAL INVESTIGATION OF BLOCK METHODS FOR SYNTHESIS IN PLA AND GATE ARRAY LIBRARY BASISES

The block methods for synthesis were realized in computer programs and investigated experimentally. The experiments were done on a series of combinational circuits from well-known MCNC benchmark set chosen from design practice. The programs run on PC Celeron 600, RAM 64 Mb.

Experiment 1. Two realizations of multi-level representation in PLAs were compared: the first one is realization in one PLA; the second one is realization in several PLAs, obtained by partition algorithm. Table 1 shows the results of Experiment 1.

Table 1. The comparison of two realizations of multi-level representation for PLAs: the first is realization in one PLA; the second is realization in several PLAs, obtained by partition algorithm.

Circuit name	n	m	k	One PLA		PLA net			
				S_{PLA}^{log}	S_{PLA}^{top}	n*, m*, k*	h	$\sum S_{PLA}^{log}$	$\sum S_{PLA}^{top}$
x1	51	35	274	37538	26715,29	40,20,500	3	***24521***	***21982,5***
Apex6	132	94	432	154656	99092,84	100,70,900	3	104864	73608,43
						105,80,900	2	113252	79032,04
						70,40,900	4	***75872***	***60442,43***
Apex7	49	35	213	28329	20680,35	30,30,500	3	17104	16603,77
						35,30,500	2	18666	16008,26
						25,20,500	4	***8825***	***10526,71***
example2	85	63	161	37513	28394,06	40,30,900	4	***19501***	20406,34
						60,40,900	2	23450	20429,21
						55,35,900	2	21227	***19194,59***
X4	94	71	371	95347	63474,61	60,35,900	4	***40638***	***35639,54***
						66,50,900	3	46910	46237,44
Frg2	143	139	3090	1313250	794310,4	50,30,500	11	***171581***	***166857,8***
Too_large	38	3	1021	***80659***	***52539,5***	50,2,900	4	102170	73014,45
Ttt2	24	21	222	15318	11824,2	24,10,900	3	10773	13958,77
						24,8,900	5	***8092***	***9907,832***
Cm150a	21	1	796	34228	26343,73	16,2,500	3	***6155***	***6422,373***
						18,3,500	2	11623	14899,51
Frg1	28	3	119	7021	***5904,407***	26,1,100	3	***5783***	6441,453
						28,2,100	2	6523	8530,519
Lal	26	19	117	8307	***6994,927***	30,16,100	5	***4574***	8248,276
Add8	17	9	2519	108317	81949,77	12,9,900	2	***9324***	***9184,462***
X3	135	99	915	337635	209646**,**7	80,50,900	5	***170755***	***122437,9***
Term1	34	10	818	63804	42979,46	30,8,500	2	49896	56050,37
						34,4,500	3	36750	27895,76
						20,8,500	8	***21327***	***23514,69***
Mux	21	1	425	18275	14706,1	16,10,100	2	1862	***3025,974***
						12,6,100	3	***1519***	3270,493
count	35	16	89	7654	7091,571	25,8,100	4	***2769***	***5072,029***
						18,6,100	6	4851	5606,067

Experiment 2. Three realizations of multi-level representation in the basis of logic gate library were compared: the basis method "Cover", combining method and block method of synthesis. The combining method is composed of two steps. The first step is transformation of multi-level representation into two-level representation. The second step is synthesis by basis method "Cover". Table 2 shows the results of Experiment 2.

Table 2. Comparison of basis method, combining method and block realization of multi-level representation

Circuit name	n	m	Basis method		Combining method			Block realization		
			L	S	k	L	S	h	L	S
cu	14	11	41	***204***	35	76	397	4	41	219
comp	32	3	110	555	3	12	***70***	2	23	123
cmb	16	4	27	***142***	56	116	630	3	38	212
cm162a	14	5	43	198	20	57	290	2	24	***117***
mux	21	1	61	***319***	425	1032	6094	5	61	327
count	35	16	111	506	89	188	1024	13	76	***356***
frg1	28	3	298	1683	119	298	1683	3	298	***1683***
cm138a	6	8	9	53	6	16	88	3	9	***53***
cm82a	5	3	20	90	11	14	74	3	13	***57***
9symml	9	1	154	738	30	70	***359***	5	141	727
lal	26	19	117	***576***	117	207	1141	3	180	849
unreg	36	16	96	432	49	80	384	2	80	***384***
z4ml	7	4	102	554	19	26	***140***	2	102	554
x3	135	99	933	***4475***	915	1948	10808	4	966	4679
pcle	19	9	39	181	16	40	214	5	36	***173***
term1	34	10	495	2407	818	2414	13403	7	465	***2360***
cm150a	21	1	61	***261***	796	2136	11886	4	65	309
too_larg	38	3	5217	30145	1027	5273	30473	4	4915	***28392***
ttt2	24	21	299	1609	222	561	2937	5	225	***1197***
sct	19	15	120	584	64	124	618	6	127	***583***
c8	28	18	159	798	70	92	464	4	92	***452***
frg2	143	139	1315	***6560***	3090	15385	85093	15	1361	6886
cm42a	4	10	13	***65***	4	20	94	3	14	72

The minimization of two-level representations of Boolean function system in partition algorithm was done by the program of joint minimization in SOP [8]. The basis method "Cover" uses the computer program from [3].

The notation in tables 1, 2 is as follows:

n – the number of arguments of realized Boolean function system (the number of input pins in the circuit);

m - the number of functions in the system (the number of output pins in the circuit);

k - the number of products in the SOP system (two-level representation);

S_{PLA}^{log} - the conditional area of one PLA(n,m,k), evaluated in conditional units (bits) by the formula (1);

$\sum S_{PLA}^{log}$ - the sum of conditional areas of PLAs, found by the block synthesis method;

S_{PLA}^{top} - the real area of one PLA(n,m,k), evaluated by the formula (2);

$\sum S_{PLA}^{top}$ - the sum of real areas of PLAs, found by the block synthesis method;

S – the circuit complexity in library gates basis (the total number of gates, required for logical elements, i.e. area for elements);

L – the number of logical elements in a circuit;

n^* – partition parameter (the number of arguments of a block);

m^* – partition parameter (the number of functions in a block);

k^* – partition parameter (the number of products in a block);

h – the number of blocks in the partition of multi-level representation of Boolean functions system.

According to the results of the experiment the following conclusions can be stated.

1. The block method for synthesis of multi-level representation by a PLA net is more preferable, than one PLA realization. The gain for area is obtained in 13 circuits from 16. The circuits with the smallest area are printed in bold type (see table 1). Only macroelement area was taken into account in this experiment, and the bound area was not. Thus the final conclusion about replacement of one PLA with PLA net can be done after layout design. Using formula (2) for area calculation is more preferable, than (1). For example, area calculation for Frg1, Lal with (1) gives advantage, but the real PLA net area is more then the one PLA area.
2. The block realization is more preferable for synthesis in logic gate library too. The better (minimum) valuations of circuit complexity are printed in bold type (see table 2). The ttransformation multi-level representation into two-level representation (combining method) is advisable in only three examples; it is not competitive with the basis method "Cover" and the block method.

REFERENCES

[1] P.N.Bibilo, "Symbolic Layout of VLSI Array Macros with an SCAS Silicon Compiler. I", *Russian Microelectronics,* Vol.27, No.2, pp.109-117, 1998

[2] P.N.Bibilo & N.A.Kirienko, "Partitioning a System of Logic Equations into Subsystem under Given Restrictions", *Proc. of the Third International Conference on Computer-Aided Design of Discrete Devices* (*CAD DD'99*), Minsk, Republic of Belarus, Vol.1, pp.122-126, 1999

[3] P.N.Bibilo & V.G.Litskevich, "Boolean Network Covering by Library Elements", *Control Systems and Computers,* Kiev, Ukraine, Vol.6, pp.16-24, 1999, (in Russian)

[4] K.R.Brayton, G.D.Hachtel, C.T.McMullen & A.L.Sangiovanni-Vincentelli, *Logic minimization algorithm for VLSI synthesis.* Kluwer Academic Publisher, Boston, e.a., 1984

[5] K.R.Brayton, R.Rudell, A.L.Sangiovanni-Vincentelli & A.R.Wang, "MIS: A Multiply-level Logic Optimisation Systems", *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* Vol.CAD-6, pp. 1062-1081, 1987

[6] L.D.Cheremisinova & V.G.Litskevich, "Realization of operations on systems of logic equations and SOPs", *Logical design,* Minsk: Institute of Engineering Cybernetics of National Academy of Sciences of Belarus, pp.139-145, 1998, (in Russian)

[7] F. Mailhot & G. De Micheli, "Algorithms for technology mapping based on binary decision diagrams and on Boolean operations", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.12, No 5, pp.599-620, 1993

[8] N.R.Toropov, "Minimization of systems of Boolean functions in DNF", *Logical Design,* Minsk: Institute of Engineering Cybernetics of National Academy of Sciences of Belarus, pp.4-19, 1999, (in Russian)

[9] A.D.Zakrevskij, *Logic Synthesis of the Cascade Circuits,* Nauka, Moscow, 1981, (in Russian)