

TOMASZ MICHALSKI*, ZBIGNIEW KOKOSIŃSKI**

FUNCTIONAL DECOMPOSITION OF COMBINATIONAL LOGIC CIRCUITS WITH *PKMIN*

DEKOMPOZYCJA FUNKCJONALNA KOMBINACYJNYCH UKŁADÓW LOGICZNYCH W PROGRAMIE *PKMIN*

Abstract

In this paper, an application of the *PKmin* program for functional decomposition of multi-input multi-output combinational circuits is presented. The main focus is on balanced multi-level decomposition of logic circuits into minimal number of blocks, such as LUTs in FPGAs. Reduction of the input redundancy is available. Decomposition schemes include parallel, joint/disjoint serial and a mixed one. The decomposition with *PKmin* can be automated by means of a heuristic algorithm or can be supervised by the designer. A distinctive feature of *PKmin* is the visualization of the design steps and the final layout of blocks and their interconnections. *PKmin* is compared in an example with the program *DEMAIN*.

Keywords: combinatorial logic circuit, FPGA, functional decomposition, LUT, PKmin

Streszczenie

W artykule przedstawiono zastosowanie programu *PKmin* do dekompozycji funkcjonalnej wielowyjściowych układów kombinacyjnych, a w szczególności do wielopoziomowej dekompozycji układów na minimalną liczbę bloków funkcjonalnych, takich jak komórki LUT w makrokomórkach FPGA. Możliwa jest wstępna redukcja wejść, a następnie dekompozycje: równoległa, szeregową łączną i rozłączną oraz mieszana. Proces dekompozycji może być zautomatyzowany lub nadzorowany przez projektanta w trybie interaktywnym. Wyróżnikiem *PKmin* jest wizualizacja procesu projektowania. Końcowy schemat układu zawiera bloki składowe LUT oraz ich połączenia. Program *PKmin* został porównany na przykładzie z programem *DEMAIN*.

Słowa kluczowe: układ kombinacyjny, FPGA, dekompozycja funkcjonalna, LUT, PKmin

* M.Sc. Eng. Tomasz Michalski, Nokia Solutions and Networks, Wrocław, Poland.

** Ph.D. Eng. Zbigniew Kokosiński, Department of Automatic Control and Information Technologies, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

1. Introduction

PKmin is a powerful design tool developed at the Cracow University of Technology for computer-aided synthesis of combinational logic circuits [14]. It supports the synthesis of multi-input multi-output (MIMO) logic circuits for Full Custom, Semi Custom and FPGA technologies. *PKmin* offers a number of algorithms for automatic reduction of function arguments, several algorithms for solving the problem of minimal cover and two approximate algorithms for logic function minimization. The detailed description of algorithms and experimental verification of their efficiency for 2-level logic synthesis was reported in [3]. In addition, the *PKmin* tool provides a functional decomposition of combinational MIMO circuits for LUT-based synthesis in FPGAs.

The aim of this paper is to present the properties of the *PKmin* design system as a prospective tool for the decomposition of logic circuits. In section 2, we characterize various methods of functional decomposition. An approximate algorithm used in *PKmin* for mixed serial-parallel balanced decomposition of logic functions is presented in section 3. If required, a prior reduction of function inputs is performed. The *PKmin* program is described in section 4. An important feature of that program is its possibility to speed up design in fully automatic mode. The resulting decomposition of the MIMO circuit contains a near minimal number of combinational blocks, such as LUTs in FPGA. For didactic purposes, *PKmin* provides a unique feature, i.e. a visualization of the decomposition phases in interactive mode, which makes the whole design process disclosed to the user. In section 5, the paper brings an illustrative example of *PKmin* features and usage. Section 6 contains conclusions of the paper.

2. Functional decomposition methods

The need for circuit decomposition has several reasons. One of them is a limited number of I/O in a single chip. In that case, the MIMO circuit has to be decomposed properly into several blocks that fit into available chips, taking into account their internal interconnection capabilities and minimizing external interconnections. For FPGA devices, containing standard size LUTs for implementation of combinatorial logic, the decomposition process aims to use such specific structures with their design limitations. In both approaches, the number of building blocks has to be minimized [1, 6, 7, 11].

There are two main decomposition schemes of MIMO circuits – serial and parallel [4, 8, 9, 13]. Each decomposition scheme can be applied only when certain necessary and sufficient conditions are satisfied. The serial decomposition is either joint or disjoint. In each step of serial decomposition, the number of circuit levels and input-output delay are increasing. In parallel decomposition, the number of circuit levels and input-output delay is minimized. In many cases, a mixed decomposition is used, combining the properties of both schemes. Balancing circuit decomposition helps to reach a tradeoff between contradictory design requirements.

One particular serial decomposition scheme was proposed by Kapralski and Skarbek [2]. The original circuit is divided into two 2-level blocks. At the first level, the minimum number of outputs is obtained; they form a subset, so called *basis* of the output set. At the second level, the remaining outputs are obtained from that basis. The basis is computed

from true table representation of a MIMO circuit using an exact algorithm MINBASE with exponential time complexity or an approximate polynomial algorithm (DRMAX, MINHMAX or MAXERMIN) [2]. All the above algorithms can be used for the reduction of logic function arguments (redundant circuit inputs). Another serial decomposition scheme was developed by Sasao and Matsuura [12]. The purpose of the decomposition is a cascaded structure of the resulting circuit, satisfying an equal size of the building blocks.

3. Mixed balanced decomposition

Mixed balanced decomposition considered in this paper is a top-down method of using both serial and parallel decomposition schemes, according to an algorithm in which a serial-parallel strategy is applied in order to satisfy certain design criteria, for instance the minimal number of components or the minimal number of block levels. The balanced strategy has to maximize advantages and avoid disadvantages of both schemes. Parallel decomposition splits function inputs into subsets of arguments essential for its subfunctions, increasing the number of building blocks. In many cases, smaller blocks are more vulnerable for efficient serial decompositions. On the other hand, serial decomposition leads to minimization of the function due to extraction of common variables, but simultaneously, the number of block levels increases, while the circuit speed decreases. Having the above in mind, the *DEMAIN* and *PKmin* subsystems devoted for balanced decomposition task have been designed. Both applications support interactive decomposition processes with a built-in help system in the form of design hints and providing some flexibility in selections of the next step of decomposition. In general, the *DEMAIN* system offers more options for the designer. In *PKmin*, the spectrum of suggested solutions is narrow because more decisions are made by the system. Guided decomposition is attractive and efficient for relatively small tasks. The interactive design becomes impractical when the function complexity increases. Most complex circuits have to be decomposed in an automated manner. Still, there is some room for tuning the system by the user. In particular, the designer has to decide about the reduction of function arguments and the reduction algorithm as well as about LUT size and some other priorities. On the next page, we present a simplified description of automatic mode of *PKmin*. To a certain degree, it follows the ideas presented in [8] and [9], but differs in details, giving priority to parallel decomposition when possible. Therefore, the decomposition results obtained by both design tools are also different. For the sake of brevity, detailed descriptions of parallel and serial decomposition can be skipped here.

4. *PKmin* program

Balanced functional decomposition of a combinational circuit into LUT cells is only one of the applications of *PKmin*. After selection in GUI a language version (PL, EN) and reading an input text file (*.k) that contains true table for the given circuit, one needs to perform the following sequence of steps: select **Synthesis** task from menu **Options** and then **Multi-level synthesis mode**; select **Functional decomposition** and **FPGA** technology in **Synthesis mode** window. If the **Minimal-argument realization** option is accepted in the **Reduction**

of input variables window, then the **Argument reduction method**, in many cases essential for efficient optimization, has to be chosen (either **Minimum basis** or **Partition calculus**) as well as **Algorithm for input matrix reduction** (one of four

Algorithm *BalancedDecomposition-PKmin*

Input: A – an input block, $in(A)$ – number of A inputs, $out(A)$ – number of A outputs.

Output: decomposition of the input block A into n -input m -output blocks LUT $_i$, $1 \leq i \leq k$, where k is the number of blocks.

BalancedDecomposition-PKmin(A,n,m)

```

repeat
  if  $in(A) < out(A)$ 
    then
      ParallelDecomposition(A,G1,G2)
    else
      begin
        SerialDecomposition(A,G,H,serial);
        if serial=false
          then
            begin
               $in(G)=n$ ;
               $out(G)=m$ ;
              if  $out(G) \leq out(A)$ 
                then
                  begin
                    ParallelDecomposition(A,G1,G2);
                    SerialDecomposition(G1,G,H,serial)
                    SerialDecomposition(G2,G,H,serial)
                  end
                else
                  begin
                    repeat
                       $in(G)=in(G)+1$ ;
                      SerialDecomposition(A,G,H,serial)
                    until serial=true or  $in(G)=n-1$ ;
                    if  $in(G)=n-1$ 
                      then
                        begin
                           $in(G)=n$ ;
                           $out(G)=out(G)+1$ ;
                          if  $out(G)=in(G)$  then  $in(G)=in(G)+1$ ;
                        end;
                    if  $in(G) < in(A)$ 
                      then SerialDecomposition(A,G,H,serial)
                      else ParallelDecomposition(A,G1,G2);
                  end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

until all resulting decomposition blocks have the size required for (n,m)-LUT implementation;

available options). **Maximum computation time** in seconds for this subtask has to be set out.

There are several settings of program parameters related to the decomposition task:

- 1) **Basic cell size** (number of LUT inputs, number of LUT outputs),
- 2) **Decomposition mode** (interactive, automatic),
- 3) **Parallel decomposition parameters** (max. number of solutions from Exact-Cover, max. number of checked decomposition variants),
- 4) **Show intermediate results** (checkbox).

After setting the decomposition parameters, the program starts the decomposition in the selected mode. The conducted tests confirmed that the automatic mode is only slightly worse than the time-consuming interactive mode, assuming that many independent trials are often required for the latter mode to obtain a good solution. In the interactive mode, each decomposition step has to be confirmed together with its decomposition strategy (serial, parallel). The program supports the decision concerning a parallel balanced decomposition by displaying both parallel components, while selection of serial decomposition opens the possibility to correct the cell size in the given step. The progress of decomposition is reported in the program main window and in the form of a color diagram in a separate **Block diagram of decomposition** window. The visualization of decomposition process is an exclusive feature of *PKmin* and becomes a part of final design report displayed in the main window. It allows for a fast evaluation of the decomposition progress after each step; therefore, it is very attractive for designers. After completing the decomposition process, true tables for all LUTs are generated and displayed in the main window. However, the output text file does not contain visualization of the decomposition.

An exemplary input and output of the *PKmin* program is shown in Fig. 1, 2 and 3. The argument reduction method used in this case is the Partition calculus.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y1	Y2
0	0	1	0	1	1	1	0	1	0	0	0
1	0	1	0	0	1	0	1	0	0	0	1
0	1	0	0	0	1	1	1	1	0	0	1
1	0	1	1	1	0	1	0	1	1	0	0
1	1	0	0	0	1	0	0	1	1	0	1
0	1	0	0	0	1	0	1	1	0	0	0
1	1	1	0	1	0	0	1	1	0	0	0
0	1	0	0	1	1	0	0	0	0	0	1
0	1	0	1	0	0	0	0	1	0	0	1
0	1	1	1	1	1	1	0	1	1	1	0
0	0	0	0	0	1	0	1	0	0	1	0
1	1	0	1	1	1	0	0	1	1	1	1
0	1	0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	0	1	0
1	1	1	1	0	1	0	0	0	1	1	0
1	1	1	1	1	0	1	0	0	1	1	0
1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	0	0	0	0	0	0	0	1	1
1	1	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1
1	1	1	1	1	0	0	0	1	0	1	0
1	0	1	0	1	1	1	1	0	1	1	1
0	1	1	0	0	0	0	1	1	0	1	0
0	1	0	0	1	1	1	0	0	0	1	1

Fig. 1. True table of a combinational circuit before decomposition

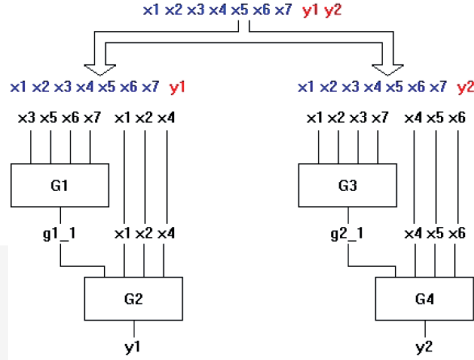
Stage of functional decomposition

Declared size of cell:

Number of inputs: 4

Number of outputs: 1

Block diagram of decomposition:



Number of cells: 4

Number of net's levels: 2

Fig. 2. Visualization of a functional decomposition of the circuit

Results of synthesis

Truth tables of cells:

Cells of level 1:

Cell G1

inputs: x_3, x_5, x_6, x_7

outputs: g_{1_1}

```

1110 1
0000 0
0110 1
1111 0
0011 0
0010 1
1010 1
1000 1
1011 1
0111 0
0001 0
1101 1
1100 0
    
```

Cell G3

inputs: x_1, x_2, x_3, x_7

outputs: g_{2_1}

```

0010 1
1000 1
0000 0
1111 1
1001 1
1010 0
0100 1
0111 1
0001 0
1101 0
1110 0
1011 0
    
```

Cells of level 2:

Cell G2

inputs: x_1, x_2, x_4, g_{1_1}

outputs: y_1

```

0011 0
1010 0
1100 0
1001 0
0101 0
0110 1
0010 1
1111 1
0001 1
1110 1
1101 1
0000 1
1011 1
    
```

Cell G4

inputs: x_4, x_5, x_6, g_{2_1}

outputs: y_2

```

1111 0
1001 1
1110 1
0111 0
1011 1
1010 0
0010 0
0011 1
1000 0
0001 0
0101 0
0000 1
0110 1
1100 1
1101 1
    
```

Fig. 3. True tables of LUT cells of the decomposed circuit

5. A design example

There are many benchmark sets available for testing many aspects of logic synthesis. For the purpose of this paper, we propose to focus on a nontrivial example, which can be captured at first glance. It is a relatively small instance of binary-to-BCD code converter. The focus is on circuit inputs, outputs, decomposition types and levels, connections of LUTs as building blocks. The tables of the resulting LUT functions are skipped here without loss of technical soundness.

Example

Find an approximate solution to the functional LUT-decomposition of binary-to-BCD code converter for the range of integer values $[0, \dots, 100]$ assuming that (4,1)-LUTs are available.

Solutions

1. BIN2BCD circuit decomposition resulting from *PKmin* program in the interactive mode consists of 14 (4,1)-LUTs on three levels (see Fig. 4) :
 - G1: (x2, x3, x4, x5, g1_1)
 - G2: (x1, x2, x3, y1)
 - G3: (x7, y8)
 - G4: (x1, x2, x3, g1_1, y2)
 - G5: (x1, x2, x3, g1_1, y3)
 - G6: (x6, g2_2, g3_1, g3_2, y4)
 - G7: (x6, g2_2, g3_1, g3_2, y5)
 - G8: (x6, g3_1, g3_2, y6)
 - G9: (x6, g2_2, g3_1, g3_2, y7)
 - G10: (x1, x2, x3, x4, g2_1)
 - G11: (x1, x2, x3, x4, g2_2)
 - G12: (x1, x2, x3, x4, g2_3)
 - G13: (x5, g2_1, g2_2, g2_3, g3_1)
 - G14: (x5, g2_1, g2_2, g2_3, g3_2)
2. In the automatic mode BIN2BCD circuit decomposition resulting from *PKmin* program in fully automatic mode consists of 17 cells – (4,1)-LUTs on two levels (see Fig. 5) :
 - G1: (x2, x3, x4, x5, g1_1)
 - G2: (x1, x2, x3, y1)
 - G3: (x7, y8)
 - G4: (x1, x2, x3, g1_1, y2)
 - G5: (x1, x2, x3, g1_1, y3)
 - G6: (g2_1, g2_2, x4, x6, y4)
 - G7: (g3_1, g3_2, x4, x6, y5)
 - G8: (g4_1, g4_2, x3, x5, y6)
 - G9: (g5_1, g5_2, x4, x6, y7)
 - G10: (x1, x2, x3, x5, g2_1)
 - G11: (x1, x2, x3, x5, g2_2)
 - G12: (x1, x2, x3, x5, g3_1)
 - G13: (x1, x2, x3, x5, g3_2)
 - G14: (x1, x2, x4, x6, g4_1)
 - G15: (x1, x2, x4, x6, g4_2)
 - G16: (x1, x2, x3, x5, g5_1)
 - G17: (x1, x2, x3, x5, g5_2)

3. For a reference, another BIN2BCD circuit decomposition resulting from DEMAIN program in the interactive mode consists of 13 cells – (4,1)-LUTs on three levels (cf. [4]):

G1: (x1, x2, x3, x4, y1)
 G2: (x1, x2, x3, x4, y2)
 G3: (x1, x2, x3, x4, g1_1)
 G4: (x1, x2, x3, x4, g1_2)
 G5: (x1, x2, x3, x4, g1_3)
 G6: (x5, g1_2, g1_3, g2_1)
 G7: (x5, g1_2, g1_3, g2_2)
 G8: (x5, g1_2, g1_3, y3)
 G9: (x6, g1_1, g2_1, g2_2, y4)
 G10: (x6, g1_1, g2_1, g2_2, y5)
 G11: (x6, g1_1, g2_1, g2_2, y6)
 G12: (x6, g1_1, g2_1, g2_2, y7)
 G13: (x7, y8)

When comparing all three designs, one can see that interactive decompositions with *PKmin* and *DEMAIN* can result in less LUT cells, but more decomposition levels. In the automatic decomposition mode in *DEMAIN*, the main objective is to minimize the number of LUTs, but the outcome can be worse than in the interactive mode. On the other hand, automatic decomposition with *PKmin* can result in less decomposition levels and more LUTs. This is possible due to the properties of *PKmin* in which parallel decomposition has a priority over serial decomposition in the mixed balanced decomposition in fully automatic program mode.

According to [4], true table of BIN2BCD written in AHDL tdf file and compiled in *MAX+PLUSII* system results in 131 cells of FLEX10K device. This structure is outperformed by all the above designs. Moreover, since there is no fully parallel decomposition of the input circuit, the designed converter must have at least two levels as in our 17-LUT design with *PKmin*. Therefore, the high speed of that circuit claimed in the literature is questionable. Another tool – *Xilinx WebPACK ISE* – generates much better solution consisting of 22 LUTs [5].

In the interactive mode, *DEMAIN* allows the user to apply different sizes of LUTs in the balanced circuit decomposition. Similarly, in the interactive mode, *PKmin* supports flexibility of selection of available LUT sizes in serial decomposition. Reduction of redundant inputs is usually a highly reasonable option.

Although both programs support interactive design process providing hints concerning possible options at the given stage of the design, there is no guarantee that the final decomposition is optimal taking into account the number of LUT cells. The requirements concerning minimal LUTs' count and minimal number of levels seem to be contradictory. Last, but not least, for large circuits, the interactive mode with its possibly exponential growth of various design paths seems to be impractical.

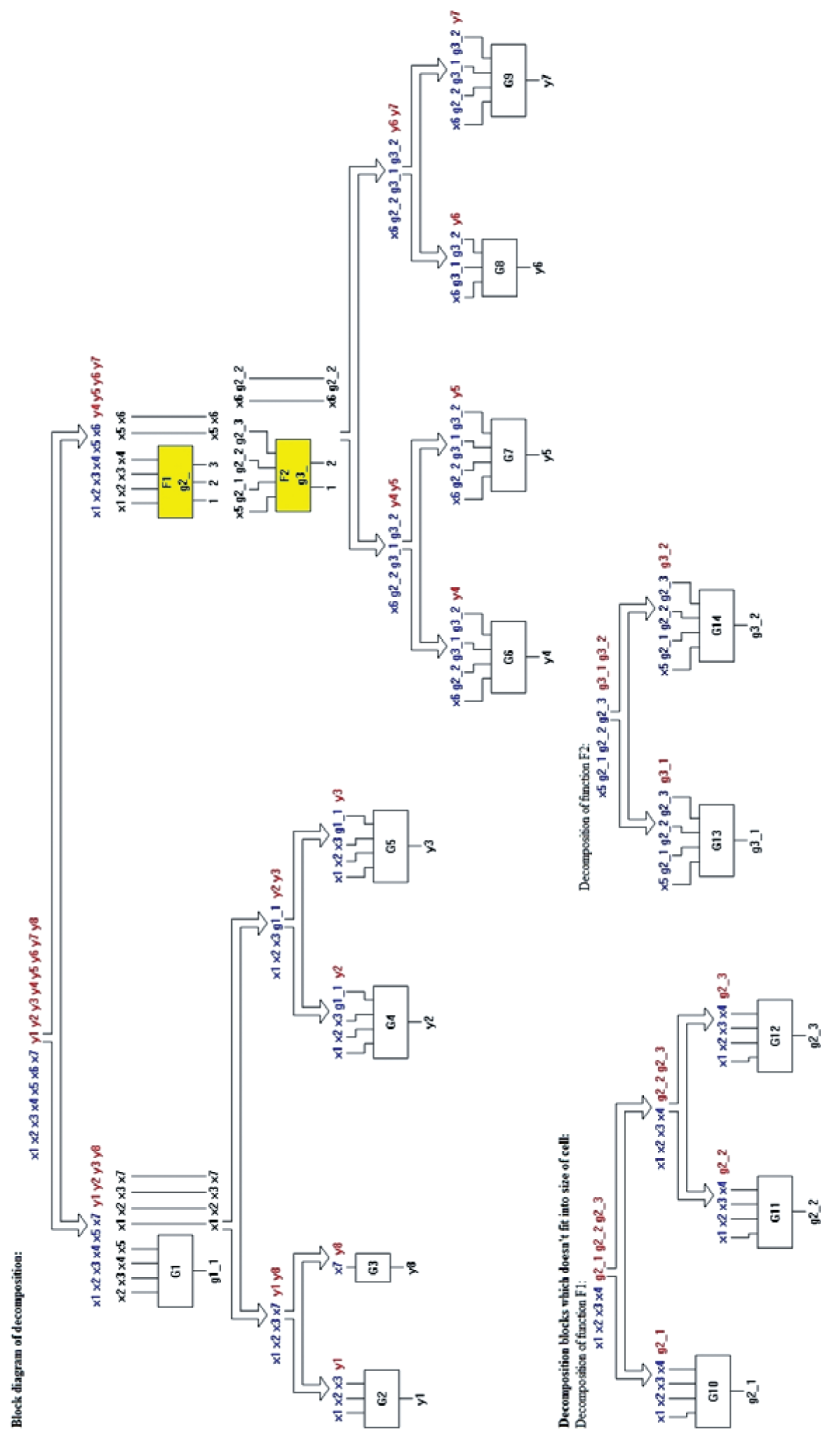


Fig. 4. Block diagram of the balanced mixed decomposition in interactive mode

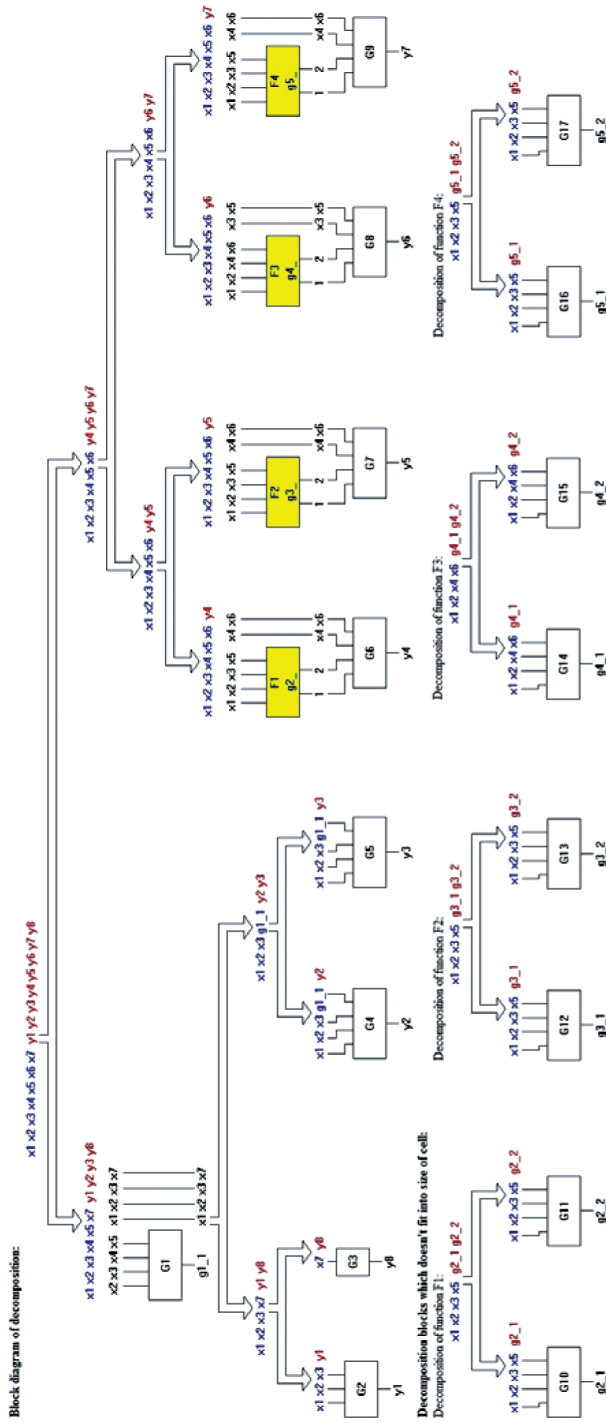


Fig. 5. Block diagram of the balanced mixed decomposition in automatic mode.

6. Conclusions

In this paper, the usability of a new design tool – *PKmin* – for balanced mixed decomposition of combinatorial circuits has been reported. The idea behind the tool is similar to that of the *DEMAIN* program, but there are both similarities and differences.

From the point of view of a designer looking for support of FPGA-based digital devices in CAD, the new tool can generate competitive designs in comparison to existing programs, especially when fully automatic decomposition of high-speed combinatorial circuit is of interest. The computer program *PKmin* and its C++ code is available at WWW site [14].

Research on the comparison of different tools devoted to automatic balanced decomposition of combinatorial logic with input instances generated at random and then processed with various settings of program parameters shall be presented in the nearest future.

References

- [1] De Micheli G., *Synthesis and optimization of digital circuits*, McGraw-Hill, 1994.
- [2] Kapralski A., Skarbek W., *Problem of searching minimum base in Boolean tables*, Podstawy sterowania, 1986, 257–265.
- [3] Kokosiński Z. Michalski T., *Synthesis of 2-level combinatorial circuits with PKmin*, Technical Transactions – Automatic Control, 1-AC/2012, 93–113 (in Polish).
- [4] Łuba T., *Design of digital circuits*, Chapter 9.4: *DEMAIN system*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2005 (in Polish).
- [5] Michalski T., *Approximation algorithms for minimization of multiple-input multiple-output combinatorial circuits*, M.S. thesis, Cracow University of Technology, Kraków 2010. (in Polish)
- [6] Mishchenko A., Chatterjee S., Brayton R., *DAG-Aware AIG Rewriting. A Fresh Look at Combinational Logic Synthesis*, Proc. 43rd ACM/IEEE Design Automation Conference, San Francisco, USA, 2006, 532–535.
- [7] Njuguna R., *A survey of FPGA benchmarks*, an electronic report available at <http://www.cse.wustl.edu/~jain/cse567-08/ftp/fpga/>
- [8] Nowicka M., Rawski M., Tomaszewicz P., Sapiecha P., *Decomposition of Boolean Circuits based on Graph Colouring Heuristics*, Materiały Krajowego Sympozjum Telekomunikacji, Tom B, Bydgoszcz 1997, 242–250.
- [9] Nowicka M., Łuba T., Rawski M., *FPGA-Based Decomposition of Boolean Functions. Algorithms and Implementation*, Proc. of the 6th International Conference on Advanced Computer Systems, Szczecin 1999, 502–509.
- [10] Nowicka M., Kraśniewski A., Zbierchowski B., *A method for designing of high speed FPGA circuits*, Materiały III Krajowej Konferencji Reprogramowalne Układy Cyfrowe RUC'2000, Szczecin 2000, 97–104 (in Polish)
- [11] Salauyou V., Klimowicz A., *Logic synthesis of digital circuits in programmable structures*, Oficyna Wydawnicza Politechniki Białostockiej, Białystok 2010 (in Polish).

- [12] Sasao T., Matsuura M., *A Method to Decompose Multiple-Output Logic Functions*, Proc. 41st Design Automation Conference, San Diego 2004, 428–433.
- [13] Sklyarov V., Skliarova I., Barkalov A., Titarenko L., *Synthesis and optimization of FPGA-based systems*, Lecture Notes in Electrical Engineering, Vol. 294, Springer 2014.
- [14] PKmin www site: <http://www.pkmin.za.pl/>
- [15] DEMAIN www site: <http://www.zpt.tele.pw.edu.pl/oprogramowanie/demain.html>

