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

FUNCTIONAL DECOMPOSITION – THE VALUE AND IMPLICATION FOR MODERN DIGITAL DESIGNING

Mariusz RAWSKI†, Tadeusz ŁUBA†, Zbigniew JACHNA‡,
Rafał RZECHOWSKI†

† Warsaw University of Technology, Institute of Telecommunications, Nowowiejska 15/19,
00-665 Warsaw, Poland, e-mail: *rawski@tele.pw.edu.pl*

‡ Military Academy of Technology, Kaliskiego 2, Warsaw, Poland,
e-mail: *Zjachna@wel.wat.waw.pl*

Abstract. *General functional decomposition has important applications in many fields of modern engineering and science. However, it is mainly perceived as a method of logic synthesis for implementation of Boolean functions into FPGA-based architectures. In this paper, an application of functional decomposition in other fields of modern engineering is presented. The experimental results demonstrate that a method of synthesis based on functional decomposition can help in implementing sequential machines using flip-flops or ROM memory. It also can be efficiently used as a method of multilevel logic synthesis for VLSI technology.*

Key Words. *Logic Synthesis, Functional Decomposition, FPGA, VLSI*

1. INTRODUCTION

Decomposition has become an important activity in the analysis and design of digital systems. It is fundamental to many fields of modern engineering and science [3], [6], [9], [16]. Functional decomposition relies on breaking down a complex system into a network of smaller and relatively independent co-operating sub-systems, in such a way that the original system's behaviour is preserved, i.e. function F is decomposed to subfunction G and H in the form described by formula F = H(A, G(B)).

Recently, new methods of logic synthesis based on functional decomposition are being developed [1], [4], [7]. One of the promising decomposition-based methods is the so-called balanced decomposition [11].

Thanks to the fact that the multi-level functional decomposition gives very good results in the logic synthesis of combinational circuits, it is viewed for the most part as a synthesis method addressed to implementation of combinational functions into FPGA-based architectures [13], [15]. However, a decomposition-based method can be used beyond this field. Since in the sequential machine synthesis after state code assignment the process of implementation is

reduced to the computation of flip-flops' excitation functions, the decomposition can be efficiently used to assist said implementation. Application of a balanced decomposition method allows the designer to decide what is the optimisation criterion – circuit area or circuit speed. Good results produced by decomposition-based logic synthesis methods in implementation of combinational circuits guarantee that this method will implement encoded sequential machines efficiently and effectively. The balanced decomposition gives the designer control over the process of excitation functions' implementation. Thanks to this, such undesirable effects as hazards can be avoided. Elimination of these effects can increase the speed of circuits.

Modern FPGA architectures contain embedded memory blocks. In many cases, designers do not need to use these resources. However, such memory blocks allow implementing sequential machines in a way that requires less logic cells than traditional, flip-flop implementation. This may be used to implement "non-vital" sequential parts of the design saving logic cell resources for more important parts. However such an implementation may require more memory than available in a circuit. To reduce memory usage in ROM-based sequential machine implementations decomposition-based methods can be successfully used [10].

Decomposition-like synthesis methods are not limited only to FPGA-based architectures [8]. The balanced functional decomposition can also be used to implement digital systems in CPLD and even VLSI technology (gate-array or standard cell). An appropriately chosen decomposition strategy allows circuit synthesis, with results comparable or even better than those achieved with classical multi-level synthesis methods based on algebraic transformation. Additionally, decomposition-based methods produce logic networks that do not require technology mapping.

In this paper, once some basic information has been introduced, the application of the decomposition to implementation of sequential machines is presented. Following that, an algorithm of implementation of function in VLSI technology, a variant of balanced functional decomposition, is presented. Subsequently, some experimental results, reached with a prototype tool that implements the balanced functional decomposition, are discussed.

The experimental results demonstrate that the decomposition is capable of constructing solutions of comparable or even better quality than the methods implemented in university or commercial systems.

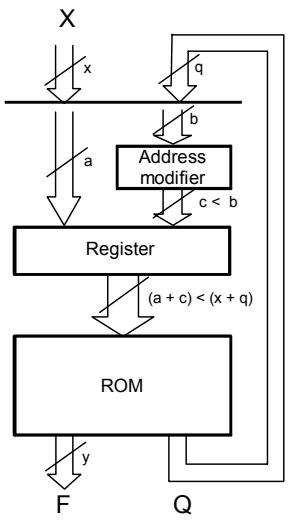

Fig. 1. Implementation of FSM using an address modifier

2. FINITE STATE MACHINE IMPLEMENTATION

The FSM can also be implemented through the use of ROM (*Read Only Memory*) [10].

The FSM defined by a given transition table can be implemented in a structure shown in Fig. 1 by means of an address modifier. The process may be considered as a decomposition of memory block into two blocks: a combinational address modifier and a smaller memory block. Appropriately chosen strategy of balanced decomposition may allow to reduce required memory size at the cost of additional logic cells for address modifier implementation. This makes possible to implement FSM that exceed available memory through using embedded memory blocks and additional programmable logic.

3. DECOMPOSITION INTO GATES

Different logic synthesis programs use different techniques to transform a synthesised network of logic gates into a form that is implementable in standard-cell or gate-array technology. These methods use either a multi-level representation of function for structural technology mapping (SIS system), i.e. a matching of the topology of synthesised network to the gate patterns defined in the technology library, or, for logical technology mapping, a calculation of the logical coverage of sub-nets of multi-level function structures by the functions defined by technology gates. However, if the multilevel structure does not respect technology constraints or is not built with a close relation to the actual target, both methods of technology mapping will disturb the optimally synthesised network, leading to losses in quality of the final product [5].

The obvious conclusion is that there is a need to use algorithms, which allow the designer to implement logic structures directly in gate level representation, with special focus on VLSI technology constraints. Attempts are being made to use evolutional algorithms or to use special properties of logic functions, such as, for example, symmetries of inputs. At the current stage, those algorithms have to be supported by the other techniques due to the maximum block size they can synthesise and the class of logic function on which they can operate [14].

Balanced functional decomposition, as used to synthesise gate-level circuits, is free of these disadvantages [16]. It permits the elimination of the whole technology mapping process and can be used to synthesise large circuits consisting of any gates available in the technology library. At the same time, functional decomposition remains a homogeneous synthesis method and does not need any supporting algorithms to perform its operations.

In the herein discussed method of functional decomposition, the technology mapping process is performed during serial decomposition, at stage of coding generation for G block's outputs. Coding of outputs of the G block allows for many variants and can be used to adjust synthesis parameters to meet technology targets and to satisfy the logical coverage of synthesised blocks by technology gates. This allows optimisation of not only the number of gates (area) of the implementation but also such implementation criteria as speed [13] and power consumption [2].

Experimental results show that such a solution gives very good results when used in stand-alone synthesis procedures, as well as in algorithms supporting other synthesis methods such as evolutional algorithms.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

The balanced decomposition was applied to implement in FPGA architectures several "real life" examples: combinational functions and combinational parts of FSMs. We used the following examples:

- bin2bcd1 – binary to BCD converter for binary values from 0 to 31,
- bin2bcd2 – binary to BCD converter for binary values from 0 to 355,
- rd88 – Sbox from Rijndael implementation,
- DESaut – combinational part of the state machine used in DES algorithm implementation,
- 5B6B – the combinational part of the 5B-6B coder,
- count4 – 4 bit counter with COUNT UP, COUNT DOWN, HOLD, CLEAR and LOAD.

Table 1

Example	***FPGA based architecture EPF10K10LC84-3***				
	DEMAIN	***MAX+Plus II***	***FPGA Express***	***Leonardo Spectrum***	***SIS***
bin2bcd1	6	41	6	6	6
bin2bcd2	39	505	225	120	136
rd88	167	332	341	245	248
DESaut	28	46	25	30	32
5B6B	41	92	100	49	51
count4	11	74	17	11	13

For the comparison following synthesis tools were used: MAX+Plus II ver. 10 Baseline, FPGA Express 3.5, Leonardo Spectrum ver. 1999.1, SIS and DEMAIN. Logic network produced by all synthesis tools were implemented in EPF10K10LC84-3, the FPGA device from FLEX family of Altera. For VLSI implementation comparison, examples from a standard benchmark set were used [17].

Application of decomposition methods in area of machine learning was demonstrated with use of examples from a standard benchmark set as well as functions from area of knowledge-based systems.

Table 1 shows the comparison of our method based on balanced decomposition as implemented in tool DEMAIN with other methods compared tools. The table shows the comparison of logic cells needed for implementation of given examples. Results of implementation in FPGA architecture show that the method based on balanced decomposition gives better results than other tools used in the comparison. The worst results are produced by the Altera MAXPlus+II.

Table 2

Example	***FF_MAX+PlusII***		***FF_DEMAIN***		***ROM***		***AM_ROM***	
	LCs/Bits	***Speed [MHz]***	***LCs/Bits***	***Speed [MHz]***	***LCs/Bits***	***Speed [MHz]***	***LCs/Bits***	***Speed [MHz]***
DESaut	46/0	41,1	28/0	61,7	8/1792	47,8	7/896	47,1
5B6B	93/0	48,7	43/0	114,9	6/448	48,0	– [3]	– [3]
count4	72/0 18/0 [1]	44,2 86,2 [1]	11/0 13/0 [2]	68,5 90,0 [2]	16/16384	– [4]	12/1024	39,5

[1] FSM described with special AHDL construction; [2] decomposition with the minimum number of logic levels, [3] decomposition not possible; [4] not enough memory bits to implement the project

Since, upon the encoding of the FSM's states, the implementation of such FSM architectures involves the technology mapping of the combinational part into target architecture, the quality of such an implementation strongly depends on combinational function implementation quality. In Table 2 a comparison of different FSM implementations are presented. Each sequential machine was described by a transition table with encoded states. We here present the number of logic cells and memory bits required (i.e. area of the circuit) and the maximal frequency of clock signal (i.e. speed of the circuit) for each method of FSM implementation. The columns falling under the FF_MAX+PlusII heading present results obtained by the Altera MAX+PlusII system in a classical flip-flop implementation of FSM. The columns under FF_DEMAIN show results of implementation of the transition table with the use of balanced decomposition. The ROM columns provide the results of ROM implementation; the columns under AM_ROM present the results of ROM implementation with use of address modifier. It can be easily noticed that the application of balanced decomposition can improve the quality of flip-flop as well as ROM implementation. Especially interesting is the implementation of the 4-bit counter. Its description with a transition table leads to strongly non-optimal implementation. On the other hand, its description when using a special Altera HDL (Hardware Description Language) construction produces very good results. However, utilization of balanced decomposition allows the designer to choose between whether area or speed is optimized. The ROM implementation of this example requires to many memory bits (the size of required memory block exceeds the available memory), thus it can not be implemented in given structure. Application of functional decomposition allows reducing the necessary size of memory, what makes implementation possible.

Results presented in Table 3 demonstrate the performance of decomposition-based method in implementing the digital circuits into two-input gates. The comparison was performed for a set of standard benchmarks [17]. The comparison of the results was carried out among the functional decomposition method, a classical, multi-level synthesis approach (represented by algorithms of the SIS system [5]), an evolutional algorithm supported by two different versions of decomposition (the Shannon decomposition and a balanced functional decomposition) [14], and decomposition supported by a function's information relationship measures [7].

Table 3

Example	***Functional decomposition algorithm***	***SIS (with modified rug script.)***	***Evolutional algorithm supported by***		***Gendec – decomposition supported by function's relationship measures.***
			Shannon decomposition + MUX	***balanced decomposition***	
xor5	4	8	–	–	4
shift	21	20	–	–	15
rd53	18	22	21+3	16	36
lion	14	13	–	–	15
alu2	61	60	512+63	71	–
adr4	27	21	–	–	–
9sym	36	188	144+31	37	–
5xp1	76	91	121+15	61	–

Balanced decomposition produces very good results in combinational function implementation in FPGA-based architectures. However, results presented in this paper show that balanced functional decomposition can be efficiently and effectively applied beyond the

implementation of combinational circuits. It can also be used in other fields of modern engineering.

ACKNOWLEDGEMENT

This paper was partially supported by project Grant KBN No. 8T11B08817.

REFERENCES

[1] Burns M., Perkowski M., Jóźwiak L.: "An Efficient Approach to Decomposition of Multi-Output Boolean Functions with Large Set of Bound Variables", *EUROMICRO'98 Conference*, Vasteras, Sweden, 1998

[2] Brzozowski I., Kos A..: "Minimisation of Power Consumption in Digital Integrated Circuits by Reduction of Switching Activity", *Proc of the Euromicro Conference*, pp.376-380. Vol. 1, Milan 1999.

[3] Brzozowski J. A., Łuba T.: "Decomposition of Boolean Functions Specified by Cubes", *Research Report CS-97-01*, University of Waterloo, Waterloo, Canada, 1997; REVISED October 1998.

[4] 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, October, 1996, pp. 1226-1236.

[5] De Micheli G.: *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, New York., 1994.

[6] Hartmanis J., Stearns R. E.: *Algebraic Structure Theory of Sequential Machines*, Prentice-Hall, 1966.

[7] Jóźwiak L., Chojnacki A.: "Functional Decomposition Based on Information Relationship Measures Extremely Effective and Efficient for Symmetric Functions", *Proc of the Euromicro Conference*, pp.150-159. Vol. 1, Milan 1999.

[8] Kravets V. N., Sakallah K. A.: "Constructive library-aware synthesis using symmetries", *Proc. Of Design, Automation and Test in Europe Conference*, 2000

[9] Łuba T.: "Multi-level logic synthesis based on decomposition". *Microprocessors and Microsystems*, 18, No. 8, pp. 429-437, 1994.

[10] Łuba T., Górski K., Wronski L.B.: "ROM-based Finite State Machines with PLA address modifiers", *EURO-DAC'92*, Hamburg, Germany, September, 1992

[11] Łuba T., Selvaraj H., Nowicka M., Kraśniewski A.: *Balanced multilevel decomposition and its applications in FPGA-based synthesis*. in G.Saucier, A.Mignotte (ed.), Logic and Architecture Synthesis, Chapman&Hall, 1995.

[12] Łuba T., Selvaraj H.: "A General Approach to Boolean Function Decomposition and its Applications in FPGA-based Synthesis". *VLSI Design, Special Issue on Decompositions in VLSI Design*, vol. 3, Nos. 3-4, pp. 289-300, 1995.

[13] Łuba T., Nowicka M., Rawski M.: "Performance-oriented Synthesis for LUT-based FPGAs", *Proc. Mixed Design of Integrated Circuits and Systems*, pp. 96-101, Lodz, 1996.

[14] Łuba T., Moraga C., Yanushkevich S., Opoka M., Shmerko V.: "Evolutionary Multilevel Network Synthesis in Given Design Style", *Proc. IEEE 30th Int. symposium on Multiple-Valued Logic*, pp.253-258, Portland,2000

[15] Rawski M., Jóźwiak L., Łuba T.: "Functional Decomposition with an Efficient Input Support Selection for Sub-functions Based on Information Relationship Measures", *Journal of Systems Architecture 47*, pp. 137-155, 2001.

[16] Rzechowski R., Jóźwiak L., Łuba T.: "Technology Driven Multilevel Logic Synthesis based on Functional Decomposition into Gates", *Proc. of the 25th EUROMICRO Conference*, pp. 368-375, Milan, Italy 1999.

[17] Collaborative Benchmarking Laboratory, Department of Computer Science at North Carolina State University, *http://www.cbl.ncsu.edu/*