

Digital Circuits Resynthesis Approach for FPGAs Based on Logic Cell with Built-In Flip-Flop

I.V. Tiunov, I.A. Lipatov, D.A. Zheleznykov

Institute for Design Problems in Microelectronics of Russian Academy of Sciences (IPPM RAS),

Zelenograd, tiunov_i@ippm.ru, lipatov_i@ippm.ru, zheleznykov_d@ippm.ru

Abstract — This paper discusses a resynthesis approach to reducing the area of the FPGA occupied by designed circuit. The approach uses the FPGA's architectural features to merge the combinational element and the sequential element into one logic cell. Merging leads to reduction in the number of elements and to a decrease in the occupied area, in the number of interconnections and, as a consequence, improvement in traceability and temporal characteristics. The proposed approach is valid for FPGAs based on logical cells with built-in flip-flop. Approbation was carried out on IWLS 2005 benchmark and other circuits.

Keywords – technology mapping, logical resynthesis, design flow, digital circuit, CAD, FPGA.

I. INTRODUCTION

Different circuit solutions, such as merging a programmable lookup table (LUT) and Flip-Flop (DFF) to one logical cell (or logical element - LE) (Fig. 1), are found in most common FPGA architectures, and using these features for optimization in most cases will lead to a significant reduction of the area occupied by the designed circuit. In addition, this leads to an improvement in the timing characteristics of the end device.

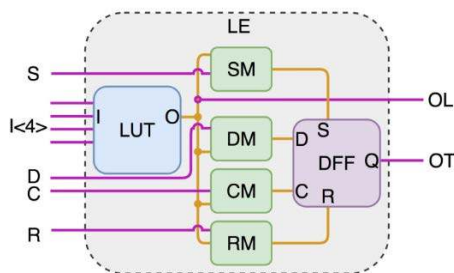


Fig. 1. The architecture of the "ideal" logic cell

It follows that the development and integration of architectural-oriented resynthesis methods implemented at the technology mapping stage into the existing design routes for FPGAs is an important task.

In the previous work [1] it was shown that the existence of such architectural features of the FPGA allows area optimization of the technologically independent synthesis. It considers only the simplest case when only one DFF is connected to the output of a logic element.

In addition to the common approach to combining the LUT table and the DFF in one logical cell, there are also other possibilities for architecture-dependent circuit

optimization. For example, in some architectures, logic cell may have two outputs: the output of the LUT and the output of the trigger. The ability to use data simultaneously from two outputs determines the potential of additional circuit optimization.

Next there is a description of the "ideal" logic cell. The presented architecture allows applying all of the resynthesis methods considered below at the stage of technology mapping.

II. DESCRIPTION OF THE "IDEAL" LOGICAL CELL

To begin with, let us describe the "ideal" (in terms of applicability of the methods considered in this paper) FPGA programmable logic cell, for which further research is carried out.

Such LE should contain: a set of gates responsible for the LUT implementation, a DFF with clock, reset and set inputs, four multiplexers on each of the trigger inputs for redirecting the signal from the programmable LUT to one of the trigger inputs.

The list of external interfaces must contain: data LUT inputs(I), data (D) and control (clock (C), reset (R) and set (S)) trigger inputs, a set of configuration inputs for programming the cell (including multiplexers), as well as the output of the trigger (OT) and the output of the lookup table (OL). This cell is schematically represented in Fig. 1.

In real FPGAs, just a few specified features of the considered architecture are used. For example, the most common is use of two outputs (OL and OT) and the possibility of redirecting the signal from the LUT to the trigger. Such approaches are used in FPGAs by Altera (Stratix II [3], Cyclone V [4], and others) and Xilinx (XC4000E [5], etc.), as well as other common architectures. More rarely, a programmable logic cell contains logic responsible for the redirection of the signal from the LUT to the control inputs of the trigger.

III. MERGING IN THE CASE OF MULTIPLE LOAD AND MERGING ON CONTROL INPUTS

A. Merging on control inputs

The cell architecture (Fig. 1) allows redirecting data from the LUT's output to one of the trigger inputs, data or the control one. It follows that merging of cells can be made not only by the data input of the trigger, but also by any of the trigger control inputs.

In the previous work [1], as mentioned earlier, the merging of cells takes place only when the LUT and the DFF are connected by the data input. Obviously, there is no such restriction for an “ideal” cell, and optimization can also be carried out in cases when the signal needs to be redirected not to the data input, but to one of the control inputs by changing the signal at the address input of the corresponding multiplexer (CM, RM or SM). It is also obvious that in order to carry out such an optimization, there should be logic in the designed circuit that controls the clock (C), reset (R) or set (S) trigger signals.

B. Merging with one of the triggers in the case of multiple load

One of the limitations of the algorithm for optimizing the technology mapping from the previous work was also the impossibility of its application for cases when more than one trigger is loaded at the output of a logic element. However, often there are situations where several elements are loaded (both DFFs and elements of the LUTs).

Let us consider one of such situations, when several elements are loaded, some of them are LUTs and some are DFFs (Fig. 2).

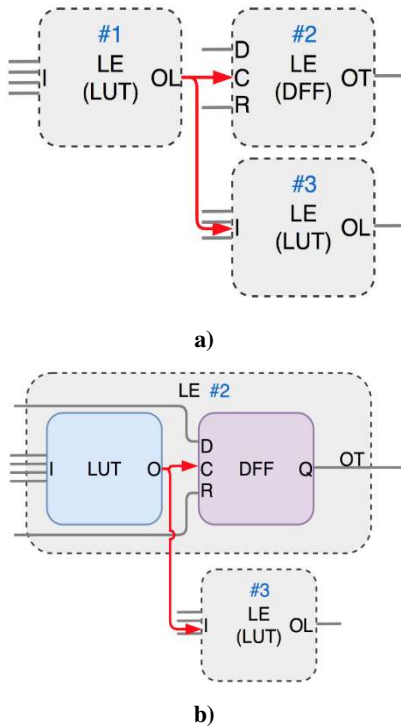


Fig. 2. Merging the LUT and the DFF on the synchronization input: a) before; b) after

Since in the considered cell data output is provided from both the LUT and the trigger, it becomes possible to merge the LUT with one of the load DFFs and to connect the remaining load elements to the output OL of the combined cell. To do this, first it is necessary to break the connection between the LUT and the DFF to be merged and with all the load elements. Next step is to merge the cells of the LUT and the selected DFF. Finally, it is

necessary to reconnect the earlier disconnected load elements to output OL of the merged cell.

Despite the fact that Fig. 2 presents the case when the LUT and the trigger are merged at the clock input, it is obvious that this method can be applied for merging by both data input and other control inputs of the trigger (reset and set).

C. Merging with multiple triggers in case of multiple load

Another variant of multiple merging can also be carried out for cases when several triggers are load (not excluding the variant of the LUT as a load). In this case it is possible to merge LUT with several DFFs at once. To do this it is needed to create a copy of the LUT for each trigger and merge as in the case of one DFF for each of the LUT-DFF pairs. An example is shown in Fig. 3.

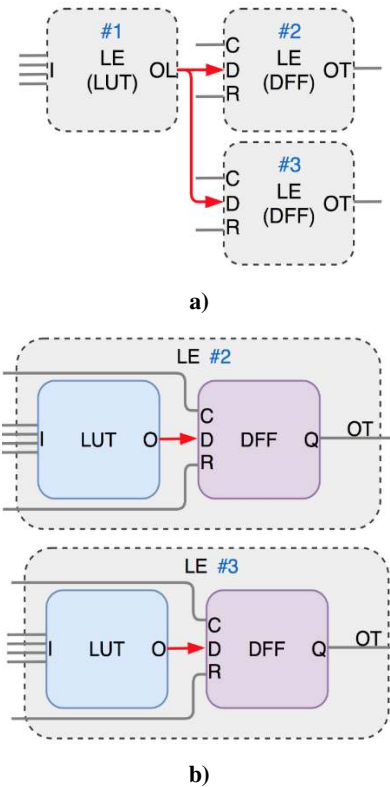


Fig. 3. Multiple merging of the LUT and the DFF: a) before; b) after

As in the case of merging with one DFF this method can be applied to all the available merging options, both by data and control inputs.

IV. INTEGRATION OF TECHNOLOGY RESYNTHESIS METHODS IN CIRCUITS DESIGN FLOW FOR FPGAS

The developed methods of architectural-oriented resynthesis at the stage of technology mapping were implemented as a cross-platform software package XCY. This package is intended for the technology mapping of circuits to the basis of logical elements of the FPGA and for their resynthesis at this design stage. The software integrated into the existing digital circuits design flow for domestic FPGAs has been developed by IPPM RAS [2].

Fig. 4 shows the stages of the CAD flow of circuits in the FPGA basis, for which their own software and algorithms were developed and implemented.

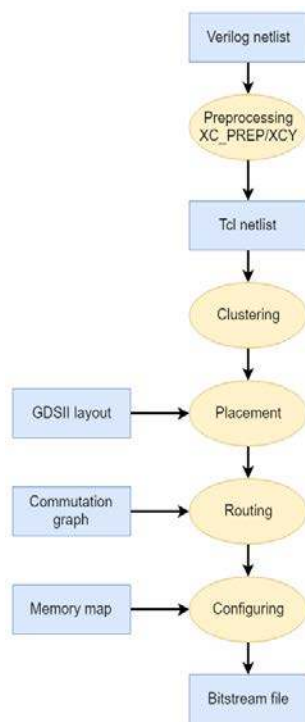


Fig. 4. The main stages of the design flow implemented by proprietary software

The XCY software package performs the technology mapping step in the design flow. Package receives synthesized description of the circuit in Verilog in terms of LUT or library elements as input data. Then it converts given data into a technologically dependent description in Tcl language. Next, the netlist in terms of the Tcl language is transmitted as input data to the XC software package which solves the problem of clustering and placement of logic elements, route, configuring and generating the bitstream file for the FPGA.

After reading the input Verilog description and creating the data structure in the memory the process of resynthesis of the circuit begins in accordance with the architectural features of the target FPGA using the described methods.

The process of resynthesis is performed in two stages by traversing all the logical elements of the designed circuit. During the first stage the program focuses only on the LUT-DFF pairs connected by the data input. When a suitable pair is found the process of elements merging starts. At the first stage preference is given to connections by data inputs since the total length of interconnections as well as the length of the critical path directly depends on this which in turn affects the performance of the designed circuit. At the second stage, checks for the presence of remaining possible connections and optimization by the following priority – reset signal (R), set (S), synchronization (C) – are performed.

Upon completion of the resynthesis process, the program can, if necessary, create additional peripheral cells (the missing cells are determined automatically based on the input description), after which a library of logical elements and a structural description of the designed Tcl scheme in terms of the XC command interface will be generated. Also the program by the user request can generate a Verilog-description in terms of optimized LUT for logical modeling of the circuit after the technology mapping stage.

As a result of the work of the XC program, the FPGA bitstream file and a number of additional files are generated: a description file for the designed scheme in SPICE, a file with input bit vectors for modeling, a graphic representation of the scheme in GDS-II format, etc.

This flow has been tested at IPPM RAS and has been successfully used in designing circuits for the FPGA of the 5510XC family at JSC MERI («АОНИИМЭ») and PJSC «Mikron».

V. TEST RESULTS

Approbation of the above methods of resynthesis on technology mapping stage was carried out both on IWLS 2005 benchmark [6] (which includes ISCAS'89, ITC'99, Faraday Benchmark, etc.), and actually designed circuits. The results of the experiments are shown in table 1.

To implement the stage of logical synthesis freely distributed software of logical synthesis – Yosys [7], [8] – was used, which, in turn, uses the Berkeley ABC software package [9] to generate a description of the circuit in terms of LUTs and DFFs.

As can be seen from the table, the result of applying the methods of multiple optimization is the reduction of the area occupied by the designed circuit up to 7%. The average value of the tested schemes is ~ 2%. The total (using the methods of this and previous work) average area gained for the considered schemes was ~ 24%. The maximum value at the same time was ~ 32%.

VI. CONCLUSION

The development of methods and algorithms for architecture-oriented resynthesis is a very important task when designing circuits in the FPGA basis. As can be seen from the obtained results, the use of such methods allows in some cases to significantly reduce the area occupied by the circuit on FPGA.

Obviously, the use of these methods will also result in a general reduction in the length of interconnections and therefore a reduction in signal propagation delays. This leads to an overall increase in performance.

The considered methods of architectural-oriented resynthesis can be applied to any FPGA having some or all of the considered architectural features.

Table 1

The results of testing the developed methods for the "ideal" cell

Benchmark	Circuit	Elements count before optimization	Elements count after single merging	Elements count after multiple merging and merging by control inputs	Improvement (%)	Total improvement (%)
ISCAS'89	s382	66	46	45	2,17	31,82
	s400	69	49	48	2,04	30,43
	s444	69	49	48	2,04	30,43
	s641	91	77	74	3,90	18,68
	s713	91	77	74	3,90	18,68
	s5378	576	457	425	7,00	26,22
	s9234	431	347	333	4,04	22,74
	s13207	1204	884	864	2,26	28,24
s38584	4457	3369	3350	0,56	24,84	
ITC'99	b14	2076	1832	1831	0,06	11,80
	b17	12544	11131	11129	0,02	11,28
Faraday	DSP	17470	14076	14000	0,54	19,86
Industrial designs	example_1	738	561	553	1,43	25,07
	example_2	792	550	539	2,00	31,94
	example_3	657	483	477	1,24	27,40
Average value:					2,21	23,96

REFERENCES

- [1] Lipatov, I. A., & Tiunov, I. V. Performance-driven technology mapping for XC5510 family FPGAs. // Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017 IEEE Conference of Russian. IEEE, 2017. P. 477-479.
- [2] Gavrilov S.V., Zheleznikov D.A., Lipatov I.A., Tiunov I.V. Marshrutproyektirovaniyadlyaotechestvennykhprogrammirusyemykhintegral'nykhskhemspetsial'nogo naznacheniya: integratsiyassushchestvuyushchimipromyshlennymisredstva miavtomatizirovannogoprojektirovaniyairesheniye problem importozameshcheniya (Design flow for domestic programmable integrated circuits for special purpose: integration with existing computer-aided design systems and solution for problems of import substitution) // Elektronnaya tekhnika. Seriya 3. Mikroelektronika. 2017. S. 5-11 (In Russian).
- [3] URL: https://www.altera.com/en_US/pdfs/literature/wp/wp-01003.pdf
- [4] URL: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-v/cv_51001.pdf
- [5] URL: <http://www.seas.upenn.edu/~ese170/handouts/FPGA.pdf>
- [6] URL: <http://iwls.org/iwls2005/benchmarks.html> (access date: 22.04.2018)
- [7] Clifford Wolf, Johann Glaser. Yosys - A Free Verilog Synthesis Suite. // Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip). 2013.
- [8] Glaser, J., & Wolf, C. Methodology and Example-Driven Interconnect Synthesis for Designing Heterogeneous Coarse-Grain Reconfigurable Architectures. // Models, Methods, and Tools for Complex Chip Design. Springer, Cham, 2014. P. 201-221.
- [9] URL: <http://www.eecs.berkeley.edu/~alanmi/abc/> (access date: 22.04.2018)