Advanced Materials Research *Online: 2013-06-13*
ISSN: 1662-8985, Vol. 705, pp 301-306
doi:10.4028/www.scientific.net/AMR.705.301

Design and Verification of a "soft" eFPGA Using New Method

Xiaodong. Xie, Ping. Li, and Wei. Li

Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China , Chengdu,China

xdxie@uestc.edu.cn, pli@uestc.edu.cn, weili@uestc.edu.cn

Keywords: Design, eFPGA, Standard cell,CAD.

Abstract. A novel design methodology for "soft" eFPGA is proposed. In comparison with the previous "soft" core design approach, a structured-description strategy is applied and the process of logic synthesis is bypassed in the new design flow. Thus, the capability of processing the bidirectional routing architectures of the mainstream eFPGAs is obtained, while the conventional "soft" core design method could only handle the eFPGA with directional routing structure. Moreover, the experiment result shows eFPGA designed with this new method is 2 times denser than that designed with the conventional method. To verify this method, a proof-of-concept eFPGA prototype is designed and also presented.

Introduction

As process geometries continue to shrink, the Non-Recurring Engineering (NRE) cost is also increasing, the ability to make post-fabrication changes is becoming more and more attractive [1,2]. A natural hardware solution to meet this growing challenge is to embed FPGAs into designs. An eFPGA(embeded FPGA) is a flexible logic fabric that consists of many programmable gates and interconnects that is able to implement any digital circuit. The designer is allowed to program these gates and the connections between them to serve different applications at the post fabrication.

However, eFPGAs are only available as "hard" IP (Intellectual Property) from vendors [3,4] with fixed fabrication process, including fixed size, shape, and I/O distribution. Due to these fixed parameters, the "hard" cores have limitations for a given application, and it makes the integration of "hard" cores a complex and daunting process. To address this issue, a "soft" or "synthesizable" eFPGA methodology [4,5] based on the ASIC design flow was proposed. The idea of "soft" eFPGA is that an eFPGA architecture is described in behavioral RTL using Verilog or VHDL alongside the rest of the user logic, then logic synthesis tools are used to map the behavioral RTL as a whole to a standard cell library, afterwards, ASIC placement and routing is performed as usual to create the IC layout. The main advantages of this approach are flexibility and ease of use. Furthermore, this approach does not restrict a design to certain foundries.

However, "soft" eFPGA incurs significant overhead in area. The studies[4] indicates that the area of eFPGA designed as soft core is much larger than that designed by the full-custom approach. In this paper, the factors to this overhead are briefly analyzed, and then a novel design method is described. The experiment of area evaluation and a proof-of-concept prototype is also presented.

Previous Design Method

Although there are several kinds of fabrics of eFPGA, for this research, island-style architecture has been selected as the reference platform because such a choice has several benefits. For example, this architecture is the basis for most of the commercial eFPGA and stand-alone FPGA; hence comparisons to other cores are more relevant. Similarly, the CAD tools for this type of architecture are more widespread and mature, hence, they can be leveraged directly without any need for new CAD tools [6]. "soft" eFPGA aimed to synthesize the behavioral RTL describing the architecture of eFPGA, and the major problem that "soft" eFPGA met is that there are several combinational logic loops in the eFPGA architectures.

Combinational logic loop is an indication of a design error, which occurs when the output of a combinational logic block is also one of its inputs. In the architectures of eFPGA, there are several potential combinational logic loops. As illustrated in Fig.1, the potential combinational loops exist because each CLB(Configurable Logic Block)output (the right pin of CLB A) is also a potential input (the bottom pin of A) to itself (via the connection block C1, C2 and the switch block M).

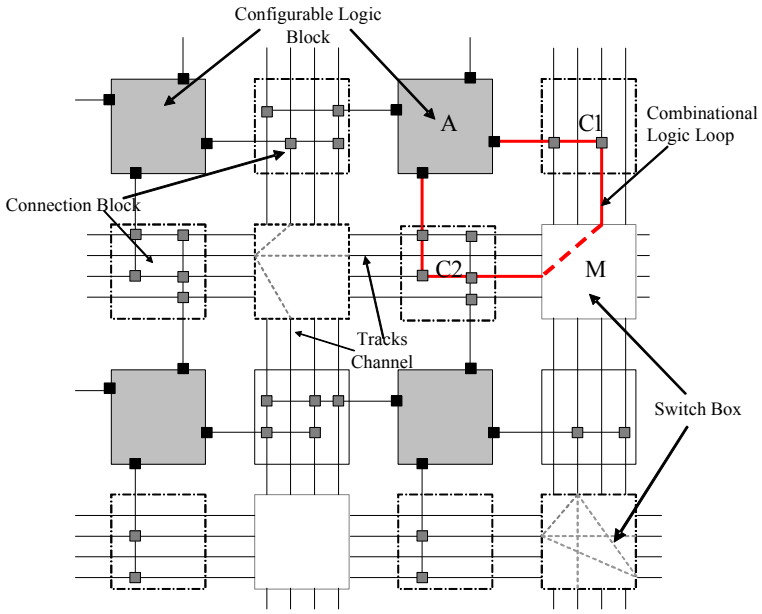


Fig.1 Potential combinational loops in eFPGA architecture

Combinational loops in eFPGA architecture cause difficulties for ASIC synthesis tools and hinder behavioral RTL from mapping to standard cells. To break the combinational loops, several directional structures were presented for "soft" eFPGA. In these structures, only unidirectional signal flows are possible, which limit the flexibility of placement and routing, consequently, extra logic and routing resources have to be added to compensate this limitation, and brings the penalty in device density.

New Design Method

As can be seen from the above, providing the eFPGA circuit net list is not generated by the synthesis tool, the bi-directional routing resources of eFPGA can be maintained and higher density could be achieved. Structured-description of a circuit has no necessity of synthesis, thus, it might be a solution to the problem of "combinational loops" in the eFPGAs' architecture. However, structured-description of the whole eFPGA seems to be a daunting task. Fortunately, there is regularity of the array in the eFPGAs and stand-alone FPGAs, GILE [7] takes advantage of this regularity and build a tileable structure called "tile" that can be replicated by grouping a CLB and the adjacent routing resources into a tile.

Based on the idea of replicated "tile", a novel design methodology for "soft" eFPGA is put forward as follows. Firstly, "tile"consisting of a CLB and CBs (Connection Block) plus a SB(Switch Block) is structured-described in Verilog or VHDL. To achieve the equivalent "synthesis" function, care is required during the design procedure. Due to the fact that the tile is of small scale, this is not a hard task. Secondly, the whole eFPGA netlist is generated by replicating and connection of "tiles". Thirdly, the excess routing tracks channels in the tiles on the boundary is cut down to save the area. Finally, the eFPGA netlist is processed alongside the rest of user logic by placement and routing tools to create physical layout which consists with the general ASIC design flow. This style of eFPGA description is more like sketching a circuit schematic, so post-description simulation is essential to guarantee the desired functionality. Part of the structured-description is given in Fig.2.

```
Architecture STRU of BASIC_module is     --Basic Tile
Component MX41D1 (……)       --4 to 1 MULTIPLEXER
Component DFPNSNRQ2(……)     -- DFF
……Architecture STRU of eFPGA is     --eFPGA ,tracks:N, size:M×M
Component BASIC_module (left_channel, bottom_channel, right_channel, top_channel: inout
std_logic_vector((N-1 downto 0)……);
Component SWITCH_block (left_channel, bottom_channel, right_channel, top_channel: inout
std_logic_vector((N-1 downto 0)…… );       Signal H_channel,V_channel: channel_array;
type channel_array is array (M-1 downto 0, M-1 downto 0) of logic_vector(N-1 downto 0);
Begin
For I in 0 to M-1 generate        -- replicate Basic Tile
  For K in 0 to M-1 generate
  BASIC_module portmap(left_channel=>V_channel(K,i), bottom_channel=>H_channel(K,i),
right_channel=>V_ channel(K+1,i), top_ channel=>H_channel(K,i+1)……);
    SWITCH_block port map(left_ channel=>H_ channel(K,i), bottom_channel=>V_channel(K,i),
    right_channel=>H_ channel(K+1,i), top_ channel=> V_channel(K,i+1)……);
  End generate;
End generate;                        --Switch Block
Architecture STRU of Switch_block is
Component TI01D1(A,OE:in std_logic;Y:out std_logic)
Begin        --TRISTATE BUFFER with active high enable
For i in 0 to N-1 generate
  TI01D1portmap(a=>left_channel (i),y=>right_track(i)……);
  TI01D1 portmap(a=> right_track(i),y=>left_track(i)……);
  TI01D1 portmap(a=> left_track(i),y=>top_track(i)……);
  TI01D1 portmap(a=> top_track(i),y=> left_track(i)……);
End generate;                    --bidrectional switch
……
Architecture STRU of BASIC_module is     --Basic Tile
Component MX41D1 (……)       --4 to 1 MULTIPLEXER
Component DFPNSNRQ2(……)     -- DFF
……
```

Fig.2 Part of the description of an eFPGA

The new design method is compared with the previous method in Fig.3. The biggest difference is that the new design method applies a structured-description strategy, and constructs the eFPGA by connecting the replicated "tiles". As the process of synthesis is bypassed in the new design flow, the "combinational loops" issue is addressed without any modefication of eFPGA architecture

.

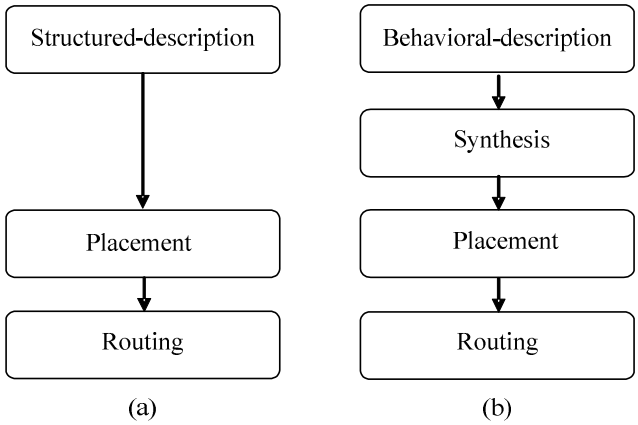


Fig.3 (a) New method design flow (b) Previous method design flow

Verification and Results

To evaluate the area of "soft" eFPGA created by this method, VPR[8] was used for placement & routing of 9 MCNC circuits to determine the characteristics of an eFPGAs tailored to implement just that circuit. Then the 9 eFPGAs were designed using the new method. To make the comparison on the same basis, the design is based on HHNEC 0.18um CMOS standard cell library. The comparison result is shown in Fig.4. eFPGA designed with this new method is 2 times denser than that designed with the previous method.

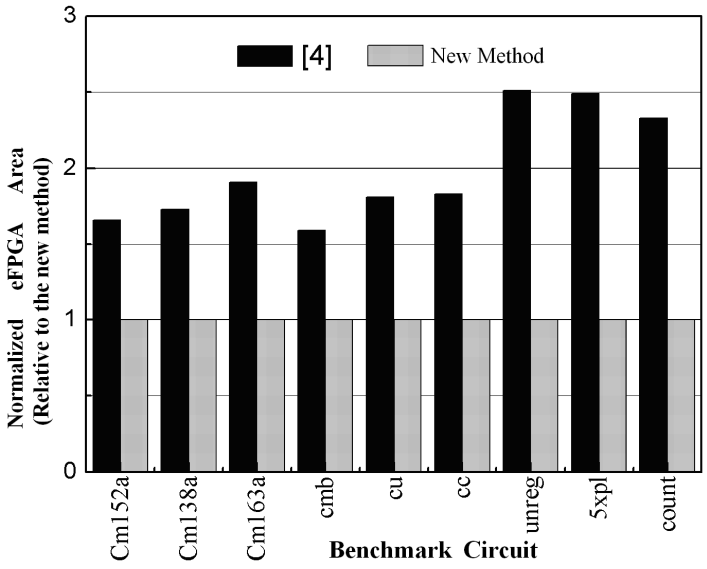


Fig.4 Area comparation of eFPGA designed by new and previous method

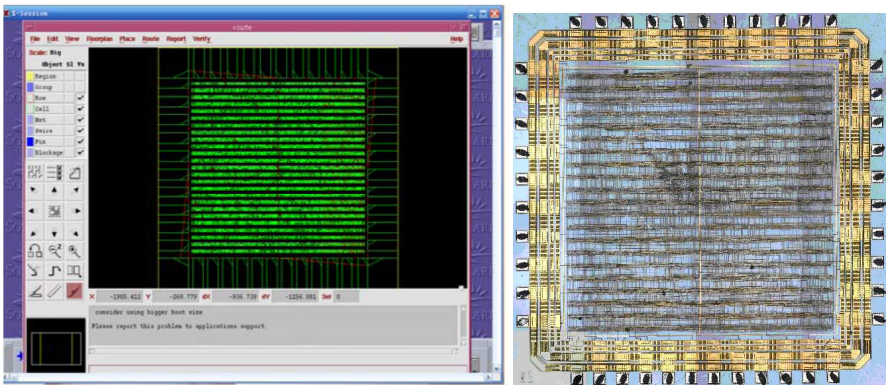
To verify the above method, an eFPPA prototype was designed. The main structure parameters are listed in Table 1. A simple I/O structure, and boundary-scan circuit which also serves for bit-stream loading were attached to the eFPGA. As standard cell library does not contain the six-transistor SRAM element, D flip-flop is employed instead, and two tri-state gates take the place of a bi-directional transistor in the switch block.

Table 1 Main structure parameters of the eFPGA prototype

Array size	4rows×4 columns
structure of CLB	4input LUT+D flip-flop
SB type	Wilton
CB type	Full connection
Tracks per routing channel	4
Length of per track	2

The eFPGA netlist was generated by the method described above with the CSMC 0.5um CMOS standard cell library, then Cadence SE (Silicon Ensemble) tools were used for placement and routing. The area of the eFPGA is 1.84 × 1.84mm^2. Fig.5 demonstrates the microphoto of the eFPGA prototype.

In order to verify the function of the eFPGA, a series of CAD toolsets have been developed as Fig.6. Quartus QUIP (the Quartus University Interface Program) from Altera [9] reads the user circuit description file (hdl), and maps it to LUTs and flip-flops, then T-vpack [10] clusters LUTs and



(a) (b)

Fig.5 (a)Layout of the eFPGA (b)Microphoto of the eFPGA

flip-flops into CLBs, and VPR is employed for placement and routing, finally, the self-developed tool "BitGen" is used for bit-flow generation.

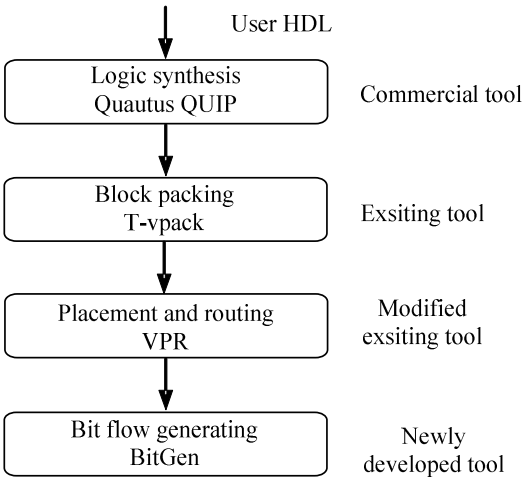


Fig.6. CAD flow and toolsets for the eFPGA

The functional test results are listed in Table 2. The eFPGA could implement the small scaled digital logic correctly which verifies the feasibility of the proposed method.

Table 2 Test result of the eFPGA

Circuit	Operating frequency	Test result
2 bits and gate		√
2 bits adder		√
3-8 decoder		√
D flip-flop	60MHz	√
3bits counter	50MHz	√
3bits FSM	50MHz	√

Conclusions

In this paper, a new design methodology for "soft" eFPGA based on structured-description strategy is presented. It generates the netlist of eFPGA architecture through copy and connection of the replicated "tiles" instead of synthesis, which keeps the bidirectional routing flexibility in the eFPGA architecture, and thus greatly cuts the area. Moreover, the feasibility of the proposed method has been verified by successfully design and implementation of an eFPGA prototype.

References

[1] B.Neumann, T.vonSydow, H.Blume, et al, Design flow for embedded FPGAs based on a flexible architecture template. Proceedings of Design, Automation and Test in Europe, 2008,pp.56-61.

[2] J.Wu,V.Aken'Ova, S.Wilton, et al, SoC Implementation Issues for Synthesizable Embedded Programmable Logic Cores. Proceedings of the Custom Integrated Circuits Conference, 2003,pp.45-48.

[3] V.Aken'Ova, R.Saleh, A "soft++" eFPGA physical design approach with case studies in 180nm and 90nm. IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures, 2006,pp.103-108.

[4] S.Wilton, N.Kafafi, J.Wu, et al, Design considerations for soft embedded programmable logic cores. IEEE Journal of Solid-State Circuits, 2005, 40(02),pp.485-496.

[5] N. Kafafi, K. Bozman, S. Wilton, Architecture and algorithms for synthesizable embedded programmable logic cores. ACM international symposium on FPGA,2003,pp.3-11.

[6] I.Kuon, A. Egier, J. Rose. Design, Layout and Verification of an FPGA using Automated Tools. ACM/SIGDA International Symposium on Field Programmable Gate Arrays - FPGA, 2005,pp. 215-226.

[7] K. Padalia, R. Fung, M. Bourgeault, et al, Automatic transistor and physical design of FPGA tiles from an architectural specification. ACM international symposium on FPGA,2003,pp.164-172.

[8] V. Betz, J. Rose, VPR: A new packing, placement and routing tool for FPGA research[C]. ACM international symposium on FPGA, 1997,pp. 213-222.

[9] Information on http://university.altera.com/research/unv-quip.html.

[10] V. Betz, J. Rose, A, Marquardt.Architecture and CAD for Deep-Submicron FPGAs.Kluwer Academic Publishers, 01(1999) 38-67.