Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 698041, 5 pages
http://dx.doi.org/10.1155/2014/698041

Research Article

Engineering Change Orders Design Using Multiple Variables Linear Programming for VLSI Design

Yu-Cheng Fan, Chih-Kang Lin, Shih-Ying Chou, Chun-Hung Wang, Shu-Hsien Wu, and Hung-Kuan Liu

Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan

Correspondence should be addressed to Yu-Cheng Fan; skystar@ntut.edu.tw

Received 7 June 2014; Accepted 18 July 2014; Published 24 August 2014

Academic Editor: Chih-Cheng Lu

An engineering change orders design using multiple variable linear programming for VLSI design is presented in this paper. This approach addresses the main issues of resource between spare cells and target cells. We adopt linear programming technique to plan and balance the spare cells and target cells to meet the new specification according to logic transformation. The proposed method solves the related problem of resource for ECO problems and provides a well solution. The scheme shows new concept to manage the spare cells to meet possible target cells for ECO research.

1. Introduction

Engineering change orders (ECO) are important technologies used for changes in integrated circuit (IC) layout and compensate for design problems. Traditionally, when chip shows errors, it often requires new photomasks for all layers. However, photomasks of deep-submicron semiconductor fabrication process are very expensive. In order to save money, ECO technology modifies only a few of the metal layers (metal-mask ECO) to reduce the cost of photomasks for all layers [1].

To perform the ECO, IC designers adopt sprinkling many unused logic gates during IC design flow. When chip is manufactured and shows design errors, IC designers modify the gate-level net-list using the presprinkling unused logic gates. At the same time, the designers track and verify the modification to check formal equivalence after ECO process. The designers must guarantee the revised design matching the revised specification.

How to achieve ECO efficiently? There are some literatures that address this problem and provide related solution. In literature [2], Tan and Jiang describe a typical metal-only ECO flow with four steps that include placement and spare cell distribution, logic difference extraction, metal-only ECO synthesis, and ECO routing [2]. Kuo et al. insert spare cells with constant insertion for engineering change and describe an iterative method to determine feasible mapping solutions for an EC problem [3]. Besides, in order to perform ECO efficiently, literature [4–9] adopt minimal change EC equations automatically. Brand proposed incremental synthesis method [4]. Huang presented a hybrid tool for automatic logic rectification [5]. Lin et al. addressed logic synthesis techniques for engineering change problems [6]. Shinsha et al. performed incremental logic synthesis through gate logic structure identification [7]. Swamy et al. achieved minimal logic resynthesis for engineering change [8]. Watanabe and Brayton presented another kind of incremental synthesis technique for engineering changes [9]. However, few researchers discuss the resource between spare cells and target cells. Therefore, in order to solve the problems, we adopt linear programming technique to plan and balance the spare cells and target cells in this paper. The proposed scheme meets the new specification according to logic transformation and overcomes the related problems of resource for ECO research.

This paper is organized as follows. In Section 2, we address typical ECO design flow. In Section 3, logic transformation is discussed. In Section 4, multiple variables linear

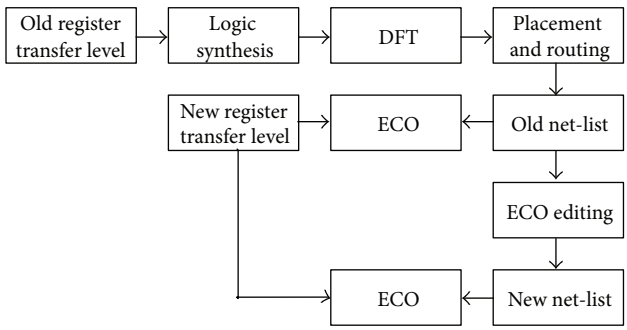


Figure 1: A typical ECO design flow.

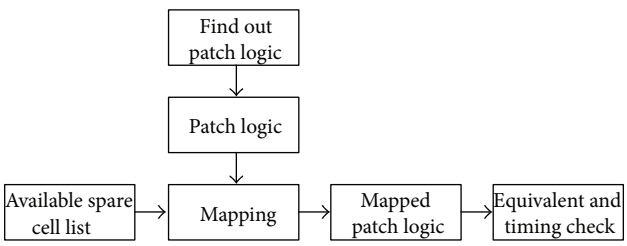


Figure 2: Two-phase ECO design flow.

programming for VLSI design is presented. In Section 5, we discuss the advantage and disadvantage of the related works. Finally, we conclude this paper in Section 6.

2. Typical ECO Design Flow

Before describing the proposed method, we address a typical manual ECO design flow in Figure 1. IC designers perform the change in register transfer level and verify fixed code matching the new specification at first. Then, old net-list is scanned to search the possible fix points. After the possible fix points are searched, IC designers modify the net-list and check the functionally equivalent between new net-list and new register transfer level [10–14].

Next, we describe two-phase ECO design flow in Figure 2. To patch the logics of the modified circuit, we prepare available spare cell list. According to logic function, the modified circuit is mapped to specified logics. After patching logic, equivalent check and timing check are performed to make sure that the new function met the new specification.

However, there are some important problems that appear during patching logic. Are there enough spare cells and types to satisfy the consumption of patch logic? How to estimate the quantity and logic types of ECO procedure? In order to solve this problem, we proposed an engineering change orders design using multiple variables linear programming for VLSI design in this paper.

3. Logic Transformation

Before discussing the engineering change orders design using multiple variables linear programming, we addressed ECO logic transformation. Figure 3 describes an ECO problem with an equation out $= (A + (BC)')'$. Figure 3(b) lists the available spare cells. According to the list, we discover the available spare cells are not enough. In order to solve the problem, we adopt another mapping solution with an equation out $= (A'BC)$ instead of the original equation in Figure 3(c). It requires one AND and one INV gate. The mapping solution in Figure 3(c) requires gates fewer than the available spare cells and is constructed with the available spare cells.

However, most of spare cells only provide basic logical functions that include AND, OR, NOT, NAND, and NOR. Half Adder (HA), Full Adder (FA), And-Or-Inverter (AOI), and Or-And-Inverter (OAI) can provide complex logical functions. We can adopt these logical cells to perform ECO function. For example, AOI22 can be implemented by two NAND and one AND cells in Figure 4. According to the existing resources of spare cells, we can resynthesize the changed function lists.

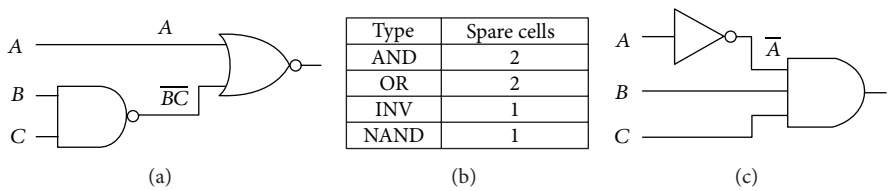
4. Multiple Variables Linear Programming for VLSI Design

Although logic transformation skill makes the ECO technology come true, a chip often does not own enough spare cells to modify the function to meet a new specification. How to allocate limited resource? We should estimate quantity of spare cells and logic transformation rule to perform optimal engineering charge orders.

In Figure 5, it describes the engineering change orders design using multiple variables linear programming for VLSI design and relation of logic transformation. "Logic A" is one kind of spare cells that can be transformed into "Logic a" or "Logic b." Similarly, "Logic B" can be transformed into "Logic a," "Logic b," or "Logic c." "Logic C" performs ECO function instead of "Logic c" or "Logic d." Besides, Logic D is transformed into "Logic c" or "Logic d." Equivalently, "Logic E" is transformed to "Logic d" or "Logic e" to achieve ECO function.

We assume X_1, X_2, X_3, X_4, and X_5 are the number of spare cells, Logic A, Logic B, Logic C, Logic D, and Logic E. Let Y_1, Y_2, Y_3, Y_4, and Y_5 be the desired number of target cells, Logic a, Logic b, Logic c, Logic d, and Logic e.

Besides, Aa is the number of spare cells (Logic A) to be transformed into Logic a and Ab is the number of spare cells (Logic A) to be transformed into Logic b. Similarly, Ba, Bb, and Bc are the number of spare cells (Logic B) to be transformed into Logic a, Logic b, and Logic c. In a similar way, Cc and Cd are the number of spare cells (Logic C) to be transformed into Logic c and Logic d, Dc and Dd are the number of spare cells (Logic D) to be transformed into Logic c and Logic d, and Ed and Ee are the number of spare cells (Logic E) to be transformed into Logic d and Logic e.



Type	Spare cells
AND	2
OR	2
INV	1
NAND	1

Figure 3: Example of an ECO problem. (a) EC equation: output = $(A + (BC)')'$. (b) Spare cells. (c) Mapping: output = $(A'BC)$.

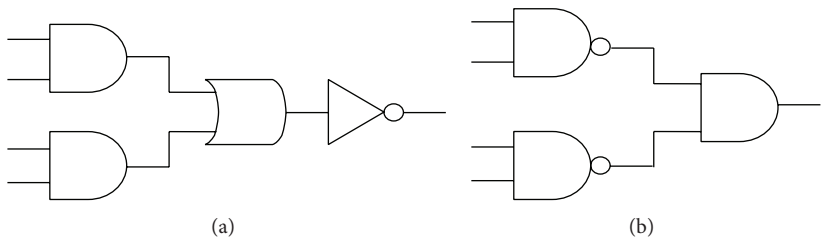


Figure 4: AOI22 can be implemented by two NAND and one AND cells.

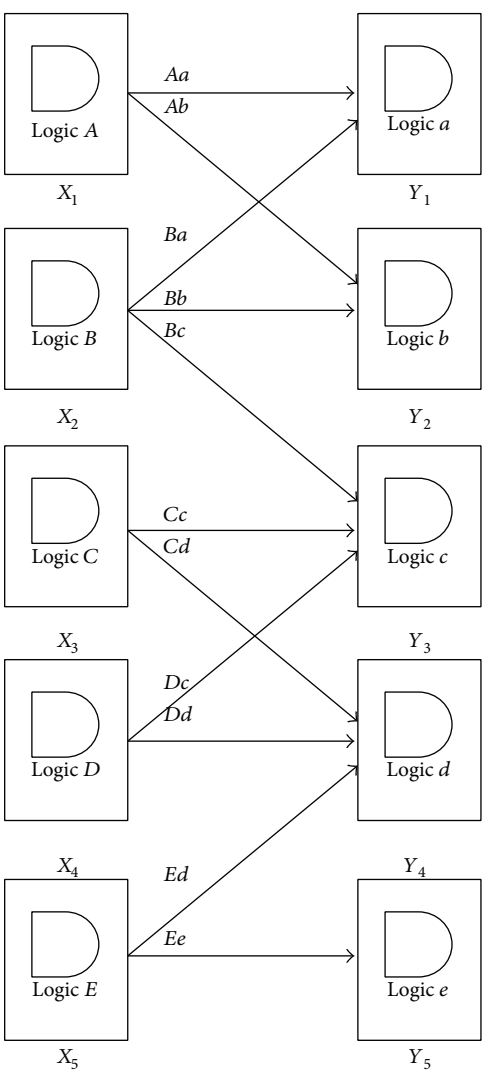


Figure 5: ECO design using multiple variables linear programming for VLSI design and relation of logic transformation.

Therefore, the restriction rule of the number of spare cells and transformed target cells in Figure 5 is written as follows:

$$\begin{aligned} X_1 &= Aa + Ab; \\ X_2 &= Ba + Bb + Bc; \\ X_3 &= Cc + Cd; \\ X_4 &= Dc + Dd; \\ X_5 &= Ed + Ee. \end{aligned} \tag{1}$$

Besides, the restriction rule of the engineering change orders design using multiple variables linear programming in Figure 5 is written as follows:

$$Aa + Ba \geqq Y_1; \tag{2}$$

$$Ab + Bb \geqq Y_2; \tag{3}$$

$$Bc + Cc + Dc \geqq Y_3; \tag{4}$$

$$Cd + Dd + Ed \geqq Y_4; \tag{5}$$

$$Ed + Ee \geqq Y_5. \tag{6}$$

However, spare cells are not often enough; designer should balance the spare cell allocation to meet all requirements of desirable cells.

We assume one case when $Bb \leqq Y_2$. In order to provide enough spare cells, we should increase the number of Ab to achieve $Ab + Bb \geqq Y_2$.

Similarly, when $Cc + Dc \leqq Y_3$, we should increase Bc number to meet $Bc + Cc + Dc \geqq Y_3$.

Therefore, we define another restriction rule of the engineering change orders design which is written as follows:

$$\begin{aligned} Aa &= X_1 - Ab; \\ Ba &= X_2 - Bb - Bc. \end{aligned} \tag{7}$$

Table 1: ECO methods comparison.

Method	Traditional ECO	Proposed method
Cell resource prediction	Constraint based	Multiple variables linear programming
Predictive precision of patching logic number	Normal precision	High precision
Balance between spare cells and target cells	Low balance	High balance
Restriction rule	Not define	Define
Resource optimization	Not define	Define
Solution boundary	Not define	Define

According to formulas (2) and (7), we can balance the number of Ab, Bb, and Bc to achieve the target number Y_1. Consider the following:

$$X_1 - Ab + X_2 - Bb - Bc \geqq Y_1. \tag{8}$$

In a similar way, we define the restriction rule of the engineering change orders design which is written as follows:

$$\begin{aligned} Bc &= X_2 - Ba - Bb; \\ Cc &= X_3 - Cd; \\ Dc &= X_4 - Dd. \end{aligned} \tag{9}$$

According to formulas (4) and (9), we can balance the number of Bc, Cc, and Dc to achieve the target number Y_3. Consider

$$X_2 - Ba - Bb + X_3 - Cd + X_4 - Dd \geqq Y_3. \tag{10}$$

We model the engineering change orders problems using multiple variables linear programming. According to the functions, we can understand the engineering change orders relation between supply and requirement. Then, designer can estimate and perform ECO using spare cell efficiently.

5. Discussion

In this Section, we discuss the advantage and disadvantage of the related works. Table 1 shows ECO method comparison. The proposed approach designs a multiple variable linear programming ECO for VLSI design. Our method can predict cell resource accurately using multiple variable linear programming techniques. Traditional ECO is not to predict it well. Besides, our scheme provides a high accurate prediction of patching logic number to balance between spare cells and target cells. It is hard for traditional ECO method to do these. Moreover, we define restriction rule, resource optimization, and solution boundary of ECO problem to increase the efficiency of the proposed ECO method and provide a well solution.

6. Conclusion

In this paper, we proposed an engineering change orders design using multiple variables linear programming for VLSI design. The paper discusses typical ECO design flow, logic transformation, and multiple variables linear programming for VLSI design. The presented scheme estimates the resource of spare cells and provides a well solution of ECO problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Science Council of Taiwan under Grant nos. NSC 101-2221-E-027-135-MY2 and 102-2622-E-027-008-CC3. The authors gratefully acknowledge the Chip Implementation Center (CIC), for supplying the technology models used in IC design.

References

[1] J. A. Roy and I. L. Markov, "ECO-system: embracing the change in placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 12, pp. 2173–2185, 2007.

[2] C. Tan and I. H. Jiang, "Recent research development in metal-only ECO," in *Proceedings of the 54th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS '11)*, pp. 1–4, August 2011.

[3] Y. M. Kuo, Y. T. Chang, S. C. Chang, and M. Marek-Sadowska, "Spare cells with constant insertion for engineering change," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 3, pp. 456–460, 2009.

[4] D. Brand, A. Drumm, S. Kundu, and P. Narain, "Incremental synthesis," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 14–18, 1994.

[5] S. Huang, K. Chen, and K. Cheng, "AutoFix: A hybrid tool for automatic logic rectification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 9, pp. 1376–1384, 1999.

[6] C. Lin, K. Chen, and M. Marek-Sadowska, "Logic synthesis for engineering change," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 2-3, pp. 282–292, 1999.

[7] T. Shinsha, T. Kubo, Y. Sakataya, J. Koshishita, and K. Ishihara, "Incremental logic synthesis through gate logic structure identification," in *Proceedings of the IEEE/ACM Conference on Design Automation*, pp. 391–397, Jun 1986.

[8] G. Swamy, S. Rajamani, C. Lennard, and R. K. Brayton, "Minimal logic re-synthesis for engineering change," in *Proceedings of*

the IEEE International Symposium on Circuits and Systems, pp. 1596–1599, 1997.

[9] Y. Watanabe and R. K. Brayton, "Incremental synthesis for engineering changes," in *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD '91)*, pp. 40–43, Cambridge, Mass, USA, October 1991.

[10] J. Wang, *Finding the Minimal Logic Difference for Functional ECO*, Taiwan Cadence Design Systems, 2012.

[11] Y. C. Fan and H. W. Tsao, "Watermarking for intellectual property protection," *IEE Electronics Letters*, vol. 39, no. 18, pp. 1316–1318, 2003.

[12] Y. Fan and H. Tsao, "Boundary scan test scheme for IP core identification via watermarking," *IEICE Transactions on Information and Systems*, vol. E88-D, no. 7, pp. 1397–1400, 2005.

[13] Y. Fan, "Testing-based watermarking techniques for intellectual -property identification in SOC design," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 3, pp. 467–479, 2008.

[14] Y. Fan and Y. Chiang, "Discrete wavelet transform on color picture interpolation of digital still camera," *VLSI Design*, vol. 2013, Article ID 738057, 9 pages, 2013.