國立交通大學

電子工程學系 電子研究所碩士班

碩 士 論 文

以最小成本選用備用標準單元

實現設計變更

Matching-Based Minimum-Cost
Spare Cell Selection for Design Changes

研 究 生：張良吉

指導教授：江蕙如 博士

中 華 民 國 九 十 八 年 七 月

以最小成本選用備用標準單元實現設計變更

Matching-Based Minimum-Cost Spare Cell Selection for Design Changes

研 究 生：張良吉　　Student: Liang-Gi Chang

指導教授：江蕙如 博士　　Advisor: Dr. Iris Hui-Ru Jiang

國 立 交 通 大 學

電子工程學系電子研究所碩士班

碩 士 論 文

A Thesis
Submitted to Department of Electronics Engineering & Institute of Electronics
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in
Electronics Engineering

July 2009

Hsinchu, Taiwan, Republic of China

中華民國 九十八年七月

以最小成本選用備用標準單元實現設計變更

研究生: 張良吉　　　　　　　　　　指導教授：江蕙如 博士

國立交通大學

電子工程學系　電子研究所

摘要

Metal-only ECO 係利用只修改金屬層的光罩來實現最後一刻之設計變更，由於 spare cells 是在 placement 階段事先埋入，因此在數量以及種類上均受到限制，使得這項工作十分具有挑戰性。本論文將 spare cell selection 這個問題簡化為 stable marriage problem，並利用名為 stable matching 的 matching-based 演算法正確地以可得的 sapre cells 實作出所要求的設計變更，同時也試著降低所費不貲之光罩成本。在實驗中經由五個工業界的實際例子驗證，對全部的例子來說，相較於廣泛被採用的手動編修的方式，或是與其自動化的版本相比，stable matching 皆能以較低之光罩成本來完成所需之設計變更。

Matching-Based Minimum-Cost Spare Cell Selection for Design Changes

Student: Liang-Gi Chang **Advisor: Dr. Iris Hui-Ru Jiang**

Institute of Electronics

Department of Electronics Engineering

National Chiao Tung University

Abstract

Metal-only ECO realizes the last-minute design changes by revising the photomasks of metal layers only. This task is challenging because the pre-injected spare cells are limited both in number and in cell types. This paper reduces the spare cell selection problem into stable marriage problem and uses a matching-based algorithm, named stable matching, that correctly implements the incremental design changes using the available space cells as well as tries to reduce the prohibitive photomask cost at the same time. The experiments are conducted on five industrial testcases. The results shows stable matching uses less photomask costs to complete design changes for all cases than the direct method that transforms the widely-used hand-editing procedure into an automatic one.

Acknowledgements

I would like to express my wholehearted gratitude to all those who gave me the possibility to complete this thesis. First, my sincerest appreciation goes to my advisor, Prof. Iris Hui-Ru Jiang, for her insightful suggestion and patient guidance. Then, I would like to be grateful to the members of my thesis committee, Dr. Evan Yu-Wen Tsai and Prof. Mango Chia-Tso Chao, for their precious suggestions and comments. Special thanks go to Huang-Bi, my best partner, for his advices and encouragements.

Last of all, I deeply show my greatest appreciation to my parents, family and friends for their invaluable support, suggestions and consideration throughout my study years.

Liang-Gi Chang

National Chiao Tung University

July 2009

Table of Content

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Engineering change order (ECO) is a modification which is made to an automatically-derived representation of a design [1]. These ECOs can be divided into two categories: functional and non-functional. For functional ECOs, modifications may be due to purposes of functionality debugging or adding new feature. On the other hand, non-functional ECOs may result from fixing timing violation or other modifications that will not affect the origin functionality. The later stage where ECO is performed, the less resources are available, and the greater challenges can be met. After the base layers (placement) are frozen during the design cycle, ECO not only can shorten design time by avoiding rebuilding the design from scratch but also can reduce fabrication time by manufacturing the base-layer photomasks in advance. After the first silicon chips are produced, ECO can save the prohibitive photomask cost by reusing the base-layer part in the next tape-out. Modifying only a few photomasks of metal layers to realize design changes is referred to as metal-only ECO. To facilitate metal-only ECO, a design is sprinkled with unused (spare) cells at placement, and their inputs are tied to either logic high or low to prevent floating signal. ECO is then performed by rewiring the inputs and outputs of spare cells.

Good metal-only ECO relies on the following three techniques:

1. Sufficient and evenly sprinkled spare cells can accommodate design changes at all possible locations [2][3][4].
2. A good incremental eco-router can handle tremendous obstacles and design rules and complete routing with minimum changes [2][3][4][5].
3. A powerful ECO synthesizer can fulfill the design changes on functionality and/or timing by including the physical information into logic synthesis wisely and modeling the impact on the photomask cost when the selected spare cells deviate from the ideal locations [2][3].

1.2 Previous Work

For timing ECO, [6] proposed a technology-remapping technique based on dynamic programming. [7] fixed the input slew, output loading, and delay violations by inserting buffers (spare cells). However, these methods might not easily be extended for functional ECO. On the other hand, for functional ECO, [8] remapped the expected cells with available spare ones whose inputs could be inserted with constant values (logic high/low). However, this method did not consider the issues of non-tree type cells (e.g., multiplexors, full-adders), freeing up unused cells, and minimizing the photomask cost. It can hardly handle timing ECO.

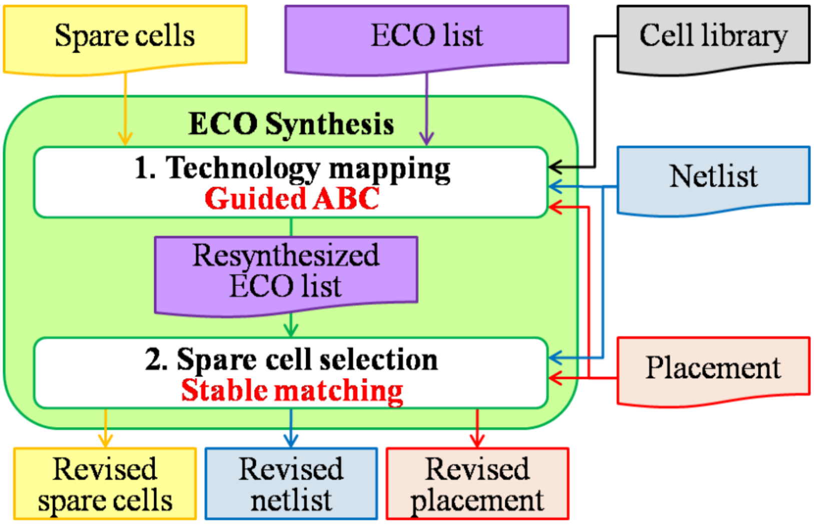


Figure 1.1: The overview of ECOS

1.3 Our Contribution

To overcome the aforementioned drawbacks, this paper proposes a matching-based ECO synthesizer, named ECOS, to complete the functional changes with the minimum photomask cost. As shown in Figure 1.1, given the list of functional changes, the original netlist and placement, and the available spare cells, ECOS first resynthesizes the given ECO list using affordable spare cell types with geometry proximity consideration. Then, each instance in the resynthesized list is repleced by an adequate spare cell based on stable matching, as well as the related nets are reconnected. Moreover, the induced unobservable cells can be freed up for later ECO runs. The objective of spare cell selection is to minimize the photomask cost of metal layers. Without loss of generality, this cost is modeled by the summation of the half-perimeter bounding box (HPBB) of each net in the revised design. This cost model benefits short interconnect delay, thus readily extending to timing ECO. Afterwards, formal equivalence checking can be performed to verify whether the revised design matches the revised functionality. ECOS has the following distinguished features:

1. It handles non-tree type spare cells and ECO functions.
2. It considers constant value insertion for spare cells.
3. It recycles freed-up cells for subsequent ECO runs.
4. It integrates physical information into resynthesis.
5. It solves the competition among spare cells.
6. It minimizes the photomask cost (also benefits timing).

7. It can readily extend to timing ECO.
8. It easily collaborates with existing synthesizers.

To demonstrate the effectiveness, we automate the common hand-editing ECO flow and the conduct the experiments on five industrial testcases. The results show that ECOS is promising.

1.4 Organization

The remainder of this thesis is organized as follows. Chapter 2 introduces the common ECO synthesis flow and formulates our problem, Chapter 3 characterizes our methodology, Chapter 4 presents our experimental results, and Chapter 5 sums up this thesis.

Chapter 2

Preliminaries and Problem Formulation

In this Chapter, we will introduce

2.1 Common ECO Synthesis Flow

Metal-only ECO is commonly performed by hand-editing the netlist [1]. However, this ad hoc method is time-consuming and resource intensive because the design related files have to be searched and edited many times during the whole process. Figure 2.1(a) shows an example design with two inputs, two outputs, and four logic cells. Spare cells include two AND and one NOT cells. The placement is also illustrated. For simplicity, the area of each cell in this example is 0, and all pins are located at the same point.

2.2 HPBB Cost Model

The photomask cost of metal layers depends on how complicated the whole routing could be. Hence, the photomask cost of each net is modeled by the half-perimeter bounding box (HPBB) over its related pins; the total cost of a design is the sum of HPBBs over all nets. For the design in Figure 2.1(a), we have its total cost:

HPBB(net1) + HPBB(net2) + HPBB(net3) + HPBB(in1) + HPBB(in2) + HPBB(out1) = 6,000 + 1,000 + 3,000 + 5,000 + 5,000 + 2,000 = 22,000.

Assume the functional ECO is to replace the functionality of cell U3 by AND. We have two options: spare cells SPARE1 and SPARE2. We prefer SPARE2 because of the better proximity to cells U1, U2, U4 and output out2. Moreover, in the revised design, the output of cell U3 becomes unused. Hence, the inputs of U3 can be connected to constant values instead. This kind of freed-up cells can be gathered and reused at subsequent ECO runs. The revised design can be improved as Figure 2.1(b). The total cost becomes 20,000.

Although not shown here, a spare cell can be used to implement more than one function, e.g., a two-input NAND cell can be a NOT cell by inserting a logic high to one input. The constant insertion can maximize the capability of each spare cell. Please note that the cell types of spare cells in most of cases do not directly match the ECO functionality. We need to translate the ECO functionality into pieces, and realize each piece by a spare cell. When the size of the ECO list is large and the resource of spare cells is limited, the ad hoc method would be time-consuming and may fail due to the competition among ECOs.

2.3 Problem Formulation

This paper solves the following metal-only ECO problem.

Minimum-Cost Spare Cell Selection for Functional ECO:

Given the original netlist, placement, a set of spare cells, and a list of functional changes, complete the ECO list using the available spare cells, create the revised netlist with the minimum cost, and generate the revised set of spare cells.

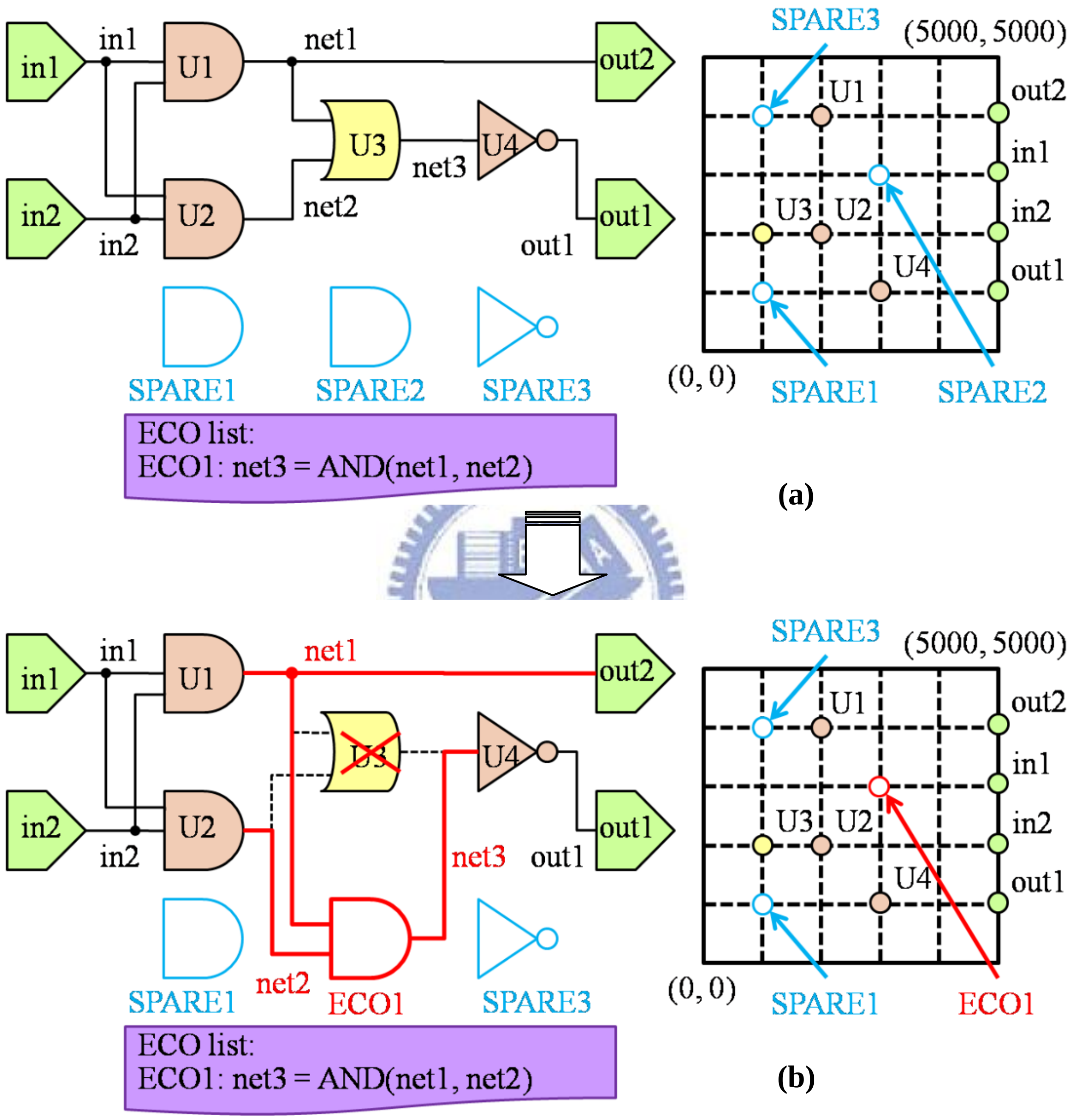


Figure 2.1: (a)The original design. (b) The revised design

Chapter 3

Matching-Based ECO Synthesis

We have developed a matching-based ECO synthesizer, named ECOS, to solve the problem formulated in Section 2.2. Figure 1.1 details ECOS' inputs/outputs and summarizes its two steps.

3.1 Technology Mapping: Guided ABC

The first step of ECOS performs technology mapping to resynthesize the given ECO list using the available spare cell types. After this step, a resynthesized ECO list is produced. We build our synthesizer based on the well-established environment, ABC [9]. Basically, ABC performs optimal-delay DAG-based technology mapping, while guided ABC uses the physical information of spare cells to direct technology mapping. The cell library is modified for each ECO. Each spare cell corresponds to one library cell; its cell area reflects the cost when it is selected for this ECO, while its cell delay is set to zero. Doing so can trigger ABC to perform area-recovery since each possible mapping has the same delay.

As mentioned in Section 2.1, the cells that become unobservable after ECO are freed up and included in the revised set of spare cells for later ECO runs. They cannot affect the costs of their input and output nets after being recycled. In Figure 2.1(a), cell U3 does not affect the costs of net1, net2, and net3 after ECO1 is applied. After cell U3 in Figure 2.1(b) is freed up, the HPBBs of touched nets will be:

HPBB(net1) = 3,000; HPBB(net2) = 0; HPBB(net3) = 0.

Moreover, we can compute the HPBB of each ECO by finding the bounding box over all active pins in its related nets, e.g., we have

HPBB(ECO1) = HPBB(net1, net2, net3) = HPBB(U1, out2, U2, U4) = 6,000.

The value can be viewed as the lower bound of the total cost induced by an ECO. The cost of selecting a spare cell for a given ECO can be computed accordingly. For ECO1, the costs of SPARE1, SPARE2, and SPARE3 are 7,000, 6,000, and 7,000, respectively. Hence, the cell library for ECO1 contains one SPARE1 cell of function/area/delay AND/7,000/0, one SPARE2 cell of function/area/delay AND/6,000/0, and one SPARE3 cell of function/area/delay NOT/7,000/0. Based on the modified cell library, guided ABC can generate the best choice for each ECO. For example, for ECO1, SPARE1, and SPARE2 have the same delay (zero), so the cell of smaller area is selected, i.e., ECO1 is resynthesized as a SPARE2 cell. Then, the resynthesized ECO list contains only the cell types of available spare cells. Sometimes, an ECO in the original ECO list may be converted into several cells. Moreover, DAG-based technology mapping cannot handle non-tree type spare cells and ECO functions. We resort this problem to ROBDDs. If the spare cell types are a mixture of only multiplexors (MUX) and inverters (NOT), the ECO list will be transformed to ROBDDs first; these ROBDDs are then simplified and converted to MUX/NOT cells. In addition, constant value insertion can naturally be implemented in technology mapping, thus maximizing the capability of spare cells. It can be seen that the guidance made for ABC indeed can also easily be built in other existing logic synthesizers provided by EDA vendors.

3.2 Spare Cell Selection: Stable Matching

Although guided ABC considers physical information of spare cells, it cannot handle the competition among ECOs. If a spare cell is the best choice of several ECOs, guided ABC duplicates it for these ECOs. To solve this problem, we do not select spare cells directly in guided ABC, but defer the decision making to step2. With the global view of costs, deferred decision making at step 2 may lead to good results.

3.2.1 The Stable Marriage Problem

Stable Marriage:

Given a set of *n* men and *m* women, marry them off in pairs after each man has ranked the women in order of preference and each women has done likewise such that no pair of man and woman who would both rather have each other than their current partners. If there are no such pairs, all the marriages are stable.

We reduce spare cell selection to the stable marriage problem, handling the competition among candidates in nature. Gale and Shapley proposed a stable matching algorithm listed in Figure 3, which is male-optimal [10].

3.2.2 The Reduction

Due to male-optimality, each ECO in the resynthesized list is modeled as a man, while each spare cell is modeled as woman. The preference is defined by the added cost resulting from assigning a spare cell to an ECO. The less added cost, the more preference. The added cost contains the difference made on the existing nets and the induced cost on the newly created nets. (Please note that the added cost is different from the cost used in guided ABC.) If there are no newly created nets among the ECOs in the resynthesized list, the preference order can be determined directly, e.g., ECO1 in Figure 2.1(a) has the following preferences:

pref(ECO1, SPARE1) = cost(ECO1, SPARE1) = 1,000;

pref(ECO1, SPARE2) = cost(ECO1, SPARE2) = 0.

Thus, ECO1 prefers SPARE2 and proposes to SPARE2. SPARE2 then accepts, and the stable matching is found. If no ECOs in the resynthesized list are connected each other, the stable matching algorithm leads to good results or all ECOs. However, when there are internal connections among some ECOs, the spare cell selection for ECO would affect each other. This case may occur when an ECO is resynthesized into several ECOs; some newly created nets connect amon ECO cells only. Because the designated spare cells for ECOs have not been decided yet, the cost cannot be determined. To estimate the induced cost on the newly created nets, the reference point of each ECO is introduced to represent a good location for spare cell selection. It is initially set as the average x- and y-coordinates over all pins on its related nets. For example, the reference point of ECO1 in Figure 2.1(a) is at:

x(ECO1) = 3,000; y(ECO1) = 2,750.

With setting the reference points, we can compute the induced cost on newly created nets and then rank the preference between ECOs and spare cells. For example, Figure

3.2(a) depicts the reference points and bounding boxes of ECO2 and ECO3. Assume a newly created net connecting ECO2 and ECO3. The preference of ECO2 proposing to SPARE1 consists of the added cost between SPARE1 and the existing nets of ECO2 and the estimated cost of the net between ECO2 and ECO3.

pref(ECO2, SPARE1) = cost(ECO2, SPARE1) + HPBB(ECO3, SPARE1).

```
StableMatching(M, W)
Initialize all m ∈ M and w ∈ W to free
while ∃ free man m with a woman w to propose to do
    w = m's highest ranked such woman
    if w is free then
        (m, w) become engaged
    else // some pair (m', w) already exists
        if w prefers m to m' then
            (m, w) become engaged
            m' becomes free
```

Figure 3.1: The stable matching algorithm [10].

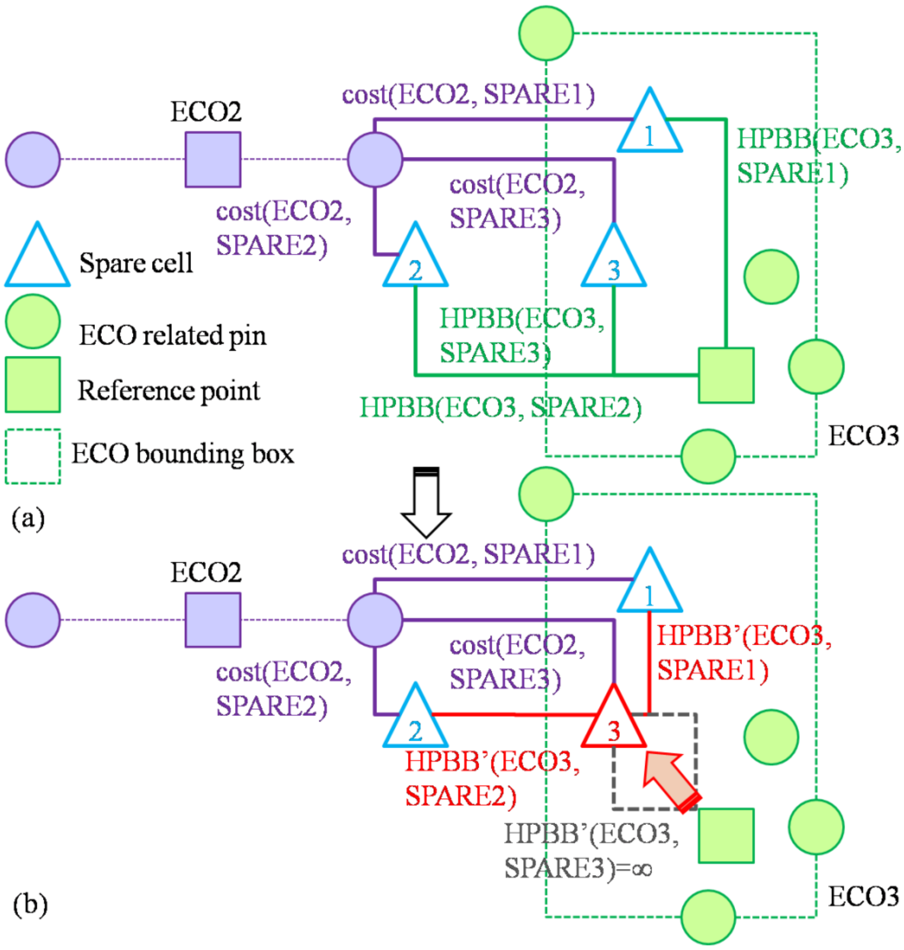


Figure 3.2: The reference points and bounding boxes: (a) Before engagement. (b) After engagement.

As the stable matching algorithm progresses, once an ECO is engaged to a spare cell, its reference point is updated to the real location of the engaged spare cell. As shown in Figure 3.2(b), after ECO3 proposes to SPARE3 and gets accepted, the estimated cost of the net between ECO2 and ECO3 is updated as follows.

HPBB'(ECO3, SPARE1) = HPBB(SPARE3, SPARE1).

HPBB'(ECO3, SPARE2) = HPBB(SPARE3, SPARE2).

HPBB'(ECO3, SPARE3) = ∞

The estimated cost between ECO3 and SPARE3 is set to a large value rather than 0 to prevent ECO2 from proposing to SPARE3. Doing so can guarantee that the method is stable and always has a solution. Spare cell selection follows the stable matching algorithm, except line 5 and 8 in Figure 3.1. The reference point of an ECO is updated when it is engaged. The preference ranking related to this ECO is updated accordingly; this update would not affect the processed proposals to maintain stability.

Chapter 4

Experimental Results

4.1 Benchmark Applications

We applied ECOS on five industrial testcases. The first three use a general cell library with basic and complex logic cells, while the last two use a cell library with basic and complex logic cells, while the last two use a cell library with only multiplexors and inverters. Table 4.1 lists the number of pins, the number of cells, the number of nets, and the number of spare cells of testcases.

Table 4.1: Statistics on testcases

Statistics					
Case	#Pins	#Cells	#Nets	#Spare	#ECO
testcase1	483	28,591	28,705	350	7
testcase2	483	28,591	28,705	2,300	49
testcase3	483	28,591	28,705	2,300	94
testcase4	33	198	181	40	3
testcase5	30	938	850	100	4

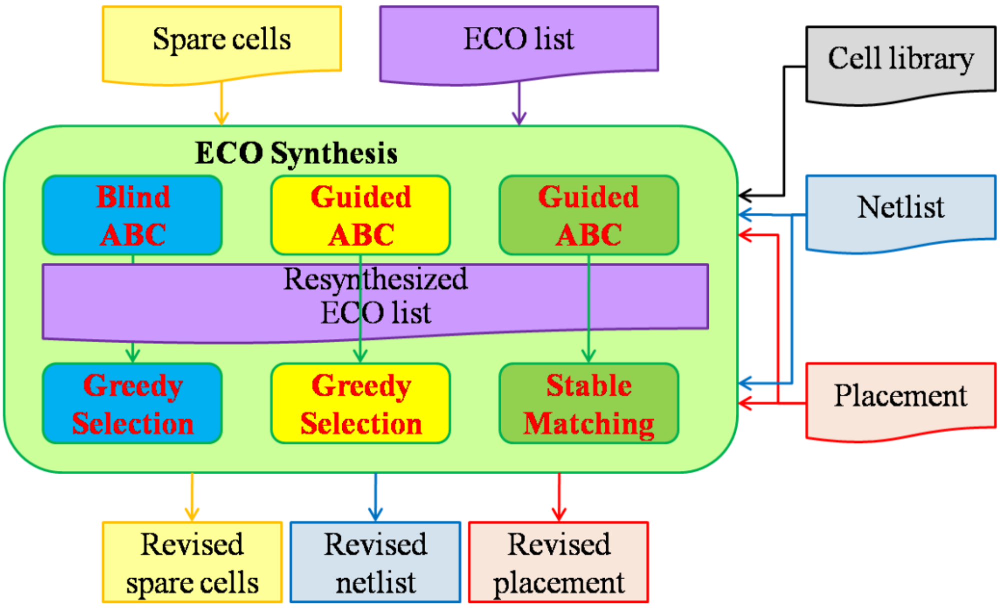


Figure 4.1: Experimental Setup

4.2 Experimental Setup

The netlist and placement of the original design are described in DEF format, while the ECO list is specified in VERILOG format. As showed in figure 4.1, we build three flows, Blind ABC + Greedy Selection, Guided ABC + Greedy Selection, and Guided ABC + Stable Matching.

4.2.1 Direct Method (Blind ABC + Greedy Selection)

We automated the common ECO method as the direct method. It searches for a spare cell with the required functionality within the bounding box of each ECO. If none, blind ABC resynthesizes each ECO with the available spare cell types (the area & delay of each spare cell are 0). Then, each ECO in the resynthesized list is directly mapped to one spare cell of the same type inside its bounding box. If failed, it is alternatively mapped to someone else outside the bounding box. Hence, direct method is like handcrafting made by engineers.

4.2.2 Greedy Method (Guided ABC + Greedy Selection)

The second part of our experiment is aimed to show how stable matching affects result; meanwhile, it reveals how much improvement that Guided ABC did. We will discuss soon later.

4.2.3 ECOS (Guided ABC + Stable Matching)

In the last experiment, we combine guided ABC and stable matching algorithm such that we can use the result to compare with section 4.2.1 to know overall improvement and to compare with section 4.2.2 to obtain the effect which stable matching caused.

Table 4.2: Total cost and HPBB of ECO comparison.

	Total cost				**HPBB of ECO**			
	Before ECO A	Direct B	Greedy C	**ECOS D**	Before ECO A	Direct B	Greedy C	**ECOS D**
testcase1	4,049,536,290	4,062,343,750	4,051,513,850	**4,051,305,350**	8,124,160	24,336,840	12,914,040	**12,621,060**
testcase2	4,142,631,960	4,414,703,560	4,354,908,860	**4,336,177,520**	78,648,240	301,202,720	198,518,940	**194,842,960**
testcase3	4,142,631,960	4,640,619,480	4,570,512,240	**4,553,033,540**	166,895,860	572,380,280	471,953,680	**466,332,680**
testcase4	2,310,270	2,652,430	2,323,150	**2,323,150**	114,800	501,200	164,080	**164,080**
testcase5	12,945,900	14,098,100	12,989,860	**12,989,860**	413,250	1,677,700	567,220	**567,220**
Imp1 (%)	7.66%	-	4.68%	**4.84%**	72.76%	-	46.41%	**47.09%**
Imp2 (%)	100.00%	-	62.61%	**65.01%**	100.00%	-	63.29%	**64.26%**

Imp1 = (B-X)/B, X = A, C or D.

Imp2 = (B-X)/(B-A), X = A, C or D.

Table 4.3: Statistics on ECO

	#ECO				
Case	#Req	#FR	Original	Blind ABC	Guided ABC
testcase1	7	7	7	16	11
testcase2	49	51	49	295	153
testcase3	94	121	94	313	162
testcase4	3	3	3	5	4
testcase5	4	4	4	6	5

Table 4.4: Statistics on CPU time

CPU time (sec)								
Step	Technology mapping		Spare cell selection			Total		
Method	Blind ABC A	Guided ABC B	Direct mapping C	Greedy selection D	Stable matching E	Direct A+C	Greedy B+D	ECOS B+E
testcase1	0.14	0.36	0.21	0.09	0.15	0.35	0.45	0.51
testcase2	0.88	3.66	0.32	0.46	1.75	1.20	4.12	5.41
testcase3	1.35	7.61	0.28	0.86	1.77	1.63	8.47	9.38
testcase4	0.10	0.14	<0.01	<0.01	<0.01	0.10	0.14	0.14
testcase5	0.14	0.17	0.01	<0.01	<0.01	0.15	0.17	0.17

4.3 Discussion

The results are listed in Table 4.2. Table 4.3 lists the number of the given ECO list, the number of freed-up cells and the sizes of resynthesized ECO lists for blind & guided ABC. The number of freed-up cells could be greater than the size of the ECO list when the freed up cell has multiple outputs. We can see guided ABC generates a much fewer ECOs in the resynthesized list than blind ABC. Generally, the smaller resynthesized ECO list may result in the smaller overlapped bounding boxes and then lead to the lower cost. Table 4.2 compares the total cost, the ECO related HPBB before & after ECO. The original values are listed here for reference. **Imp1** means the cost normalized to the direct method, and **Imp2** means the impact normalized to the direct method. The direct method incurs average 7.66% degradation on the total cost. Considering the HPBB of ECO (which can be viewed as the lower bound of the cost induced by ECO), ECOS on average outperforms the direct method by 4.84% on the total cost and 47.09% on the HPBB. Meanwhile, ECOS always produces the better results.

Comparing with greedy method and ECOS, we can observe ECOS still produces the better results all the times. As mentioned in section 4.2.2, it is because of stable matching. Without the help from stable matching, this problem becomes order-dependent. For instance, we have 2 ECOs and 3 spare cells in figure 4.2. The ECO lists are:

ECO1: net3 = AND (net1, net2);

ECO2: net2 = AND (in1, in2).

If we do ECO1 first, ECO1 will select SPARE2 since SPARE2 will not increase his HPBB. Then ECO2 is forced to select SPARE1. Therefore, the total cost is summation of Cost(ECO1) and Cost(ECO2), which is 2,000. In Contrast, if we do ECO2 first, we can obtain a total cost that is 1,000. Obviously, if more than one ECO prefer the same spare cell, there are always competitions needed to be solved. Figure 4.3 shows the possible competition condition. The freed up cells and the spare cells are centralized in the left part of the design, especially the left-top corner. As a result, if we handle ECOs once by a time, we may loss the opportunity to find the optimal solution. That is why ECOS performs better than greedy method.

For testcase4 shown in Figure 4.4, guided ABC generates 4 instead of 5 ECOs in the resynthesized list. The bounding box and all related pins of each ECO are highlighted. The greedy method and ECOS have the same results. It can be seen that the results are much better than what the direct method gets, e.g., ECO3 is implemented by two spare cells, ECO3_0 and ECO3_1, in geometry proximity.

Moreover, Table 4.4 summarizes CPU times. Compared with the timing-consuming manual method, the automatic methods are efficient.

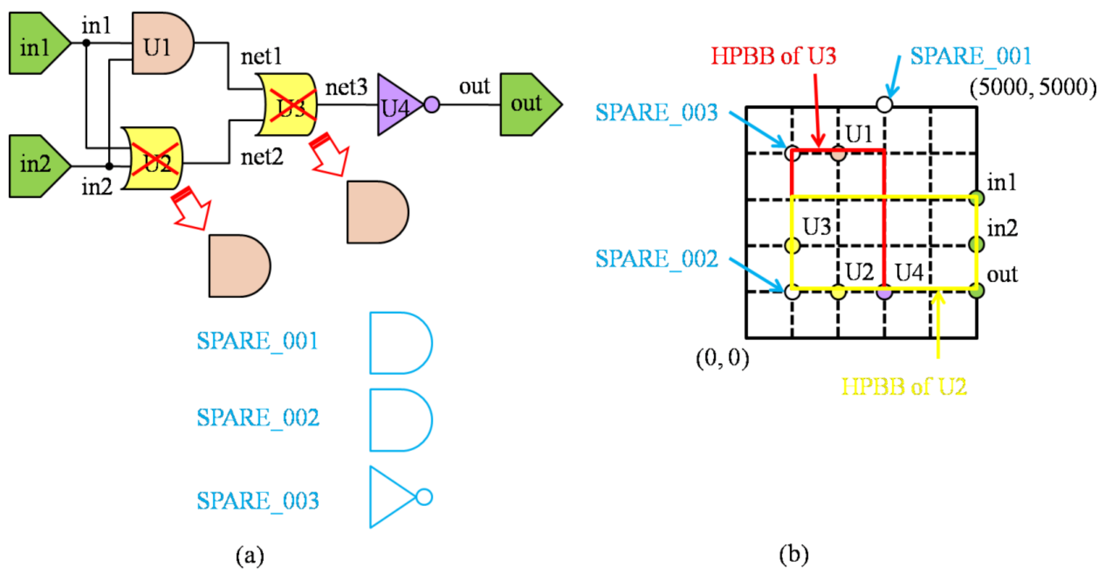


Figure 4.2: (a) Netlist and ECO list (b) Placement.

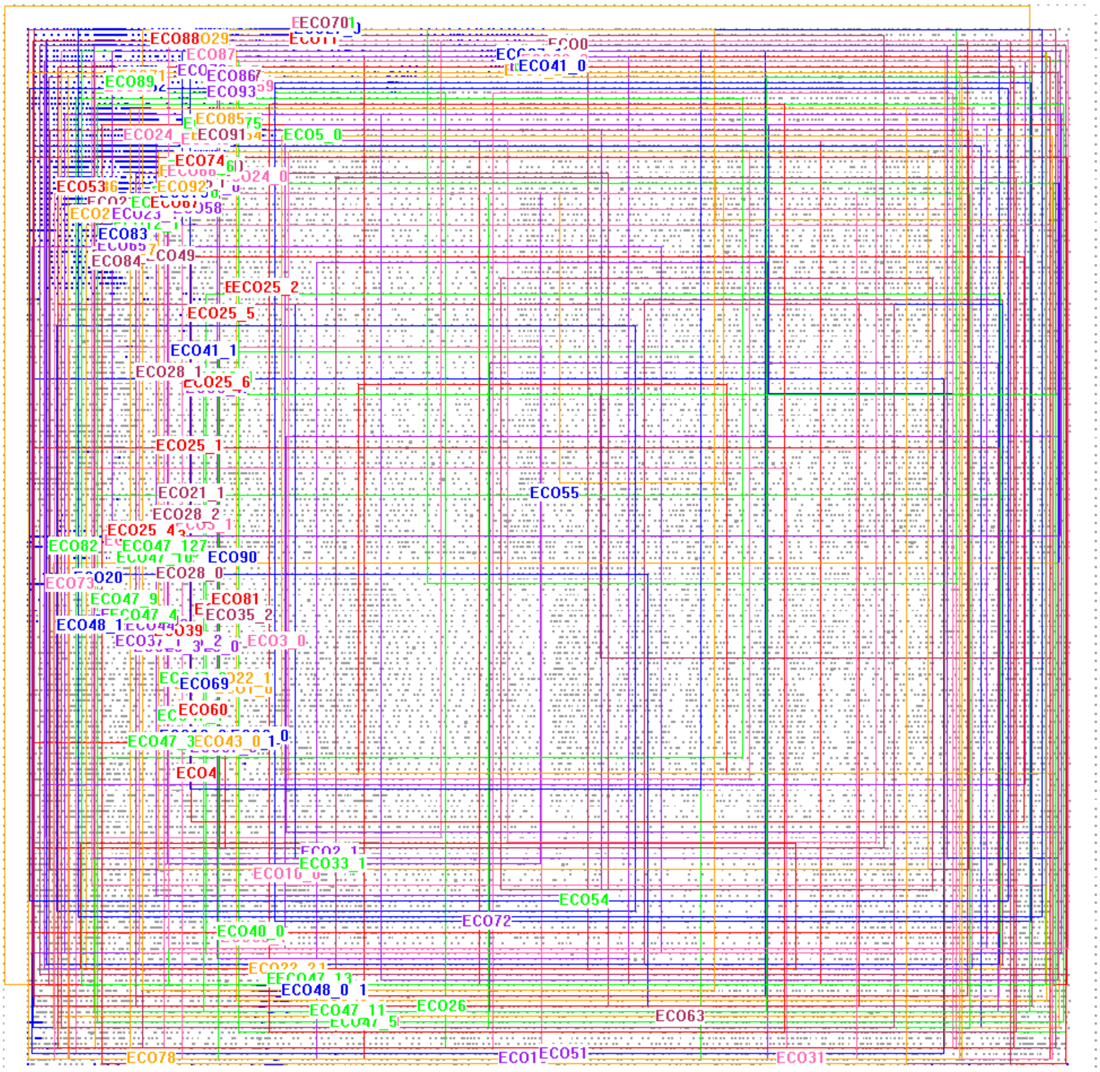
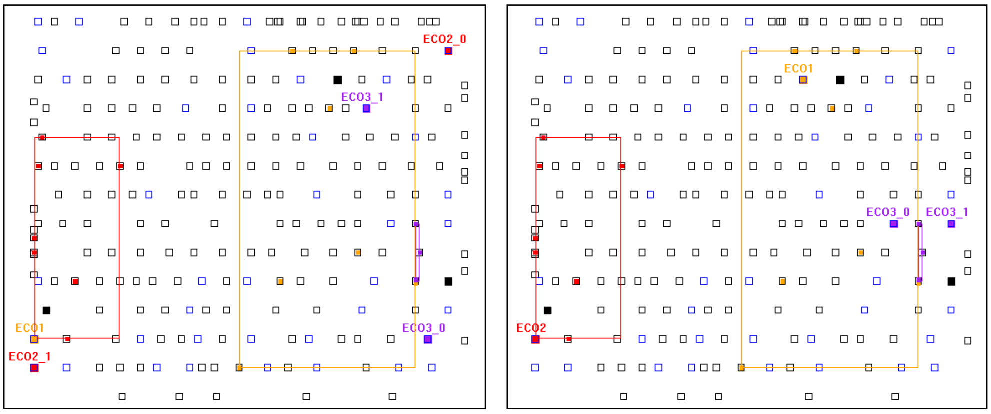


Figure 4.3: Illustration of one real design



(a) Direct　　　　(b) Greedy/ECOS

Figure 4.4: Testcase4: The HPBB, selected spare cells, and related pins of each ECO are highlighted.

Chapter 5

Conclusions

In this thesis, we proposed a metal-only ECO synthesizer, named ECOS, to efficiently implement incremental functional and timing design changes. Unlike the timing-consuming hand-editing method, we can reach the same goal faster and even better. We can easily extend ECOS to timing ECO without any change. Experimental results revealed that guided ABC can make the synthesis more flexible and reasonable, while stable matching can alleviate the competition. In addition, ECOS considers the photomask cost throughout the whole flow.

Bibliography

[1] S. Golson. “The Human ECO Compiler.” In proceedings of Synopsys Users Group, 2004, pp. 1—57.

[2] Synopsys, Inc. http://www.synopsys.com/.

[3] Cadence Design System. http://www.cadence.com/.

[4] Magma Design Automation. http://www.magma-da.com/.

[5] J.-Y. Li and Y.-L. Li. “An Efficient Tile-Based ECO Router with Routing Graph Reduction and Enhanced Global Routing Flow.” In Proceedings of International Symposium on Physical Design, 2005, pp. 7—13.

[6] Y.-P. Chen, J.-W. Fang and Y.-W. Chang. “ECO Timing Optimization Using Spare Cells and Technology Remapping.” In Proceedings of International Conference on Computer-Aided Design, 2007, pp. 530—535.

[7] C.-P. Lu, M. C.-T. Chao, C.-H. Lo, and C.-W. Chang. “A Metal-Only-ECO Solver for Input Slew and Output Loading Violations.” In Proceedings of International Symposium on Physical Design, 2009, pp. 191—198.

[8] Y.-M. Kuo, Y.-T. Chang, S.-C. Chang and M. Marek-Sadowska. “Engineering Change Using Spare Cells with Constant Insertion.” In Proceedings of International Conference on Computer-Aided Design, 2007, pp. 544—547.

[9] ABC: A system for sequential synthesis and verification. http://www.eecs.berkeley.edu/~alanmi/abc/.

[10] D. Gale and L. S. Shapley. “College Admissions and the Stability of Marriage.” In American Mathematical Monthly, 1962, Vol. 69, pp. 9—14.