Redundant-Wires-Aware ECO Timing and Mask Cost Optimization

Shao-Yun Fang[1], Tzuo-Fan Chien[1] and Yao-Wen Chang[12]
[1]Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan
[2]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
yuko703@eda.ee.ntu.edu.tw; hoolank@eda.ee.ntu.edu.tw; ywchang@cc.ee.ntu.edu.tw

***Abstract*—Spare cells are often used in engineering change order (ECO) timing optimization. By applying spare-cell rewiring techniques, timing-violated paths in a design can be fixed. In addition, mask re-spin cost economization has become a critical challenge for modern IC design, and it can be achieved by reducing the number of layers used to rewire spare cells. This paper presents the first work for the problem of ECO timing optimization considering redundant wires (unused wires or dummy metals) to minimize the number of rewiring layers. We first propose a multi-commodity flow model for the spare-cell selection problem and apply integer linear programming (ILP) to simultaneously optimize all timing-violated paths. The ILP formulation minimizes the number of used spare cells and considers the routability of the selected spare cells. Then, we develop a tile-based ECO router which minimizes the number of rewiring layers by reusing redundant wires. Experimental results based on five industry benchmarks show that our algorithm not only effectively resolves timing violations but also reduces the number of rewiring layers under reasonable runtime.**

I. Introduction

In modern circuit design, engineering change order (ECO), which can be classified into functional ECO and timing ECO, is often used to handle incremental design changes. Instead of redesigning a circuit from scratch, ECO gives the opportunities such that a design could meet new specifications or design constraints with only small circuit modifications. In addition, reducing the number of re-spin masks becomes critical since the costs of photomasks are drastically increased [5]. Reusing transistor-layer masks and only changing the metal-layer masks, which is known as metal-only ECO, can significantly reduce re-spin cost since transistor-layer masks are much more expensive than metal-layer ones.

Spare cells are often used to realize metal-only ECO. For functional ECO, Kuo et al. utilized constant insertion to increase the functional flexibilities of spare cells [7], and Modi and Marek-Sadowska proposed a simulated annealing-based framework and further optimized the total wire length [10]. Recently, Jiang et al. proposed a matching-based ECO synthesizer using spare cells to handle functional changes [6]. These studies emphasize on functional correction, and timing is not the main focus.

For timing ECO, the most popular techniques utilizing spare cells to fix timing violations are buffer insertion and gate sizing. By inserting an adjacent buffer on a timing-violated path or by replacing an original gate on the path with a spare cell which has the same function and stronger driving capability may both improve the circuit timing. After selecting spare cells to fix timing violations, spare-cell rewiring (ECO routing) is performed to complete the optimization.

To perform gate sizing and buffer insertion using spare cells, Chen et al. proposed a timing optimization flow based on dynamic programming [2]. Later, Lu et al. utilized buffer insertion to resolve timing violations under input-slew and output-loading constraints [9]. Recently, Chen et al. proposed a reconfigurable ECO flow to optimize circuit timing and minimize IR drop by gate sizing and buffer insertion as well [1]. All of them sequentially perform gate sizing and/or buffer insertion on one timing-violated path at a time. Although these sequential manners can handle the problem efficiently, the lack of global view may cause them to use more spare cells to fix timing violations. Rewiring more spare cells typically implies that it is more difficult to rewire all nets. It is also possible that the timing violations cannot be resolved since the sequential methods cannot guarantee to find a desired solution.

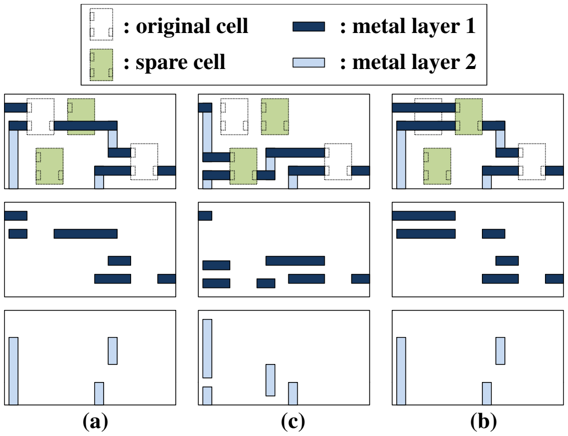


Fig. 1. (a) A timing-violated path. (b) Spare cell rewiring by changing the horizontal and the vertical metal layers. (c) Spare cell rewiring by changing only the horizontal metal layer, and thus the mask for the vertical metal layer can be reused to save the cost.

On the other hand, ECO routing is usually complicated due to the large amount of existing wires/obstacles. A gridless router has more flexibility to deal with such a routing environment than a grid-based one. Cong et al. proposed an implicit connection graph to model the routing plane [3]. However, their method tends to generate too many graph nodes to perform a searching operation. Li et al. presented a tile-based ECO router that increases its routing speed with a routing graph reduction technique [8]. Another issue that has not been addressed in the previous work is how to minimize the number of rewiring layers, and therefore minimize mask re-spin cost. See Fig. 1 for an example. Fig. 1(a) shows a timing-violated path consisting of two original cells and two routing layers (one horizontal metal layer and one vertical metal layer). If the lower spare cell is chosen for performing gate sizing, the two metal masks must be changed for ECO routing, as shown in Fig. 1(b). However, if we

This work was partially supported by ITRI, SpringSoft, Synopsys, TSMC, and NSC of Taiwan under Grant No's. NSC 98-2622-E-002-005-A2, NSC 98-2221-E-002-119-MY3, NSC 97-2221-E-002-237-MY3, NSC 96-2628-E-002-249-MY3, and NSC 96-2628-E-002-248-MY3.

978-1-4244-8192-7/10/$26.00 ©2010 IEEE

choose the upper spare cell for gate sizing, as illustrated in Fig. 1(c), by using the unused/redundant wires left by the original cell, the vertical metal layer can remain unchanged. Therefore, we can save half of the mask re-spin cost.

In this paper, we introduce the problem of ECO timing optimization and rewiring layer minimization. To the best of our knowledge, this is the first work for this problem. We propose a post-mask ECO timing optimization flow consisting of two stages: (1) the routing resource-aware spare-cell selection stage followed by (2) the redundant-wires-aware ECO routing stage. Our algorithm iteratively performs spare-cell selection and ECO routing to find the minimum number of changed metal masks and via masks. Experimental results based on five industry benchmarks show that our algorithm not only effectively resolves timing violations but also reduces the number of rewiring layers under reasonable runtime.

The rest of this paper is organized as follows: Section II gives the preliminaries and the problem formulation of this paper. In Section III, the flow of our algorithms is given. Section IV and Section V detail the spare-cell selection algorithm and the ECO routing stage, respectively. Experimental results are reported in Section VI. Finally, we conclude our work in Section VII.

II. Preliminaries

In this section, the preliminaries of ECO timing optimization problem and redundant wires are given. First, the timing model and the bounding polygon technique used to identify spare-cell candidates are described in Section II-A. Section II-B introduces two types of redundant wires, and the problem formulation of ECO timing optimization and rewiring layer minimization is presented in Section II-C.

A. ECO Timing Optimization

A timing path is defined as a circuit path which begins with a primary input or a flip-flop output pin and ends with a primary output or a flip-flop input pin. In the spare-cell selection stage, we first identify the timing paths violating the timing constraints, which are defined as *ECO paths* [2].

To optimize ECO timing paths, we use gate sizing and buffer insertion techniques illustrated in Fig. 2. We denote G^E as the set of gates along the ECO paths and G^S as the set of spare cells. Fig. 2(a) shows an ECO path composed of three gates. Suppose there are a buffer-type spare cell g_2^S and a gate-type spare cell with the same function and stronger drivability g_1^S adjacent to the ECO path. To fix the timing violation, we may try to insert g_2^S in an adjacent net, and replace g_1^E with the adjacent spare cell g_1^S, as illustrated in Fig. 2(b).

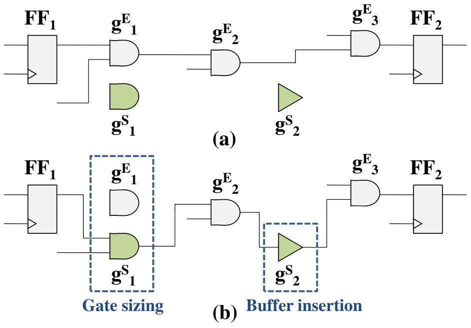


Fig. 2. (a) An ECO path and some adjacent spare cells. (b) A probable solution by buffer insertion and gate sizing techniques.

To identify gate sizing and buffer insertion candidates for ECO paths, we use the same timing model and bonding polygon technique as in [2] to select spare cells only in the valid region of a gate and prune other spare cells outside the bonding box. Another observation presented in [2] is the shielding effect. Shielding effect states that sizing a gate $g(i)$ and inserting a buffer substantially affect the delays of its fan-in and fan-out gates. But the influences on those gates not directly connected to $g(i)$ can be ignored. This property inspires our concurrent ECO timing optimization method.

B. Redundant Wires

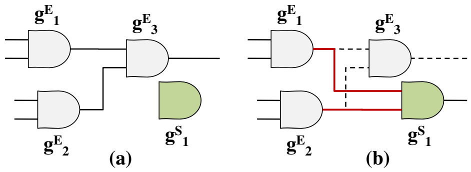


Fig. 3. (a) The original netlist. (b) The netlist after gate-sizing operation.

In addition to optimizing all timing paths, the other objective is to minimize the number of rewiring layers. Minimizing the number of used spare cells and reusing redundant wires are two possible ways to reach the objective. There are two types of redundant wires that can be reused: (1) unused wires and (2) dummy metals.

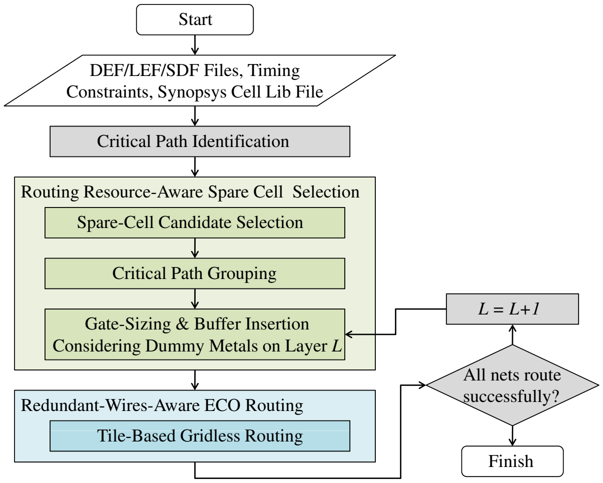
Unused wires are generated from the original timing-violated paths. When gate-sizing and buffer-insertion operations are performed to fix the timing violations, the wires of original paths are released and thus free, as shown in Fig. 3. Dummy metals used to achieve layout uniformity are the second type of redundant wires. If the unused wires and the dummy metals are on the metal layers whose masks are unchanged, they remain on the metal layers and thus can be reused as redundant wires. By reusing redundant wires, rewiring nets can cross over many obstacles originally existing in a layout. As a result, it is possible that the number of rewiring layers can be reduced in the ECO routing stage.

C. Problem Formulation

Based on the above definition, the problem formulation of redundant-wires-aware ECO timing and mask cost optimization can be described as follows:

Problem 1: Given a routed layout with a set of placed spare cells, perform metal-only ECO timing optimization by using gate-sizing and buffer-insertion techniques such that all timing violated paths are fixed and the number of rewiring layers is minimized.

III. The ECO Timing Optimization Algorithm Flow



Start
DEF/LEF/SDF Files, Timing Constraints, Synopsys Cell Lib File
Critical Path Identification
Routing Resource-Aware Spare Cell Selection
Spare-Cell Candidate Selection
Critical Path Grouping
Gate-Sizing & Buffer Insertion Considering Dummy Metals on Layer L
L = L+1
All nets route successfully?
Redundant-Wires-Aware ECO Routing
Tile-Based Gridless Routing
Finish

Fig. 4. The ECO timing optimization flow.

In this section, we present the overview of our ECO timing optimization algorithm. Fig. 4 shows our ECO timing optimization flow.

Given a routed chip layout as the input, we apply static timing analysis (STA) to identify the timing-violated paths as the ECO paths. Our ECO timing optimization flow consists of two major stages: the routing resource-aware spare-cell selection stage followed by the redundant-wires-aware ECO routing stage.

In the routing resource-aware spare-cell selection stage, we first identify the spare-cell candidates of each ECO path. After that, we group the ECO paths that share spare-cell candidates. For each group, we construct a multi-commodity flow model and apply ILP to solve the spare-cell selection problem. Under the constraints that all timing violations have to be fixed, an optimal solution with the minimum number of selected spare cells is generated. Finally, a tile-based gridless router minimizing the number of rewiring layers by utilizing redundant wires is applied to complete the routing. Since the spare-cell selection ILP formulation depends on how many rewiring layers are used, our algorithm needs to perform the spare-cell selection and the ECO routing in an iterative way to find the minimum number of rewiring layers.

We detail the two stages in Section IV and Section V, respectively.

IV. Routing-Resource-Aware Spare-Cell Selection

The objective of the spare-cell selection problem is to economize routing resources when all paths meet their timing constraints. In this section, we propose a multi-commodity flow model for this problem and formulate the model into an ILP formulation. First, we show the transformation from ECO paths and their spare-cell candidates to a multi-commodity flow model in Section IV-A. Then, the basic ILP-formulation for routing-resource-aware spare-cell selection problem is presented in SectionIV-B. Finally, in Section IV-C, we provide an ILP reduction technique and analyze the computational complexity of the ILP formulation.

A. Multi-Commodity Model

On an ECO path, we use the same timing model and the same bounding polygon technique in [2] to select spare-cell candidates for each gate. Unlike the work [2] that handle ECO paths sequentially, we perform ECO timing optimization on related ECO paths simultaneously by grouping ECO paths according to their spare-cell candidates. Two ECO paths would be grouped together if they share some spare-cell candidates.

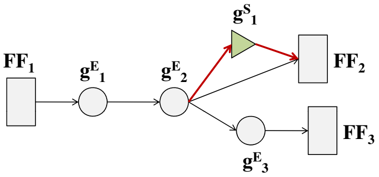


Fig. 5. An example of the multi-commodity flow model, in which a buffer-type spare cell can be selected by an ECO path.

By observing that the spare-cell selection problem is analogous to the multi-commodity flow problem, we transform the grouped ECO paths and their spare-cell candidates into a corresponding multi-commodity flow model. Fig. 5 gives an example. There are two ECO paths in the group. One is from flip-flop $FF1$ to flip-flop $FF2$, and the other is from flip-flop $FF1$ to flip-flop $FF3$. Spare cell g_1^S is a buffering candidate of the net driven by gate g_2^E along ECO path 1. To incorporate the spare cell into our flow model, we add edges (g_2^E, g_1^S) and $(g_1^S, FF2)$. Therefore, Fig. 5 shows two possible paths routed from g_2^E to $FF2$. In the case of overlapping ECO paths, gate-sizing and buffer-insertion operations must be performed on all the paths simultaneously. As shown in Fig. 6(a), there are two overlapping ECO paths. The spare cell g_1^S is a sizing candidate of the gate g_3^E. Since the gates g_1^E and g_2^E must route through either the gate g_3^E or the spare cell g_1^S at the same time, we add the nodes defined as *intermediate nodes* to achieve this goal, as shown in Fig. 6(b). By adding intermediate nodes, we can force the two ECO paths to route through the same intermediate node in our ILP formulation.

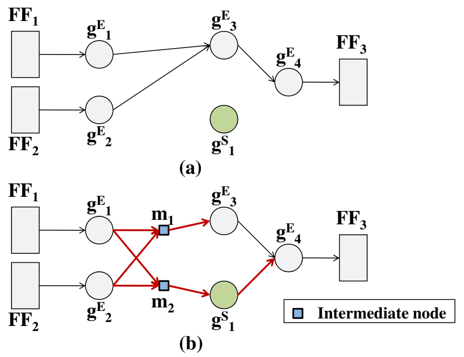


Fig. 6. Incorporating a spare cell into the flow model with overlapped ECO paths. (a) The original netlist. (b) The corresponding multi-commodity flow model with intermediate nodes.

B. Basic ILP Formulation

After we construct the multi-commodity flow model, we can get a directed graph $H = (V, E)$ in which a node represents a gate, an I/O pin, or a flip-flop in the model, and a directed edge represents a possible way/edge of ECO paths. Then, we can use an ILP formulation to select spare cells. Our objectives are to economize routing resources for ECO routing and to select spare cells with better routability. We observed that rewiring nets with more redundant wires in their neighborhoods are more likely to be rewired successfully. Therefore, we try to use the minimum number of spare cells to fix all timing violations and use those spare cells with higher redundant-wire densities. On the other hand, since we cannot decide which unused wires would be generated before the spare-cell selection, we use the density of dummy metals to predict the routability of a spare cell.

The notations used in the ILP formulation are as follows:

- K: group of ECO paths.
- G^E: set of gates along the given ECO paths.
- G^S: set of spare-cell candidates.
- E: set of edges in the directed graph.
- I: set of intermediate nodes.
- $s(k)$: source flip-flop of a path k, $k \in K$.
- $t(k)$: sink flip-flop of a path k, $k \in K$.
- $e_{i,j}$: directed edge representing interconnection from gate i to gate j.
- $K(e_{i,j})$: set of ECO paths flowing through edge $e_{i,j}$.
- $d(e_{i,j})$: delay of gate i and interconnection delay from gate i to gate j.
- $D(e_{i,j})$: dummy metal gain of routing through edge $e_{i,j}$. If gate i or gate j is a spare cell, $D(e_{i,j})$ is determined by the dummy metal density in the bounding box that encloses gate i and all its fan-out gates. Otherwise, $D(e_{i,j})$ is assigned a large number M that is not less than any dummy metal gain.
- $f^k(e_{i,j})$: 0-1 integer variable that denotes if an ECO path k is routed through $e_{i,j}$. $f^k(e_{i,j}) = 1$ if the path k is routed from gate i to gate j through $e_{i,j}$; $f^k(e_{i,j}) = 0$, otherwise.
- T_{clock}: clock period.

- $T_{setup}(k)$: setup time of the end point of an ECO path k. If the end point is a primary output, $T_{setup}(k)$ is zero.
- $L_s(k)$: clock latency from the clock root to the source flip-flop of an ECO path k. If the source is a primary input, $L_s(k)$ is zero.
- $L_t(k)$: clock latency from the clock root to the sink flip-flop of an ECO path k. If the sink is a primary output, $L_t(k)$ is zero.

Based on the notations, the spare-cell selection problem can be formulated as follows:

$$maximize \quad \sum_{e_{i,j}\in E}\sum_{k\in K} f^k(e_{i,j}) \times D(e_{i,j})$$

$$subject\ to \quad \sum_{e_{s(k),j}\in E} f^k(e_{s(k),j}) = 1, \forall k \in K, \tag{1}$$

$$\sum_{e_{i,t(k)}\in E} f^k(e_{i,t(k)}) = 1, \forall k \in K, \tag{2}$$

$$\sum_{e_{i,j}\in E} f^k(e_{i,j}) \times d(e_{i,j}) \le T_{clock} + L_t(k) - L_s(k) - T_{setup}(k), \forall k \in K, \tag{3}$$

$$\sum_{e_{h,i}\in E} f^k(e_{h,i}) = \sum_{e_{i,j}\in E} f^k(e_{i,j}), \forall i \in G^E \cup G^S \cup I, \forall k \in K, \tag{4}$$

$$\sum_{e_{i,j}\in E}\left(\frac{1}{|K(e_{i,j})|}\sum_{k\in K(e_{i,j})} f^k(e_{i,j})\right) \le 1, \forall j \in G^S, \tag{5}$$

$$f^{k_1}(e_{i,j}) = f^{k_2}(e_{i,j}), \forall k_1, k_2 \in K_{e_{i,j}}, \forall e_{i,j} \in E. \tag{6}$$

The objective function is to maximize the total dummy metal gain, which would be achieved by selecting the minimum number of spare cells and selecting the spare cells with higher dummy metal densities while the timing constraints are all satisfied. Constraint 1 and Constraint 2 ensure that all ECO paths can find a unique way by assigning the amount of flow from the source flip-flop to the sink flip-flop to be 1. Constraint 3 ensures that the slack of each ECO path is non-negative. Constraint 4 is used to guarantee that the number of paths flowing into a node is equal to that draining from the node. Constraint 5 guarantees that no two non-overlapping ECO paths route through a spare cell. Fig. 7 shows an example for this constraint. There are three ECO paths in Fig. 7(a). $s(k)$ and $t(k)$ represent the source and the sink of an ECO path k, respectively. The notation $\{k1, k2, \ldots\}$ on an edge denotes the set $K(e_{i,j})$ of an edge $e_{i,j}$. Fig. 7(b) shows the multi-commodity flow model of Fig. 7(a). Since the buffer-type spare cell g_2^S can only be used by either ECO path 3 or ECO paths 1 and 2, Constraint 5 ensures that at most one of the two paths passes through g_2^S. Finally, under Constraint 6, the overlapping part of any two ECO paths can be forced to perform gate sizing or buffer insertion simultaneously. As shown in Fig. 7, ECO path 1 and ECO path 2 are overlapped on gate g_3^E. Therefore, they have to either perform gate sizing on spare cell g_1^S or maintain the original paths passing through g_3^E at the same time. This can be guaranteed by adding the intermediate nodes and Constraint 6.

Although the above formulation guarantees to find an optimal solution, it may suffer from high time complexity when the number of ECO paths grows exponentially. Therefore, it is necessary to provide ILP reduction techniques to reduce the numbers of variables and constraints.

C. Optimality-Preserving ILP Reductions

Now we present ILP reduction techniques to reduce the numbers of variables and constraints. In order to maintain the solution optimality, only the redundant solutions can be reduced.

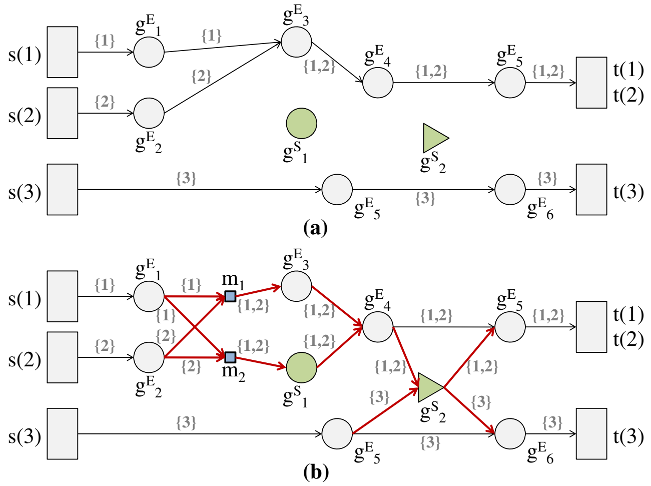


Fig. 7. (a) The original netlist. (b) The corresponding multi-commodity flow model.

Since ECO paths do not route through every edge, we do not need to define $|K|$ decision variables $f^k(e_{i,j}), \forall k \in K$, for each edge. Instead, we only define $|K(e_{i,j})|$ decision variables on edge $e_{i,j}$. Therefore, the number of the decision variables is reduced from $|E||K|$ to $\sum_{e_{i,j}\in E}|K_{e(i,j)}|$. Further, with the variables reduction, we can reduce the number of Constraint 4 from $(|G^E| + |G^S| + |I|)|K|$ to $\sum_{i\in G^E\cup G^S\cup I}|\bigcup_{j\in FO(i)} K(e_{i,j})|$, where $FO(i)$ is the set of fan-out nodes of i.

In addition, the number of Constraint 6 dominates all other constraints,which is $\sum_{e_{i,j}\in E}|K(e_{i,j})|(|K(e_{i,j})| - 1)/2$. This can be reduced to $\sum_{e_{i,j}\in E}(2|K(e_{i,j})| + 3)$ by applying the following formulation. For each edge $e_{i,j}$, we define

$$A(e_{i,j}) = \mathbf{AND}(f^{k_1}(e_{i,j}), f^{k_2}(e_{i,j}), \ldots), \tag{7}$$

$$O(e_{i,j}) = \mathbf{OR}(f^{k_1}(e_{i,j}), f^{k_2}(e_{i,j}), \ldots), \quad k_1, k_2, \ldots \in K(e_{i,j}). \tag{8}$$

Then, Constraint 6 can be transformed as follows:

$$A(e_{i,j}) = O(e_{i,j}), \forall e_{i,j} \in E. \tag{9}$$

We need $|K(e_{i,j})| + 1$ constraints to formulate each of $A(e_{i,j})$ and $O(e_{i,j})$ in the model. Therefore, we can reduce the number of Constraint 6 from $\sum_{e_{i,j}\in E}|K(e_{i,j})|(|K(e_{i,j})| - 1)/2$ to $\sum_{e_{i,j}\in E}(2|K(e_{i,j})| + 3)$. Constraint 6 can be transformed into the following constraints.

$$A(e_{i,j}) - f^k(e_{i,j}) \le 0, \forall k \in K(e_{i,j}), \forall e(i,j) \in E, \tag{10}$$

$$A(e_{i,j}) - \sum_{k\in K(e_{i,j})} f^k(e_{i,j}) + |K(e_{i,j})| - 1 \ge 0, \quad \forall e_{i,j} \in E, \tag{11}$$

$$O(e_{i,j}) - f^k(e_{i,j}) \ge 0, \forall k \in K(e_{i,j}), \forall e_{i,j} \in E, \tag{12}$$

$$O(e_{i,j}) - \sum_{k\in K(e_{i,j})} f^k(e_{i,j}) \le 0, \forall e_{i,j} \in E, \tag{13}$$

$$A(e_{i,j}) = O(e_{i,j}), \forall e_{i,j} \in E. \tag{14}$$

We have the following theorem for the complexity reduction by applying the above techniques:

Theorem 1: By applying the above techniques, the number of ILP variables can be reduced from $|E||K|$ to $\sum_{e_{i,j}\in E}|K(e_{i,j})|$ and the number of ILP constraints can be reduced from $O(|K||G^E| + |K||G^S| + |K|^2|E|)$ to $O(|K|(|G^E| + |G^S| + |E|))$.

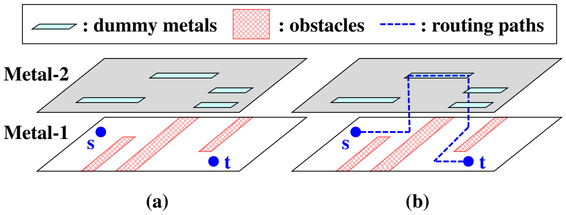
Note that since $|E|$ dominates the complexity, the problem size can significantly be reduced. Moreover, since only redundant variables are deleted, we have the following theorem for the solution optimality (i.e., the maximum total dummy metal gain):

Theorem 2: The reduced ILP formulation preserves the solution optimality and provides a solution with the maximum total dummy metal gain.

V. ECO Routing Considering Redundant Wires

In this section, we utilize the two types of redundant wires for spare-cell routing to minimize the number of rewiring layers.

Since the I/O pins of spare cells are on metal-1 layer, if we use metal-L, $L \neq 1$, to rewire spare cells, the metal masks and via masks from metal-1 to metal-L must be continuously changed. Therefore, our router iteratively finds the minimum number of rewiring layers by starting from using only metal-1. Note that the usages of the two types of redundant wires are different. The unused wires on different layers may be connected with each other through existing vias. On the other hand, dummy metals on different layers are not connected through vias. Therefore, only the dummy metals on the lowest unchanged metal layer can be used to rewire (since the via layer right below this lowest unchanged metal layer can be used to connect the dummy metals, while the via layer above this lowest unchanged metal layer remains unchanged).



(a) (b)

Fig. 8. (a) Routing configuration before rewiring. (b) Rewiring a net by reusing a dummy metal.

For those metal layers used for rewiring, we keep all existing wires unchanged but remove unused wires for better routability. The existing wires become a large amount of obstacles. Since the direction of the dummy metals on the lowest unchanged layer is perpendicular to that of the top rewiring layer, as shown in Fig. 8(a), the dummy metals are useful to route over the obstacles, as shown in Fig. 8(b).

As the flow shown in Fig. 4, after a solution is generated by solving ILP, the ECO router tries to route all nets on metal-1. Once any rewiring net cannot be routed by using only metal-1, the dummy metals on metal- 2 have to be considered and the ILP problem has to be solved again. By this iterative manner, the minimum number of rewiring layers can be found when all rewiring nets can be routed successfully.

VI. Experimental Results

TABLE I
Statistics of the benchmark circuits.

Circuit name	Gate count	# Spare cells	# ECO paths	TNS (ns)	# Layers	# Nets
Industry1	28927	860	16	9.8	8	28167
Industry2	200504	860	80	312.0	8	199849
Industry3	91107	860	27	319.0	8	90443
Industry4	18932	860	22	70.0	8	18265
Industry5	38011	8600	137	161.0	8	29603

Our algorithm was implemented in the C++ programming language on a 2.33GHz Linux workstation with 12 GB memory. The public SCIP [11] is applied to solve the ILP. The experiments were evaluated with five industrial designs adopted in [2]. The profiles of these circuits are given in Table I, where "Circuit name" lists the names of circuits, "Gate count" lists the number of gates of each circuit, "# Spare cells" gives the number of spare cells, "# ECO paths" gives the number of timing-violated paths before optimization, "TNS" gives the amount of total negative slacks, "# Layers" gives the number of layers of each circuit, and "# Nets" gives the number of nets of each circuit which have been routed.

Table II compares two algorithms: (1) DCP which is a sequential spare-cell selection algorithm based on dynamic programming in [2] and (2) our routing resource-aware spare-cell selection with ILP reductions. Both of DCP and our algorithm apply the bounding polygon technique proposed in [2]. Since DCP does not fix all the timing violations in Industry3 and Industry4, the numbers of used spare cells in Industry3 and Industry4 of DCP are not available. Table II shows that our routing resource-aware spare-cell selection algorithm not only resolves all timing violations, but also uses much fewer (14% in average compared with DCP) spare cells than DCP. Especially for a circuit with many timing-violated paths, like Industry5, DCP uses many more spare cells because the sequential optimization method performs gate-sizing and buffer-insertion operations without considering global optimality. Further, when a large amount of spare cells are used to fix timing violations, the success rate of rewiring all nets is decreased. Even worse, timing violations may not be resolved by DCP, like Industry4, because the sequential manner does not guarantee to get a desired solution.

In addition, since there is no previous work on the ECO timing optimization considering redundant wires, we compare our ECO timing optimization flow with DCP followed by our redundant-wires-aware ECO routing in Table III. In the column of "# changed masks", L represents metal masks and V represents the extra via mask used to access redundant wires. For example, $1L + 1V$ means that the changed masks are metal-1 and one additional via mask via-12 such that we can use the redundant wires on metal-2. $2L+1V$ means that we use metal-1, via mask via-12, and metal-2 to rewire, but we do not use redundant wires on metal-3. Since DCP does not fix all the timing violations in Industry3 and Industry4, we do not rewire the circuits. For Industry5, DCP followed by our redundant-wires-aware ECO routing cannot route all the rewiring nets by using metal-1 to metal-8 because the number of used spare cells are so many that the number of rewiring nets significantly increased. In the cases without changing the extra via mask such as Industry2 in our results, the rewiring nets cannot be routed to the upper adjacent layer and thus the numbers of reused dummy metals and reused unused wires are zero.

Observing from the experimental results, minimizing the number of used spare cells and considering redundant wires in the spare-cell selection stage can reduce the number of rewiring layers. Besides, the number of reused dummy metals in each case are much more than the number of reused unused wires, as shown in Table III. Namely, dummy metals are the main redundant wires that can be reused to rewire. Therefore, considering the density of dummy metals on routing paths in the spare-cell selection stage to predict the routability in ECO routing stage is desirable.

Fig. 9 shows the routing layout of Industry5. In Industry5, we use metal-1 as the rewiring layer, and we further reuse the dummy metals on metal-2 and the unused wires on metal-2 to metal-8 to complete ECO routing. The red lines are existing wires on metal-1. The black lines in the figure mark the routing paths of the rewiring nets. Fig. 10(a) and Fig. 10(b) show parts of a rewiring net on metal-1 and metal-2, respectively. Since metal-1 is a horizontal layer, the obstacles are mostly horizontal wires such that the rewiring nets are difficult to route vertically. However, as shown in Fig. 10(b), by reusing a dummy metal, the rewiring net can easily cross over the obstacles without detour.

TABLE II

Comparison between DCP [2] and our routing resource-aware spare-cell selection.

Circuit name	Initial	DCP [2]				Ours with ILP reduction			
	TNS (ns)	TNS (ns)	Imp. ratio	# Used spare cells	Run time (s)	TNS (ns)	Imp. ratio	# Used spare cells	Run time (s)
Industry1	9.8	0.00	100.00%	4	6.12	0.00	100.00%	1	42.42
Industry2	312.0	0.00	100.00%	13	25.71	0.00	100.00%	7	109.48
Industry3	319.0	14.93	95.32%	**N/A**	12.33	0.00	100.00%	4	26.29
Industry4	70.0	6.27	91.04%	**N/A**	24.31	0.00	100.00%	13	55.41
Industry5	161.0	0.00	100.00%	122	1761.16	0.00	100.00%	7	3182.87
Avg.			97.27%	46.33	**0.54x**		100.00%	6.4	**1.00**

TABLE III

Comparison between DCP [2] followed by our redundant-wires-aware ECO routing and our complete ECO timing optimization flow.

Circuit name	DCP [2] + Our redundnat-wires-aware ECO routing							Our spare-cell selection + Our redundnat-wires-aware ECO routing						
	TNS (ns)	# Used spare cells	# Rewiring nets	# Reused dummy metals	# Reused unused wires	# Changed masks	Run time (s)	TNS (ns)	# Used spare cells	# Rewiring nets	# Reused dummy metals	# Reused unused wires	# Changed masks	Run time (s)
Industry1	0.00	4	8	0	0	**2L + 1V**	37.20	0.00	1	2	32	1	**1L + 1V**	46.96
Industry2	0.00	13	32	0	0	**2L + 1V**	127.43	0.00	7	13	0	0	**2L + 1V**	377.56
Industry3	14.93	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**	12.33	0.00	4	5	12	0	**1L + 1V**	1149.89
Industry4	6.27	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**	24.31	0.00	13	17	237	8	**1L + 1V**	68.69
Industry5	0.00	122	331	**N/A**	**N/A**	**N/A**	2063.25	0.00	7	21	402	19	**1L + 1V**	3206.91

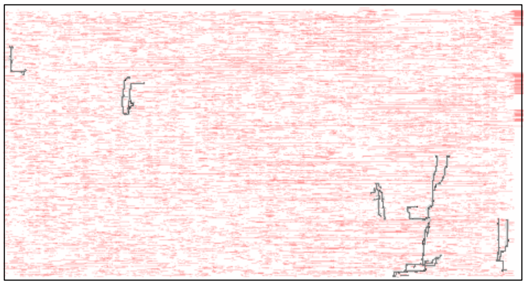


Fig. 9. The routing wires on metal-1 in Industry5. The black lines mark the routing paths of the rewiring nets.

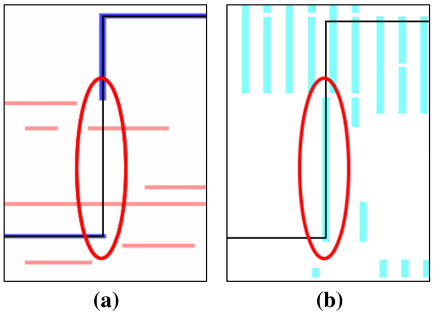


Fig. 10. (a) Part of a rewiring net on metal-1. (b) Part of a rewiring net on metal-2.

VII. Conclusions

This paper has presented a post-mask ECO timing optimization flow considering redundant wires reused to minimize mask re-spin cost. Compared with the prior work, our spare-cell selection algorithm offers a more global view on timing optimization and simultaneously performs gate sizing and buffer insertion on overlapping ECO paths. Experimental results based on five industry designs have shown that our flow with redundant wires reuse not only robustly resolves all timing violations, but also reduces mask re-spin cost under reasonable runtime.

References

[1] H.-T. Chen, C.-C. Chang and T.-T. Hwang, "New spare cell design for IR drop minimization in engineering change order," *Proceedings of Design Automation Conference*, pp. 402–407, July 2009.

[2] K.-H. Ho, Y.-P. Chen, J.-W. Fang, and Y.-W. Chang, "ECO timing optimization using spare cells and technology remapping," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 29, No. 5, pp. 697–710, May 2010.

[3] J. Cong, J. Fang, and K.-Y. Khoo, "DUNE - A multilayer gridless routing system," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 20, No. 5, pp. 633–647, May 2001.

[4] J. Dion and L. M. Monier, "Contour: A tile-based gridless router," Western Research Laboratory Research Report, Palo Alto, CA, Research Report, March 1995.

[5] Reports of the International Technology Roadmap for Semiconductors, 2007, http://www.itrs.net/.

[6] I. H.-R. Jiang, H.-Y. Chang, L.-G. Chang, and H.-B. Hung, "Matching-based minimum-cost spare cell selection for design changes," *Proceedings of Design Automation Conference*, pp. 408–411, July 2009.

[7] 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, March 2009.

[8] Y.-L. Li, J.-Y. Li, and W.-B. Chen, "An efficient tile-based ECO router using routing graph reduction and enhanced global routing flow," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 26, No 2, pp. 345–358, February 2007.

[9] 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," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 29, No. 2, pp. 240–245, February 2010.

[10] N.-A. Modi and M. Marek-Sadowska, "ECO-Map: Technology remapping for post-mask ECO using simulated annealing," *Proceedings of International Conference on Computer-Aided Design*, pp. 652–657, October 2008.

[11] Solving Constraint Integer Programs. http://scip.zib.de/.