行政院國家科學委員會專題研究計畫 成果報告

整合可製造性設計方法應用於 X-結構時脈繞線合成之研究
研究成果報告(精簡版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 97-2221-E-343-008-
執 行 期 間 ：97 年 08 月 01 日至 98 年 07 月 31 日
執 行 單 位 ：南華大學資訊工程學系

計 畫 主 持 人：蔡加春

計畫參與人員：碩士班研究生-兼任助理人員：古琳正
博士班研究生-兼任助理人員：郭仲傑

報 告 附 件 ：出席國際會議研究心得報告及發表論文

處 理 方 式 ：本計畫涉及專利或其他智慧財產權，1 年後可公開查詢

中 華 民 國 98 年 08 月 17 日

行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※整合可製造性設計方法應用於 **X-**結構時脈繞線合成之研究 ※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：☑個別型計畫　　□整合型計畫

計畫編號：NSC 97－2221－E－343－008－

執行期間：97 年 8 月 1 日至 98 年 7 月 31 日

計畫主持人：蔡加春 教授

共同主持人：

計畫參與人員：郭仲傑（博士生）

古琳正、尤志豪（碩士生）

及臨時大專生

執行單位：南華大學資工系

中　華　民　國　　98　年　8　月　16　日

整合可製造性設計方法應用於 X-結構時脈繞線合成之研究

Integrated Methods of Design for ManufacturabilityApplied for X-Architecture Clock Routing Synthesis

計畫類別：個別型計畫
國科會計劃編號：NSC 97-2221-E-343-008
執行期限：97 年 8 月 1 日至 98 年 7 月 31 日
主持人：南華大學資工系 蔡加春教授

摘要：天線效應指的是在電漿奈米製程中，大量的電荷累積在連接閘極的金屬層上，導致介電層電場過強，而對於閘極造成退化破壞的現象，進而影響晶片的可靠度。本研究特別針對於以往從未在 X 架構上的繞線技術考慮天線效應的問題，試著提出以插入跳線與金屬層指定並行的解決方案，並且針對於若是應用於時脈繞線適用性作出詳細的探討。已知具有 n 個時脈端點之 X 架構時脈繞線、金屬層數及天線效應的臨界限，我們所提出的演算法稱為 PADJI 能夠以時間複雜度 $O(n^2)$ 插入跳線與金屬層指定來完成無天線效應及控制時脈偏移。以我們提出的方法測試所有時脈繞線的標準例子，實驗結果顯示出，相較於使用傳統上的插入跳線的方式來修正天線效應的問題，我們的方法將可以在插入跳線、時脈延遲、功率消耗及穿孔值分別節省 48.21%、0.0242%、0.0186%與 20.36%。

關鍵字：天線效應、時脈樹、可製造性設計、插入跳線、金屬層指定、X結構繞線

Abstract: Antenna effect is a phenomenon in the plasma-based nanometer process. The metal wire directly exposed to plasma may accumulate sufficient charges to damage thin gate oxide and furthermore influence chip reliability. This work proposes a discharge-path-based antenna detection and fixing with jumper insertion algorithm for X-architecture clock tree. The layer assignment technique is integrated with the proposed algorithm to reduce inserted jumpers. Given an X-clock tree with *n* clock sinks, layer configuration, and the upper bound of antenna effect, the proposed algorithm named PADJI can run in $O(n^2)$ to achieve antenna violation free and consider via timing impact caused by jumper insertion during skew compensation. In terms of number of inserted jumpers, clock delay, power consumption, and required total via cost listed in the experimental results on benchmarks, PADJI can fix antenna violations and achieves improvements of 48.21%, 0.0242%, 0.0186%, and 20.36%, respectively, compared with other jumper insertion method for X-clock routing.

Keywords: Antenna effect, clock tree, design for manufacturability, jumper insertion, layer assignment, X-architecture routing.

I. Introduction

With the continuous and rapid increasing complexity of very-large-scale-integration (VLSI) designs in present nanometer fabrication technologies, product reliability and manufacturing yield have become important issues in design and manufacture fields [1-2]. The fine feature size of modern IC technology is typically achieved by using plasma-based processes. In nanometer era, more stringent process requirements cause some advanced high-density plasma reactors applied in the production lines for achieving fine-line patterns. However, the plasma-based process contains a potential behavior to charge conducting components of fabricated structures. The announced experimental results report that the accumulated charges may affect the quality of thin gate oxide [3]. This phenomenon is called an antenna effect or a plasma-induced gate-oxide damage.

During wafer manufacturing, the metal wire not covered by a shielding layer of oxide is directly exposed to plasma and accumulates sufficient charges to affect the quality of thin gate oxide. Chen and Koren [4] proved that the amount of such charging is proportional to the plasma-exposed area and they also defined the ratio of plasma-exposed area, $A_{s,metal}$, to gate-oxide area, A_{poly}, as an antenna ratio, AR, to predict the antenna effect. The definition of AR is given as follows.

$$AR = \frac{\text{plasma - exposed area}}{\text{gate - oxide area}} = \frac{A_{s,metal}}{A_{poly}} \le k_{th}, \quad (1)$$

where k_{th} is the threshold of AR.

For a metal wire with width w_M, length, l_M, and thickness t_M, as shown in Fig. 1(a), $A_{s,metal}$ and A_{poly} are defined as $2(w_M+l_M)t_M$ and $w_M l_M$, respectively. According to TSMC 0.18μm technology file [5], the required k_{th} is 400 and the upper bound of antenna effect, L_{max}, can be derived as 200μm. If the length of a metal wire or the accumulated length of concatenate metal wires is longer than L_{max}, an antenna violation will be occurred. Thus, the upper bound could be a limited length, a limited exposed area, a ratio of antenna strength (length or area) to gate-oxide area, or a model of antenna strength caused by conductors.

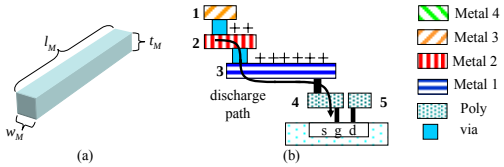


Fig. 1. (a) A metal wire and (b) a cross-section view for illustrating antenna effect.

Take the cross-section view shown in Fig. 1(b) as an example. The manufacturing process supports four metal layers and all the interconnects are manufactured in the order of poly, metal 1, metal 2, metal 3, and metal 4. Obviously, metal 3 is the topmost layer, L_{top}, in this structure, as shown in Fig. 1(b). The total exposed wire areas of the gate oxide are counted from poly (segment 4), metal 1 (segment 3) to metal 2 (segment 2) and may accumulate enough charges to damage the gate oxide with Fowler-Nordheim (F-N) tunnelling current through the discharge path [6-7]. The wire area of/after the topmost metal layer is irrelevant to the antenna consideration [6]. Hence, the accumulated length of gate oxide is the sum of the lengths of segments 2-4. If the accumulated length is longer than L_{max}, an antenna violation will be introduced.

In order to eliminate or reduce the number of damaged gates in the plasma-based processes and to ensure reliability of ICs, a routing rule considering antenna effect is derived. It follows the allowable maximum antenna size or antenna ration [4-5] to determine an antenna violation. Three popular solutions for fixing the antenna violation are introduced as follows.

1) Insert a jumper to break the wire which has antenna violation. This method can shorten the wire which collects charges during manufacturing but incurs extra two vias for each jumper for connecting higher layer, as shown in Fig. 2(a). The accumulated length of gate is the sum of the lengths of segments 4 and 5 after jumper insertion. Ho *et al.* [8-9] proposed an antenna avoidance full-chip routing with jumper insertion for spanning trees. Su and Chang [10-13] proposed an optimal greedy jumper insertion algorithm to minimize the number of jumpers in a spanning tree without/with obstacles. They gave the range of L_{max} as 50μm to 200μm for 90nm to 250nm CMOS technologies.

2) Assign the antenna-critical wire with the highest layer, as shown in Fig. 2(b). Wu *et al.* [14-15] applied layer assignment technique to fix antenna violations by a steiner-tree partition algorithm.

3) Embed a protection diode on each input port of a standard cell during layout design to prevent charge damage, as shown in Fig. 2(c). This solution may consume the cell area resources and increase manufacturing cost. Moreover, the diode is always embedded even the wire connecting the cell is antenna violation free. Huang *et al.* [16-17] transformed the antenna-critical wires, routing grids, and feasible positions for placing diodes into a flow network, and then, solved the diode insertion and routing problem with minimum-cost network-flow algorithm. Jiang and Chang [18-19] applied both diode and jumper insertions to achieve high antenna fixing rate due to limited routing resource in a chip. According to TSMC 0.25μm technology file, they gave L_{max} as 50μm and 100μm in the experiments.

In modern high-density VLSI routings, the diode insertion solution will cause layer congestion problem, increase extra capacitances on wires, and generate leakage power. Thus, the jumper insertion solution is more popular than the former one and guides the router or physical synthesis tools to achieve antenna violation free and higher yield.

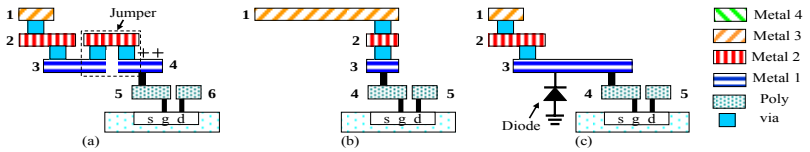


Fig. 2. Fix antenna violation with (a) jumper insertion, (b) layer assignment, and (c) embedded diode.

As VLSI fabrication technology gets into deep-submicron (DSM) era and faces process variation, interconnect delay gradually dominates chip performance. Hence, minimizing clock delay and clock skew of synchronize designs are important for preventing fault functions induced by different arrival times. Moreover, the antenna effect should also be considered during manufacturing clock tree. Many previous literatures [6-17] proposed their methods applying one or more of the above three solutions to fix antenna violations for general routings. Tsai *et al.* [20] first solved the antenna effect with jumper insertion and layer assignment methods for X-clock routing and named the proposed algorithm JILA. However, the approach detects all the antenna violations with the topmost layer of layer configuration not the discharge paths connecting gates. Hence, the number of antenna violations will be over-estimated and result unnecessary inserted jumpers, worse clock delay, more via cost, and extra power consumption.

Because inserted jumpers introduce extra vias and degrade the timing performance of clock tree, accurate antenna effect detection, effective jumper insertion, and compressed via timing impact are desired. This work proposes a discharge-path-based antenna effect detection algorithm for a given X-architecture clock tree and then inserts jumpers to fix antenna violations. Furthermore, a wire sizing technique is employed to maintain zero skew of X-clock tree after inserting jumpers with extra vias. Experimental results based on the clock routings generated from the X-architecture clock router in the study in [21] on three sets of benchmarks indicate that the proposed algorithm can significantly achieve more improvements in the number of inserted jumpers, clock delay, skew, via cost, and power consumption compared with JI and JILA [20], respectively.

II. Problem Formulation

In order to detect and fix antenna violations, the metal wires those not only connect gates but also collect charges are the discharge paths and should be visited. Next, the total lengths of discharge paths are compared with the upper bound of antenna effect, L_{max}. The objective of this work is to detect antenna violations with discharge paths in a given X-clock tree, and then, all the violations are fixed with minimum number of inserted jumpers. After inserting jumpers, a wire sizing technique is used to compensate clock skew. This work applies PMXF algorithm [21] to construct the initial clock tree based on X-architecture [22]. In the layer configuration of the adopted X-architecture, the horizontal and vertical wires are assigned with metal 1 (M1) and metal 2 (M2), as well as, ±45° wires are assigned with metal 3 (M3) and metal 4 (M4), respectively.

Figure 3 illustrates antenna effect and fixes antenna violation with jumper insertion. A part of clock tree based on X-architecture is shown in Fig. 3(a) and s_i and s_j are clock sinks. A sink is the clock input port of a standard cell or an IP (intellectual property). $P_B(s_i, s_j)$ is the bending point and one or two more vias are inserted at the point to connect different metal layers. $P_T(s_i, s_j)$ is the tapping point [23] of s_i and s_j for determining zero skew property of clock tree. Figure 3(b) is the cross-section view of Fig. 3(a) and two discharge paths, DP_1 and DP_2, through clock sinks, s_i and s_j, respectively, are presented.

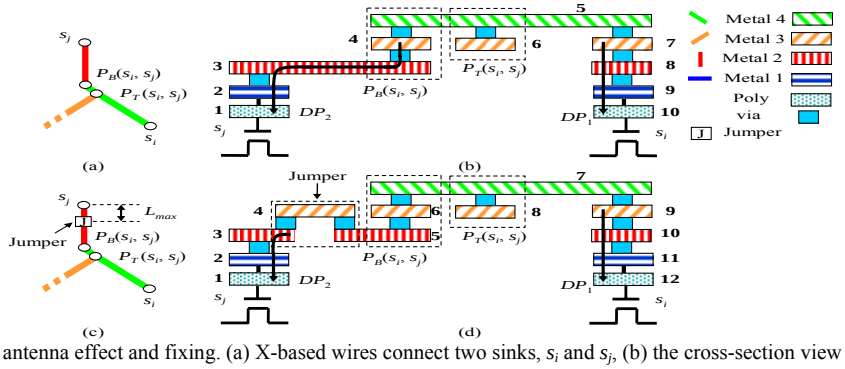


Fig. 3. Illustrations of antenna effect and fixing. (a) X-based wires connect two sinks, s_i and s_j, (b) the cross-section view of (a), (c) a jumper is inserted to fix antenna violation, and (d) the cross-section view of (c).

During metallization, chips are usually manufactured "layer-by-layer." When a metal interconnect is being assembled, it will contain several pieces of metals. Long interconnects act as temporary capacitors and accumulate charges gained from the energy provided during fabrication steps such as plasma etching or chemical mechanical polishing (CMP). When a long metal connects a thin gate oxide or clock sink, the accumulated charges will be discharged through the discharge path of clock sink. Meanwhile, if the length of DP is longer than L_{max}, the clock sink that DP flows through will be damaged. As shown in Fig. 3(b), DP_1 is the discharge path of s_i. The accumulated length of s_i, $L_{acu}(s_i)$, is the sum of the lengths of segments 9-12 due to metal 4 is the topmost layer, L_{top}, of DP_1. When a sink directly connects the topmost layer of discharge path, like s_i, this sink will not surfer antenna damage [9]. In contrast, s_j has another discharge path, DP_2, and $L_{acu}(s_j)$ is the sum of the lengths of segments 1-4 due to metal 4 is L_{top} of DP_2. Because $L_{acu}(s_j)$ is longer than L_{max}, s_j has antenna violation. Hence, this work applies the technique illustrated in Fig. 2(a) to insert a jumper on a wire connecting s_j to reduce the length of effective conductor associated with sink s_j. Fig. 3(c) shows that a jumper is inserted on the wire connecting s_j and $P_B(s_i, s_j)$ to break the original discharge path. Here, the distance between s_j and the inserted jumper is L_{max}. Figure 3(d) is the cross-section view of Fig. 3(c). After inserting a jumper, DP_2 can be shortened to the sum of the lengths of segments 1-3. Segments 5 and 6 will not collect charges due to they are floating [9]. As aforementioned, jumpers are implemented with vias which will consume extra routing resource and increase delay time. Therefore, it is desired to minimize the number of inserted jumpers during fixing antenna violations.

Based on the above discussion, this work visits each sink in the given X-clock tree and then determines how many sinks have antenna violations with their discharge paths. Next, jumpers are inserted on the metal wires to fix all the antenna violations and the number of inserted jumpers should be minimized. Thus, the problem of antenna detection and fixing for an X-clock routing can be formulated as follows.

Given an X-clock tree with a set of n clock sinks, $S=\{s_1, s_2, \ldots, s_n\}$, layer configuration, and the upper bound of antenna effect L_{max}, the objective of this work is to detect and fix antenna violations in the given X-clock tree with jumper insertion and maintain zero skew.

III. Delay Model of Metal Wire with Via Timing Impact Consideration

Elmore delay (ED) model [24] is widely applied to calculate wire delay for clock tree construction. The equivalent circuit of a wire e_i with width w_i and length l_i based on ED model is shown in Fig. 4(a), where r and c_a denote the sheet resistance and the unit area capacitance, respectively. Because ED model may overestimate wire delay and limit accuracy, fitted Elmore delay (FED) model [25] is alternately proposed. Figure 4(b) shows the equivalent circuit of e_i based on FED model, where c_f is the fringing capacitance.

The delay of e_i connecting a load capacitance, $C_{L,i}$, at sink i based on ED and FED models are respectively

formulated as follows.

$$Delay(i) = r(\frac{l_i}{w_i})\left(\frac{c_a w_i l_i}{2} + C_{L,i}\right) \text{ for ED model} \tag{2}$$

and

$$Delay(i) = r(\frac{l_i}{w_i})\left[\frac{(Dc_a w_i + Ec_f)l_i}{2} + FC_{L,i}\right] \text{ for FED model,} \tag{3}$$

where coefficients, D, E, and F, are obtained by curve fitting technique. The load capacitance, $C_{load,i}$, is defined as the capacitance of a tree rooted at sink i or node i. The $C_{load,i}$ for ED and FED models are respectively defined as bellow.

$$C_{load,i} = \begin{cases} \sum_{j \in T(i)} (c_a l_j w_j + C_{L,j}) \text{ for node } i \text{ based on ED model} \\ \sum_{j \in T(i)} \left[\frac{(Dc_a w_j + Ec_f)l_j}{F} + C_{L,j}\right] \text{ for node } i \text{ based on FED model,} \end{cases} \tag{4}$$

where $T(i)$ is the set of tree edges at the downstream of node i. Hence, the clock delay from *root* to sink k based on ED and FED models can be respectively calculated using

$$\text{Delay}(root, k) = \sum_{i \in P(k)} r(\frac{l_i}{w_i})\left[\frac{c_a l_i w_i}{2} + C_{load,i}\right] \text{ for ED model} \tag{5}$$

and

$$\text{Delay}(root, k) = \sum_{i \in P(k)} r(\frac{l_i}{w_i})\left[\frac{(Dc_a w_i + Ec_f)l_i}{2} + FC_{load,i}\right] \text{ for FED model,} \tag{6}$$

where $P(k)$ is the set of tree edges those are along the path from *root* to sink k. When a via is inserted at bending point to connect different metal layers, the via timing impact should be considered in clock delay calculation during clock tree construction. A via is modeled as a π-model circuit and both its resistance and capacitance are k-times of those of a wire. The bias k is a constant and set as 2 [26]. The equivalent circuits of vias based on ED and FED models are shown in Figs. 4(c) and 4(d), respectively, where n is the number of vias.

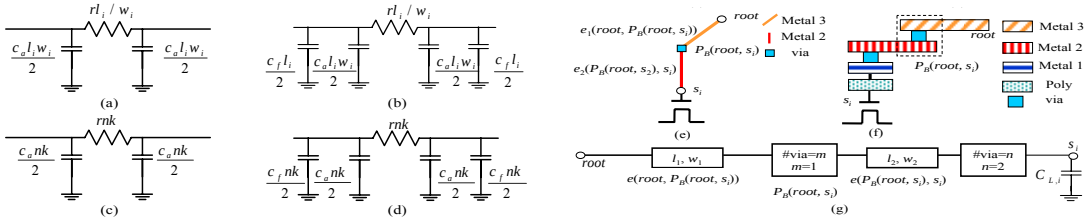


Fig. 4. The equivalent circuits of wire e_i are based on (a) ED and (b) FED models, respectively. The equivalent circuits of vias are based on (c) ED and (d) FED models, respectively. (e) X-based wires connect *root*, bending point, and sink s_i, (f) the cross-section view of (e), and (g) the equivalent model of (e).

Figure 4(e) shows that the wire $e_1(root, P_B(root, s_i))$ connects *root* and the bending point, $P_B(root, s_i)$, as well as, $P_B(root, s_i)$ connects s_i with $e_2(P_B(root, s_i), s_i)$. Figure 4(f) shows the cross-section view of Fig. 4(e) and demonstrates that one via is inserted at $P_B(root, s_i)$ and two vias are inserted at s_i, respectively. The equivalent model of Fig. 4(e) is shown in Fig. 4(g). Finally, the total delay of wires connecting *root* up to s_i with via timing impact consideration based on ED model is derived as follows.

$$\begin{aligned}
&Delay\ (root\ , s_i)\ \text{for ED model} \\
&= \sum_{i \in P(s_i)} r(\frac{l_i}{w_i})\left[\frac{c_a l_i w_i}{2} + C_{load\ ,\ i}\right] \\
&= mkr\ (\frac{mkc_{\ a}}{2} + C_{L,i}) + \frac{rl_2}{w_2}(\frac{c_a l_2 w_2}{2} + mkc_{\ a} + C_{L,i}) + nkr\ (\frac{nkc_{\ a}}{2} + c_a l_2 w_2 + mkc_{\ a} + C_{L,i}) + \frac{rl_1}{w_1}(\frac{c_a l_1 w_1}{2} + nkc_{\ a} + c_a l_2 w_2 + mkc_{\ a} + C_{L,i}) \\
&= \frac{rc_a}{2}\left[l_2^{\ 2} + 2nkl_{\ 2}w_2 + (m+n)^2 k^2 + \frac{2mkl_{\ 2}}{w_2} + l_1^{\ 2} + 2l_2 l_1 + \frac{2(m+n)kl_1}{w_1}\right] + \left[r(\frac{l_2}{w_2} + \frac{l_1}{w_1}) + (m+n)kr\right]C_{L,i} \\
&= \frac{rc_a}{2}\left[\frac{l}{w} + (m+n)k\right]\left[lw + (m+n)k\right] + r\left[\frac{l}{w} + (m+n)k\right]C_{L,i}\Big|_{w = w_1 = w_2,\ l = l_1 + l_2} \\
&= r\left[\frac{l}{w} + (m+n)k\right]\left\{\frac{c_a\left[lw + (m+n)k\right]}{2} + C_{L,i}\right\} \\
&= r(\frac{l}{w} + via_{\ i}k)\left[\frac{c_a(lw + via_{\ 2}k)}{2} + C_{L,i}\right]\Big|_{via_{\ i} = m+n},
\end{aligned} \tag{7}$$

where l_1 and w_1 are the length and width of $e_1(root, P_B(root, s_i))$, l_2 and w_2 are the length and width of $e_2(P_B(root, s_i), s_i)$, m and n are the number of inserted vias at $P_B(root, s_i)$ and s_i, respectively, and via_i is the total via count of s_i. The clock delay, $Delay(root, s_i)$, based on FED model is also derived as follows.

$$\begin{aligned}
&Delay\ (root\ , s_i)\ \text{for FED model} \\
&= \sum_{i \in P(s_i)} r(\frac{l_i}{w_i})\left[\frac{(Dc_{\ a}w_i + Ec_{\ f})l_i}{2} + FC_{\ load,i}\right] \\
&= r\left[\frac{l}{w} + (m+n)k\right]\left\{\frac{Dc_{\ a}\left[lw + (m+n)k\right] + Ec_{\ f}\left[l + (m+n)k\right]}{2} + FC_{\ L,i}\right\} \\
&= r(\frac{l}{w} + via_{\ i}k)\left[\frac{Dc_{\ a}(lw + via_{\ i}k) + Ec_{\ f}(l + via_{\ i}k)}{2} + FC_{\ L,i}\right]\Big|_{via_{\ i} = m+n}.
\end{aligned} \tag{8}$$

IV. Discharge-Path-Based Antenna Detection and Fixing with Jumper Insertion

In order to detect and fix antenna effects, the proposed algorithm is presented in Fig. 5 and named PADJI (discharge-path-based antenna effect detection and fixing with jumper insertion). For a given X-clock tree with a set of n clock sinks, $S=\{s_1, s_2, \ldots, s_n\}$, layer configuration, and the upper bound of antenna effect L_{max}, the objective of this work is to detect antenna effect with discharge paths and then fix antenna violations with minimum number of inserted jumpers and zero skew.

```
Algorithm: PADJI ()  /*Discharge-path-based effect antenna detection and fixing with jumper insertion */
Input: A given X-clock tree with a set of sinks, S, layer configuration, and the upper bound of antenna effect, Lmax
Output: A zero skew X-clock tree with antenna violation free and minimum number of inserted jumpers
1  for each si in S
2  { if( e(si).layer = the Ltop of layer configuration )
3    { si is antenna violation free. }
4    else
5    { DP(si) = DischargePath(si);  /* Obtain a set of discharge paths of si, DP(si) */
6      for each dpk in DP(si)
7      { etotal(si) = etotal(si) + dpk – Duplicate(∀em∈dpk, etotal(si));
         /* Obtain a set of segments those accumulate charges and may damage si. */
8        if( |etotal(si)| ≥ Lmax )
9        { InsertJumper(si);  /* Insert a jumper to fix antenna violation occurred on si. */
10         break;
11       }
12     }
13   }
14   WireSizing(si);  /* Adjust wire width to maintain zero skew. */
15 }
```

Fig. 5. The proposed PADJI algorithm.

The proposed PADJI algorithm consists of several procedures in a *for* loop, and its overview is given as follows. For each sink $s_i \in S$ in the given X-clock tree, if the layer of a metal wire connecting s_i is the top layer, L_{top}, of layer configuration, s_i is antenna violation free; otherwise, the proposed algorithm will detect each discharge path of s_i. The *DischargePath* procedure visits all the discharge paths those relate s_i and then obtains a set of discharge paths, $DP(s_i)$, which may collect charges to damage s_i. Next, the proposed algorithm moves the metal wires of each discharge path, dp_k, into the accumulated wires of s_i, $e_{total}(s_i)$. The length of $e_{total}(s_i)$, $|e_{\text{total}}(s_i)|$, is used to estimate the violation of antenna effect. Because each dp_k may contain several pieces of edges, e_m, the duplicate edge is excluded to avoid length overestimation and unnecessary jumper insertion. If $|e_{total}(s_i)|$ is equal to or longer than L_{max}, the *InsertJumper*(s_i) procedure will insert a jumper to fix the antenna violation of s_i. Finally, the *WireSizing* (s_i) procedure is employed to adjust the width of wire connecting s_i for skew compensation. After introducing the overview, each procedure will be described in the following subsections in detail.

Discharge-Path-based Antenna Detection — DischargePath()

For a pair of sinks s_i and s_j in a part of X-clock tree shown in Fig. 6(a), the *DischargePath* procedure visits all the discharge paths relating the given sink s_i and collects charges along the edges of s_i into a set of discharge paths $DP(s_i)$ for calculating the total accumulated lengths. Figure 6(b) illustrates the tree topology of Fig. 6(a) based on binary tree structure to model the X-clock tree. For calculating the number of inserted vias and estimating clock delays in an X-clock tree, this work adopts two via definitions proposed in [27]. When a via belongs to an internal node or a leaf node in a binary X-clock tree, the via is defined as a node via, NV, such as the square dots in Fig. 6(b). The number of NVs is equal to the difference of the highest and lowest metal layers. When a via locates at a bending point of a tree edge, the via is defined as an edge via, EV, such as the triangle dots in Fig. 6(b). The number of EVs is counted by the difference of metal layers assigned in the tree edge.

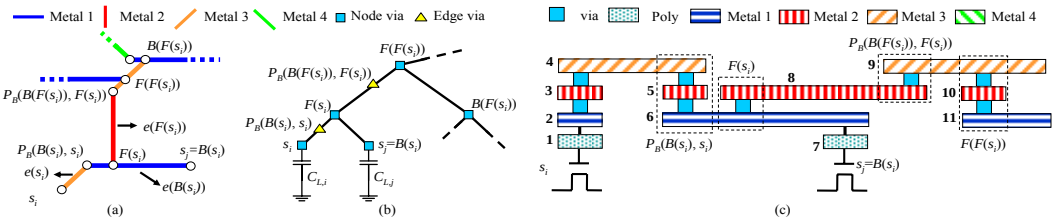


Fig. 6. (a) A part of X-clock tree, (b) the topology of binary X-clock tree with two via definitions, and (c) the cross-section view of (a).

Based on the properties of binary X-clock tree shown in Fig. 6(b), sink s_j is s_i's brother and denoted as $B(s_i)$. The tapping point of s_i and s_j is their father and denoted as $F(s_i)$. The bending point of s_i and s_j (also called $B(s_i)$) is denoted as $P_B(B(s_i),s_i)$. For applying discharge paths to detect antenna effect, this work also defines the connecting edges and their layer as follows. The edge connecting s_i and $P_B(B(s_i),s_i)$ is denoted as $e(s_i)$ and the layer of $e(s_i)$ is denoted as $e(s_i)$.layer; the edge connecting $B(s_i)$ and $P_B(B(s_i),s_i)$ is denoted as $e(B(s_i))$ and the layer of $e(B(s_i))$ is denoted as $e(B(s_i))$.layer; the edge connecting $F(s_i)$ and its bending point $P_B(B(F(s_i)),F(s_i))$ is denoted as $e(F(s_i))$. The layer of $e(F(s_i))$ is denoted as $e(F(s_i))$.layer. Figure 6(c) is the cross-section view of Fig. 6(a). Because $e(s_i)$.layer is M3, the number of NVs locating on s_i is three for connecting poly layer up to M3. Moreover, the number of EVs locating on $P_B(B(s_i), s_i)$ is two for connecting M1 up to M3 due to $e(B(s_i))$.layer and $e(s_i)$.layer are M1 and M3 respectively.

After introducing the adopted notations, the proposed *DischargePath* procedure is presented in Fig. 7. For a given sink s_i, the top layer of s_i, $L_{top}(s_i)$, is first determined by the maximum layer of $e(s_i)$.layer and $e(B(s_i))$.layer. Then, three cases for s_i are derived as follows for getting the set of discharge paths, $DP(s_i)$ in the procedure.

Case1 $e(B(s_i))$.layer < $L_{top}(s_i)$ (lines 2-7): In the case, the charges accumulated on $e(B(s_i))$ will not affect s_i. Hence, the procedure have to check the layer and location of $F(s_i)$. If $e(F(s_i))$.layer is higher than $L_{top}(s_i)$ and $F(s_i)$ locates on $e(s_i)$, $e(s_i)$ will be inserted into dp_k of $DP(s_i)$. In addition, the exceptional s_i is antenna violation free.

Case2 $e(s_i)$.layer = $e(B(s_i))$.layer (lines 8-15): In the case, both $e(s_i)$ and $e(B(s_i))$ may collect charges to damage s_i and the procedure just checks the layer of $F(s_i)$. If $e(F(s_i))$.layer is higher than $L_{top}(s_i)$, $e(s_i)$ and $e(B(s_i))$ will be inserted into dp_k of $DP(s_i)$. Or, if $e(F(s_i))$.layer equals $L_{top}(s_i)$, the number of effective conductors will be increased and collect more charges. Hence, the procedure not only inserts $e(s_i)$ and $e(B(s_i))$ into dp_k of $DP(s_i)$ but also gives $F(s_i)$ to the *DischargePath* procedure for determining extra edges. In addition, the exceptional s_i is antenna violation free.

Case3 $e(B(s_i))$.layer > $L_{top}(s_i)$ (lines 16-27): This case can also be viewed as that $L_{top}(s_i)$ is equal to $e(B(s_i))$.layer. When $e(F(s_i))$.layer is higher than $L_{top}(s_i)$, the procedure will check the location of $e(F(s_i))$. If $F(s_i)$ locates on $e(s_i)$, only $e(s_i)$ will be inserted into dp_k of $DP(s_i)$. Otherwise, both $e(s_i)$ and $e(B(s_i))$will be inserted into dp_k of $DP(s_i)$. When $e(F(s_i))$.layer satisfies the constrain, [$e(s_i)$.layer, $L_{top}(s_i)$) and $F(s_i)$ locates on $e(s_i)$, the procedure not only inserts $e(s_i)$ into dp_k of $DP(s_i)$ but also gives $F(s_i)$ to the *DischargePath* procedure for determining extra edges. Here, the constrain is an interval of $e(F(s_i))$.layer, as well as, the internal means that $e(F(s_i))$.layer is higher than or equal to $e(s_i)$.layer and $e(F(s_i))$.layer is lower than $e(B(s_i))$.layer. In addition, only $e(s_i)$ will be inserted into dp_k of $DP(s_i)$ for the exceptional s_i.

```
Procedure: DischargePath(s_i)
Input: A sink or node s_i
Output: A set of discharge paths of s_i
1  L_top(s_i) = Max( e(s_i).layer, e(B(s_i)).layer );  /* Determine the L_top of s_i. */
2  if( e(B(s_i)).layer < L_top(s_i) ) /* Case 1 */
3  {  if( e(F(s_i)).layer > L_top(s_i) and F(s_i) locates on e(s_i) )
4     {  Insert e(s_i) into dp_k of DP(s_i);  }
5     else
6     {  s_i is antenna violation free.  }
7  }
8  else if( e(s_i).layer = e(B(s_i)).layer )  /* Case 2 */
9  {  if( e(F(s_i)).layer > L_top(s_i) )
10    {  Insert e(s_i) and e(B(s_i)) into dp_k of DP(s_i);
11    else if( e(F(s_i)).layer = L_top(s_i) )
12    {  Insert e(s_i), e(B(s_i)), and DischargePath(F(s_i)) into dp_k of DP(s_i);  }
13    else
14    {  s_i is antenna violation free.  }
15 }
16 else  /* Case 3 */
17 {  if( e(F(s_i)).layer > L_top(s_i) )
18    {  if( F(s_i) locates on e(s_i) )
19       {  Insert e(s_i) into dp_k of DP(s_i);
20       else
21       {  Insert e(s_i) and e(B(s_i)) into dp_k of DP(s_i);  }
22    }
23    else if( e(s_i).layer ≤ e(F(s_i)).layer < L_top(s_i) and F(s_i) locates on e(s_i) )
24    {  Insert e(s_i) and DischargePath(F(s_i)) into dp_k of DP(s_i);  }
25    else
26    {  Insert e(s_i) into dp_k of DP(s_i);  }
27 }
```

Fig. 7. The proposed *DischargePath* procedure for determining a set of discharge paths of the given sink.

For explaining the proposed *DischargePath* procedure clearly, a simple case is illustrated. Figure 8(a) shows a part of initial X-clock tree without antenna effect consideration, where s_9, s_{10}, s_{13}, and s_{14} are sinks, s_{21}, s_{23}, and s_{27} are the tapping points of (s_9, s_{10}), (s_{13}, s_{14}), and (s_{21}, s_{23}), respectively. Here, s_{13} is taken as an example and first given to the *DischargePath* procedure to detect antenna violation. Meanwhile, s_{14} and s_{23} are respectively denoted as $B(s_{13})$ and $F(s_{13})$ due to s_{14} is paired with s_{13} and s_{23} is the tapping point of (s_{13}, s_{14}). As shown in Fig. 8(a), $e(s_{13})$.layer is M1 and $e(B(s_{13}))$.layer is M3, *DischargePath* procedure can determine that $L_{top}(s_{13})$ is M3 and s_{13} satisfies the condition of Case3. Next, $e(F(s_{13}))$.layer is M2 and $F(s_{13})$ not only satisfies the interval constrain of [M1, M3) but also locates on $e(s_{13})$. Hence, the procedure inserts $e(s_{13})$ into dp_1 of $DP(s_{13})$ and gives $F(s_{13})$ to the *DischargePath* procedure for determining extra edges. In the procedure *DischargePath*($F(s_{13})$) (or *DischargePath*(s_{23})), $e(s_{23})$.layer is M2, $e(B(s_{23}))$.layer is M4, and $e(F(s_{23}))$.layer is M1. Therefore, *DischargePath*($F(s_{13})$) only returns $e(s_{23})$. Finally, $DP(s_{13})$ contains $e(s_{13})$ and $e(s_{23})$. For the other sink s_{14}, $e(s_{14})$.layer is M3, $e(B(s_{14}))$.layer is M1, and $F(s_{14})$ does not locates on $e(s_{14})$. Therefore, s_{14} satisfies the condition of Case1 and has no antenna violation. The antenna effects of other two sinks s_9 and s_{10} can be detected by the same process. Figure 8(b) shows that two jumpers are inserted on $e(s_{13})$ and $e(s_{10})$ to fix the antenna violations, respectively. Note that the distance between the antenna-critical sink s_{13} (or s_{10}) and the inserted jumper is L_{max}.

In order to verify the proposed procedure of *DischargePath*, Figure 8(c) shows the cross-section view of Fig. 8(a). The discharge path set of s_{13}, $DP(s_{13})$, contains two discharge paths, dp_1 and dp_2. The length of dp_1 in $DP(s_{13})$ is the sum of the lengths of segments 5-7, as well as, the length of dp_2 in $DP(s_{13})$ is the sum of the lengths of segments 6-8,

and 10. Here, the length of each segment is X-architecture distance not Manhattan distance. Therefore, $|e_{total}(s_{13})|$ is the sum of the lengths of segments 5-8 and 10 whereas the duplicate segments have been removed. Moreover, the lengths of segments 5, 7, and 10 can be ignored due to they are metallised for vias. Finally, $|e_{total}(s_{13})|$ can be simplified as the sum of the lengths of segments 6 and 8. Note that the lengths of segments 6 and 8 are equal to $|e(s_{13})|$ and $|e(s_{23})|$, respectively. For the other sink s_{14}, it directly connects M3 which is the L_{top} of dp_1 in $DP(s_{14})$ and has no antenna violation. Because $|e_{total}(s_{13})|$ is longer than L_{max} and antenna violation is occurred, a jumper is inserted to break the original segment 6 shown in Fig. 8(c) into two segments 6 and 9 shown in Fig. 8(d). It is obvious that the length of $|e_{total}(s_{13})|$ is short enough to avoid antenna violation. For sinks s_9 and s_{10}, another jumper is also inserted on $e(s_{10})$ to fix antenna violation.

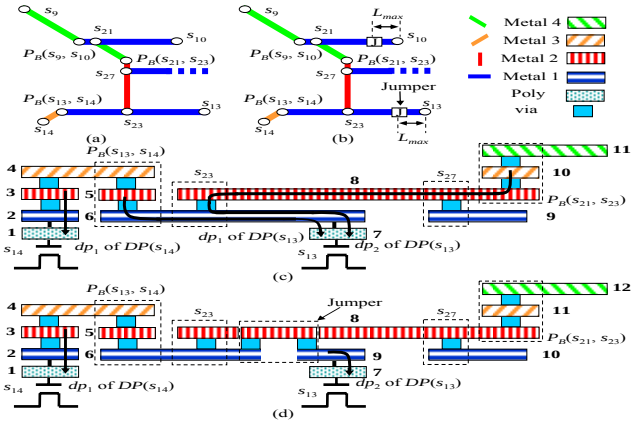


Fig. 8. (a) A part of initial X-clock tree without antenna violation consideration, (b) two jumpers are respecitvely inserted to fix the antenna violations of (a), (c) the cross-section view of (a), and (d) the cross-section view of (b) with jumper insertion.

Adjust Wire width after Jumper Insertion — WireSizing()

After inserting jumpers to fix antenna violations, clock skew is another occurred side-effect in X-clock tree. Hence, the proposed algorithm applies a wire sizing technique to achieve zero skew. As shown in Fig. 9(a), a pair of sinks (s_{13}, s_{14}), their tapping point s_{23}, and an inserted jumper are taken as an example to explain the *WireSizing* procedure. After inserting a jumper to fix the antenna violation of s_{13}, the number of required vias of s_{13}, $\#via_{13}$, is increased from 1 to 3 (see the physical structure in Fig. 8(d)). via_{13} includes one NV on s_{13} and two vias belong to the inserted jumper. Therefore, the wire $e(s_{23}, s_{13})$ connecting the tapping point s_{23} and sink s_{13} should be sized to achieve zero skew.

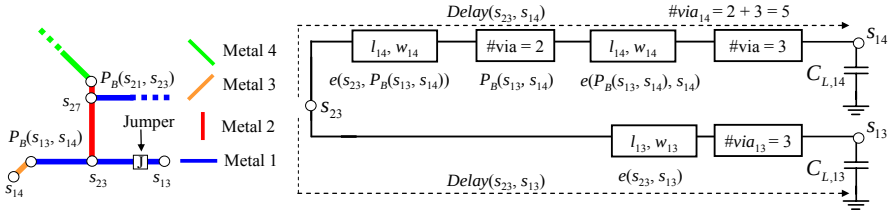


Fig. 9. (a) A part of X-clock tree with jumper insertion and (b) the equivalent model of (a) for determining the width of wire $e(s_{23}, s_{13})$.

Fig. 9(b) shows the equivalent model of Fig. 9(a) for explaining how to derive the width of $e(s_{23}, s_{13})$. First, s_{14} is antenna violation free without jumper insertion, therefore, the number of vias of s_{14}, $\#via_{14}$, is fixed and can be counted as the sum of three NVs on sink s_{14} and two EVs on bending point. Meanwhile, $Delay(s_{23}, s_{14})$ based on the ED model can be calculated by (7). Next, we formulate $Delay(s_{23}, s_{13})$ and force the two delays to be equal to each other for deriving the width of $e(s_{23}, s_{13})$, w_{13}, as follows.

$$Delay\ (s_{23}, s_{13}) = Delay\ (s_{23}, s_{14}),\ \text{for ED model} \tag{9}$$

$$r\left(\frac{l_{13}}{w_{13}} + via_{13}k\right)\left[\frac{c_a(l_{13}w_{13} + via_{13}k)}{2} + C_{L,13}\right] - Delay\ (s_{23}, s_{14}) = 0$$

$$\frac{rvia_{13}kc_a l_{13}}{2} w_{13} + \left(\frac{c_a via_{13}k}{2} + C_{L,13}\right)\frac{rl_{13}}{w_{13}} + \left[\frac{rc_a(via_{13}k)^2}{2} + \frac{rc_a l_{13}^2}{2} + rvia_{13}kC_{L,13} - Delay\ (s_{23}, s_{13})\right] = 0$$

$$\frac{rvia_{13}kc_a l_{13}}{2} w_{13}^2 + \left\{r\left[\frac{c_a l_{13}^2}{2} + via_{13}k\left(\frac{c_a via_{13}k}{2} + C_{L,13}\right)\right] - Delay\ (s_{23}, s_{14})\right\} w_j + \left[rl_{13}\left(\frac{c_a via_{13}k}{2} + C_{L,13}\right)\right] = 0$$

$$w_{13} = \left(-b \pm \sqrt{b^2 - 4ac}\right) / 2a,$$

where $a = \dfrac{rvia_{13}kc_a l_{13}}{2}$

$$b = r\left[\frac{c_a l_{13}^2}{2} + via_j k\left(\frac{c_a via_{13}k}{2} + C_{load,13}\right)\right] - Delay\ (s_{23}, s_{13})$$

$$c = rl_{13}\left(\frac{c_a via_{13}k}{2} + C_{load,13}\right).$$

In the study of [20], the JI (Jumper Insertion) algorithm focuses on antenna effect fixing with jumper insertion for X-clock tree and adopts FED model for calculating delay with the timing impact consideration of inserted vias. Here, *Delay*(s_{23}, s_{13}) and *Delay*(s_{23}, s_{14}) based on FED model can be respectively formulated and calculated, as well as, w_{13} can be also derived as (10).

$$\begin{aligned} &Delay(s_{23}, s_{13}) = Delay(s_{23}, s_{14}), \text{ for FED model} \\ &r(\frac{l_{13}}{w_{13}} + via_{13}k)\left[\frac{Dc_a(l_{13}w_{13} + via_{13}k) + Ec_f(l_{13} + via_{13}k)}{2} + FC_{L,13}\right] = Delay(s_{23}, s_{14}) \\ &w_{13} = \left(-b \pm \sqrt{b^2 - 4ac}\right)/2a, \end{aligned} \tag{10}$$

where

$$a = \frac{rvia_{13}kDc_a l}{2}$$

$$b = r\left[\frac{Dc_a l + Ec_f via_{13}k}{2} l + via_{13}k(\frac{Dc_a + Ec_f}{2} via_{13}k + FC_{L,13})\right] - Delay(s_{23}, s_{14})$$

$$c = rl\left[\frac{Dc_a via_{13}k + EC_f(l + via_{13}k)}{2} + C_{L,13}\right].$$

Power Estimation

This work estimates the power consumption for a whole clock tree by charging each equivalent wire, e_i, and the capacitance of each sink i or node i, $C_{load,\,i}$. The total power consumption in a clock tree is represented as follows.

$$\text{Power} = \sum_{\forall e_i} C_{load,\,i} F_{clk} V_{dd}^{\,2} \tag{11}$$

For comparative study, PMXF [21] algorithm is first applied to construct an initial X-clock tree without antenna effect consideration as shown in Fig. 10(a). The X-clock tree contains a set of 16 clock sinks, $S = \{s_1, s_2, \ldots, s_{16}\}$, dumped from the IBM benchmark *r*1 [30]. Next, JI [20] and the proposed PADJI algorithms respectively load the initial X-clock tree to determine the accumulated length of each sink for detecting antenna violations. Here, FED model with 130nm fabrication parameters is used to calculate delay and the upper bound of antenna effect L_{max} is given as 200μm. Finally, JI and PADJI insert jumpers to fix all the antenna violations in the initial X-clock tree, as shown in Figs. 10(b) and 10(c), respectively.

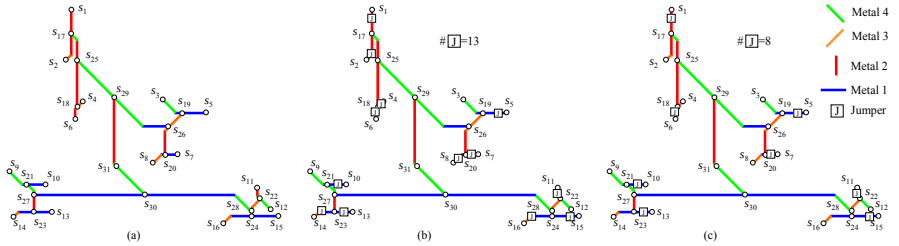


Fig. 10. (a) The initial 16-sink X-clock tree constructed by PMXF, as well as, antenna violation fixing by (b) JI [20] and (c) PADJI algorithms.

Table I lists the comparison of JI and PADJI algorithms in the accumulated lengths of each sink, $|e_{total}(s_i)|$. Columns 1, 2, 8 and 9 give the labels and coordinates of each sink and node, respectively. Column 3 gives the load capacitance, C_L, of each sink. Columns 4 and 5 and the last two columns give the labels and coordinates of bending point, $P_B(B(s_i),s_i)$, and tapping point, $F(s_i)$, of each pair of sinks or nodes. Note that the accumulated length of each sink, $|e_{total}(s_i)|$, determined by JI and PADJI are listed in columns 6 and 7, respectively. As shown in Table I, s_7 and s_8 are a pair of sinks, as well as, $P_B(s_7,s_8)$ and s_{20} are their bending point and tapping point, respectively. JI algorithm obtains that both s_7 and s_8 violate antenna rule, but the proposed PADJI algorithm just obtains one. This work will demonstrate the difference later. Finally, JI algorithm detected that there are 13 accumulated lengths of sinks longer than L_{max}. In contrast, PADJI detected eight.

TABLE I THE COORDINATES, BENDING POINTS, TAPPING POINTS, AND ACCUMULATED LENGTHS OF THE 16 SINKS AND OTHER NODES IN THE GIVEN X-CLOCK TREE BASED ON 130NM FED MODEL AND L_{MAX}=200μM.

Sink	Coordinate	C_L(pF)	$P_B(B(s_i), s_i)$	$F(s_i)$	$\|e_{total}(s_i)\|$ (μm)		Node	Coordinate	$P_B(B(s_i), s_i)$	$F(s_i)$
					JI	PADJI				
s_1	(18596.0, 36216.0)	0.078	$P_B(s_1, s_2)$ (18596.0, 29879.0)	s_{17} (18596.0, 32834.6)	6337.0	6337.0	s_{17}	(18596.0, 32834.6)	$P_B(s_{17}, s_{18})$ (19444.0, 31986.6)	s_{25} (19444.0, 29044.8)
s_2	(17841.0, 29124.0)	0.035			1067.7	0.0	s_{18}	(19444.0, 22366.0)		
s_4	(20313.0, 23235.0)	0.078	$P_B(s_4, s_6)$ (19310.0, 22232.0)	s_{18} (19444.0, 22366.0)	1418.4	0.0	s_{19}	(34327.3, 21556.0)	$P_B(s_{19}, s_{20})$ (31950.0, 19178.7)	s_{26} (32445.4, 19674.1)
s_6	(19310.0, 20716.0)	0.035			1516.0	1516.0	s_{20}	(31950.0, 15681.0)		
s_3	(31657.0, 23412.0)	0.046	$P_B(s_3, s_5)$ (33513.0, 21556.0)	s_{19} (34327.3, 21556.0)	0.0	0.0	s_{21}	(12226.2, 11439.0)	$P_B(s_{21}, s_{23})$ (13276.8, 10388.4)	s_{27} (13276.8, 10004.2)
s_5	(37908.0, 21556.0)	0.037			4395.0	4395.0	s_{23}	(13276.8, 7520.0)		
s_7	(33860.0, 15681.0)	0.044	$P_B(s_7, s_8)$ (31749.0, 15681.0)	s_{20} (31950.0, 15681.0)	2111.0	5608.7	s_{22}	(45633.5, 9408.5)	$P_B(s_{22}, s_{24})$ (44404.8, 8179.9)	s_{28} (44404.8, 7608.0)
s_8	(30550.0, 14482.0)	0.045			1695.6	0.0	s_{24}	(44404.8, 6943.0)		
s_9	(10014.0, 13282.0)	0.058	$P_B(s_9, s_{10})$ (11857.0, 11439.0)	s_{21} (12226.2, 11439.0)	0.0	0.0	s_{25}	(19444.0, 29044.8)	$P_B(s_{25}, s_{26})$ (28814.6, 19674.1)	s_{29} (24724.4, 23764.3)
s_{10}	(14975.0, 11439.0)	0.075			3118.0	3118.0	s_{26}	(32445.4, 19674.1)		

s_{13}	(15850.0, 7520.0)	0.040	$P_B(s_{13}, s_{14})$	s_{23}	4778.0	7646.4	s_{27}	(13276.8, 10004.2)	$P_B(s_{27}, s_{28})$	s_{30}
s_{14}	(10822.0, 7270.0)	0.041	(11072.0, 7520.0)	(13276.8, 7520.0)	353.6	0.0	s_{28}	(44404.8, 7608.0)	(42008.6, 10004.2)	(29162.4, 10004.2)
s_{11}	(45204.0, 10920.0)	0.067	$P_B(s_{11}, s_{12})$	s_{22}	1082.0	1082.0	s_{29}	(24724.4, 23764.3)	$P_B(s_{29}, s_{30})$	s_{31}
s_{12}	(47147.0, 7895.0)	0.032	(45204.0, 9838.0)	(45633.5, 9408.5)	0.0	0.0	s_{30}	(29162.4, 10004.2)	(24724.4, 14442.2)	(24760.9, 14405.6)
s_{15}	(48326.0, 6943.0)	0.064	$P_B(s_{15}, s_{16})$	s_{24}	6913.0	8149.8	s_{31}	(24760.9, 14405.6)	-	-
s_{16}	(40398.0, 5928.0)	0.032	(41413.0, 6943.0)	(44404.8, 6943.0)	1435.4	0.0				
				#Violations	13	8				

In order to clearly demonstrate the difference between JI and PADJI algorithms, this work takes two sinks s_7 and s_8 extracted from Fig. 10 as an example and presents the local views and cross-section views of X-clock routings constructed and fixed by PMXF, JI, and PADJI, respectively in Fig. 11. Figure 11(a) shows the local view of a pair of sinks, (s_7, s_8), in the initial X-clock tree constructed with PMXF and its cross-section view is shown in Fig. 11(d).

Here, JI algorithm is first discussed as follows. When a sink connects the L_{top} of layer configuration, JI thinks that the sink is antenna violation free. In contrast, if the sum of total lengths of wires connecting a sink is longer than L_{max}, JI will insert a jumper connecting the L_{top} of layer configuration to fix the violation. In this example, JI takes M4 as L_{top}. In Fig. 11(d), JI derives that the accumulated length of s_7, $|e_{total}(s_7)|$, is the sum of the lengths of segments 5-7. Because segments 5 and 7 are short enough to be ignored, Table I lists that $|e_{total}(s_7)|$ is approximated as the length of segment 6 as 2110μm. Due to $|e_{total}(s_7)|$ is longer than L_{max}, JI inserts a jumper to fix the antenna violation of s_7, as shown in Fig. 11(b). Moreover, the cross-section view of Fig. 11(b) is presented in Fig. 11(e). The jumper inserted on s_7 can shorten the length of $|e_{total}(s_7)|$ but need three vias to connect segments 8 and 11, as well as, other three vias to connect segments 11 and 9, respectively. For the other sink s_8, $|e_{total}(s_8)|$ is the sum of the lengths of segments 1-4 and approximated as the length of segment 4 as 1695.6μm, as listed in Table I. JI also inserts another jumper to fix the antenna violations of s_8, due to $|e_{total}(s_8)|$ is longer than L_{max}. As shown in Fig. 11(e), the jumper inserted on segment 4 can also shorten the length of $|e_{total}(s_8)|$ but need two extra vias to connect segments 4-5 and segments 5-6, respectively. Finally, JI totally inserts two jumpers to fix antenna violations and consumes extra eight vias for the pair of sinks (s_7,s_8).

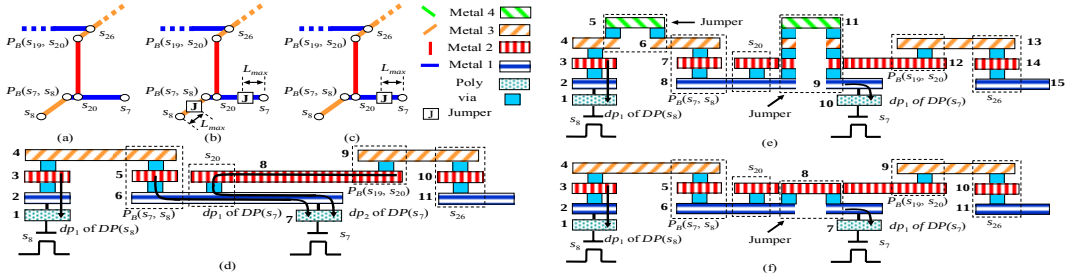


Fig. 11. Take s_7 and s_8 as an example. (a) The local view of X-clock tree with antenna violations, (b) two jumpers required by JI are inserted to fix antenna violations of (a), (c) only one jumper is required and inserted by PADJI. (d) The cross-section view of (a), where s_7 and s_8 respectively contain two and one discharge path, (e) the cross-section view of (b), and (f) the cross-section view of (c).

Second, the proposed PADJI algorithm is discussed as follows. As shown in Fig. 11(d), sinks s_7 and s_8 respectively have two and one discharge path. For sink s_7, the first discharge path of s_7, dp_1 of $DP(s_7)$, consists of segments 5-7 and the second one, dp_2 of $DP(s_7)$, consists of segments 6-8. Segment 8 connecting s_{20} and $P_B(s_{19},s_{20})$ may collect charges to damage s_7. However, the segment is not visited and included by JI when detecting s_7. For sink s_8, the top layer of segments connecting s_8 is M3 and the discharge path of s_8, dp_1 of $DP(s_8)$, only contains segments 1-3. Hence, segment 4 is overestimated by JI when detecting s_8. Moreover, the jumper inserted on s_8 should be removed. As aforementioned, that is the reason PADJI detects each discharge path of a given sink to precisely obtain its accumulated length for determining antenna violation.

For a paired of sinks s_7 and s_8, and s_{20}, their bending point and tapping point $s_{20}$are respectively denoted as $B(s_7)$ and $F(s_7)$. As shown in Fig. 11(a), $e(s_7)$.layer is M1 and $e(B(s_7))$.layer is M3, hence, the *DischargePath* procedure in PADJI can obtain that $L_{top}(s_7)$ is M3 and s_7 satisfies the condition of Case3. Next, $e(F(s_7))$.layer is M2 and $F(s_7)$ not only satisfies the interval constraint of [M1, M3) but also locates on $e(s_7)$. Hence, the procedure inserts $e(s_7)$ into $DP(s_7)$ and gives $F(s_7)$ to the *DischargePath* procedure for determining extra edges. In the *DischargePath*($F(s_7)$) (or *DischargePath*(s_{20})) procedure, $e(s_{20})$.layer is M2, $e(B(s_{19}))$.layer is M3, and $e(F(s_{26}))$.layer is M1. Therefore, s_{20} also satisfies the condition of Case3 and *DischargePath*($F(s_{20})$) returns $e(s_{20})$. Therefore, PADJI obtains that $DP(s_7)$ contains $e(s_7)$ and $e(s_{20})$, as well as, $|e_{total}(s_7)|$ can be approximated as the sum of $|e(s_7)|$ and $|e(s_{20})|$ as 5607.7μm. Here, $e(s_7)$ and $e(s_{20})$ are segments 6 and 8, respectively. For sink s_8, the *DischargePath* procedure in PADJI gets that s_8 satisfies the condition of Case1 and has no antenna violation. Finally, PADJI inserts only one jumper to fix the antenna violation of s_7 as shown in Fig. 11(c). The cross-section view of Fig. 11(c) is presented in Fig. 11(f) that the inserted jumpers just need two extra vias for connecting M2. Compared with JI algorithm, the proposed PADJI algorithm takes less inserted jumpers and vias to fix antenna violations.

Table II lists the experimental results on the 16-sink X-clock trees respectively constructed and fixed by PMXF, JI, and PADJI algorithms in terms of the number of inserted jumpers, delay, skew, power, and total vias. Note that the

ratio is defined as JI/PMXF or PADJI/PMXF, as well as, total vias is the sum of node vias and edge vias. The upper bound of antenna effect, L_{max}, is given as 200μm. In the table, JI inserts five more jumpers than PADJI to fix antenna violations. For the column "Skew" in Table II, the applied PMXF algorithm can construct a zero skew X-clock tree but ignore via timing impact. Hence, this work considers the issue during X-clock construction with PMXF and then results non-zero skew in the initial X-clock tree. In addition, this work applies the wire sizing technique to achieve zero skew. As listed in Table II, applying discharge path to detect antenna violation can avoid over-estimation and unnecessary inserted jumpers. It is not surprising that PADJI has better performance of 0.0604%, 0.048%, and 39.1304% than JI in terms of delay, power, and total vias, respectively.

TABLE II COMPARISON OF PMXF [21], JI [20], AND THE PROPOSED PADJI ALGORITHM WITH L_{MAX}=200μM ON THE 16-SINK X-CLOCK TREES.

#Sinks	#Inserted Jumper		Delay (μs)			Skew (μs)			Power (W)			Total vias		
	JI	PADJI	PMXF	JI	PADJI	PMXF	JI	PADJI	PMXF	JI	PADJI	PMXF	JI	PADJI
16	13	8	0.021500	0.021518	0.021505	0.000009	0.000011	0	0.002083	0.002085	0.002084	92	144	108
Ratio	-	-	-	1.000837	1.000233	-			-	1.000960	1.000480	-	1.565217	1.173913

Time Complexity Analysis

For a given X-clock tree with a set of sinks, this work applies binary tree structure to model an X-clock tree with antenna violation free. Since every sink in the tree will be examined by the proposed PADJI algorithm, the *for* loop runs in $O(n)$. In the *for* loop, the *DischargePath* procedure obtains a set of discharge paths for each sink and the worst case is that each sink contains k paths. Thus, the procedure runs in $O(k)$. The other processes in the *for* loop can insert a jumper to fix antenna violation with *InsertJumper* procedure and adjust wire width to obtain zero skew with *WireSizing* procedure in constant time. Finally, the time complexity of the proposed PADJI algorithm is $O(n^2)$.

V. Integration of PADJI and Layer Assignment Technique

Layer assignment technique [14-15] is another method for fixing antenna violations. If a wire segment is antenna-critical, it has to be assigned with the higher layer or the L_{top} of layer configuration to avoid antenna effect. For instance, the initial X-clock tree shown in Fig. 11(a) can be totally routed with single layer M4 which is the L_{top} of layer configuration for antenna avoidance. Each sink needs four NVs to connect poly layer up to M4, as well as, the number of EVs is zero due to all the edges connecting sinks or nodes are placed at the same layer. Therefore, layer assignment technique can obtain the optimal X-clock routing with antenna violation free and just totally need 64 vias which is much less than those, 144 and 108 vias, of JI and PADJI algorithms, respectively. However, the routing solution is not practical enough, especially for modern complex multi-level VLSI designs. Hence, this work proposes another layer assignment procedure. The approach can provide several layer candidates for an antenna-critical sink and be integrated with the proposed PADJI algorithm to insert jumpers when layers are congested. The congestion is the ratio of the number of wires to the number of tracks in a region. The proposed algorithm which integrates PADJI and layer assignment technique is presented in Fig. 12 and named PADJILA (discharge-path-based antenna detection and fixing with jumper insertion and layer assignment). For a given X-clock tree with a set of n clock sinks, $S = \{s_1, s_2, \ldots, s_n\}$, layer configuration, and the upper bound of antenna effect, L_{max}, the objective of the proposed PADJILA algorithm is not only fixing antenna violations in the given X-clock tree but also minimizing the number of inserted jumpers and required vias.

Algorithm: ***PADJILA***() /**Discharge-path-based antenna effect detection and fixing with jumper insertion and layer assignment */*
Input: *A given X-clock tree with a set of sinks, S, layer configuration, and the upper bound of antenna effect, L_{max}*
Output: *A zero skew X-clock tree with antenna violation free and minimum number of inserted jumpers*
1 for each s_i in S
2 { **if**($e(s_i)$.layer = the L_{top} of layer configuration)
3 { s_i is antenna violation free. }
4 **else**
5 { $DP(s_i)$ = ***DischargePath***(s_i); /* Obtain a set of discharge paths of s_i, $DP(s_i)$ */
6 for each dp_i in $DP(s_i)$
7 { $e_{total}(s_i) = e_{total}(s_i) + dp_i - Duplicate(\forall e_i \in dp_i, e_{total}(s_i))$;
/* Obtain a set of segments those accumulate charges and may damage s_i. */
8 **if**($|e_{total}(s_i)| \geq L_{max}$)
9 { ***InsertJumper***(s_i); /* Insert a jumper to fix antenna violation occurred on s_i. */
10 $via_{original}$ = #via(s_i, $B(s_i)$); /* Store #vias before launching *LayerAssignment*(s_i) */
11 $Layer_{candidates}(s_i)$ = ***LayerAssignment***(s_i); /* Obtain the layer candidates of $e(s_i)$.layer */
12 $via_{modified}$ = #via(s_i, $B(s_i)$);
/* Count #vias when assign $Layer_{candidates}(s_i)$ to $e(s_i)$.layer without the inserted jumper*/
13 **if**($via_{original} > via_{modified}$ or active layers are flexible)
14 { Removed the inserted jumper and
$e(s_i)$.layer is one of $Layer_{candidates}(s_i)$ with less #via(s_i, $B(s_i)$) }
15 break;
16 }
17 }
18 }
19 ***WireSizing***(s_i); /* Adjust wire width to maintain zero skew. */
20 }

Fig. 12. The proposed PADJILA algorithm.

The overview of PADJILA is given as follows. To simplify the presentation, we just discuss how to fix the antenna-critical sink with jumper insertion and layer assignment method (lines 8-16 of the PADJILA algorithm). As aforementioned, if the accumulated length of sink s_i, $|e_{total}(s_i)|$, is longer than L_{max}, s_i is antenna-critical and should be fixed by jumper insertion. For a pair of sinks s_i and its brother $B(s_i)$, $via_{original}$ includes the sum of node vias of s_i and $B(s_i)$, total edge vias which are inserted on the wires connecting s_i and $B(s_i)$, and the extra vias required by inserted jumpers. In other word, $via_{original}$ is the total vias of s_i and $B(s_i)$ before applying layer assignment method. Then, the *LayerAssignment* procedure obtains the acceptable layer candidates of $e(s_i)$ and gives them to $Layer_{candidates}(s_i)$. Next, $via_{modified}$ acts as $via_{original}$ to count and then store the total vias required by s_i and $B(s_i)$, where $e(s_i)$.layer applies $Layer_{candidates}(s_i)$ and the jumper inserted on $e(s_i)$ is assumed to be removed . If $via_{original}$ is more than $via_{modified}$, or, the active layers are flexible enough to be assigned, the inserted jumper will be removed and $e(s_i)$.layer can apply one of $Layer_{candidates}(s_i)$ to decrease the required vias. The active layers are defined as the layers from the current routing layer, $e(s_i)$.layer, up to the L_{top} of layer configuration. After introducing the overview of PADJILA, the proposed layer assignment procedure is presented in Fig. 13 in detail.

```
Procedure: LayerAssignment(s_i)
Input: A sink s_i
Output: A set of layer candidates of s_i
1  if( e(B(s_i)).layer = L_top of layer configuration )
2  {  return e(B(s_i)).layer;  }
3  else
4  {  if( e(B(s_i)).layer > e(F(s_i)).layer )
5     {  return e(B(s_i)).layer to L_top of layer configuration;  }
6     else if( e(B(s_i)).layer = e(F(s_i)).layer )
7     {  return e(B(s_i)).layer+1 to L_top of layer configuration;  }
8     else
9     {  if( F(s_i) locates on e(s_i) )
10       {  return e(F(s_i)).layer to L_top of layer configuration;  }
11       else
12       {  return e(B(s_i)).layer+1 to L_top of layer configuration;  }
13    }
14 }
```

Fig. 13. The proposed *LayerAssignment* procedure.

For a sink s_i given to the proposed *LayerAssignment* procedure, the acceptable layer candidates of $e(s_i)$ can be obtained immediately. When $e(B(s_i))$.layer equals the L_{top} of layer configuration, the procedure will return $e(B(s_i))$.layer. Otherwise, the procedure will compare the layers of $e(B(s_i))$ and $e(F(s_i))$ (lines 3-14 in *LayerAssignment*).

When $e(B(s_i))$.layer is higher than $e(F(s_i))$.layer, the procedure will return $e(B(s_i))$.layer up to the L_{top} of layer configuration. If $e(B(s_i))$.layer is equal to $e(F(s_i))$.layer, the layer candidates of $e(s_i)$ will be $e(B(s_i))$.layer+1 up to the L_{top} of layer configuration. For the exceptional cases, the procedure will check $F(s_i)$ does locate on $e(s_i)$ or not. If it does, the procedure will return $e(F(s_i))$.layer up to the L_{top} of layer configuration. Otherwise, the layer candidates of $e(s_i)$ will be $e(B(s_i))$.layer+1 up to the L_{top} of layer configuration.

In order to verify the proposed PADJILA algorithm and to compare with other works, Fig. 14(a) shows an initial 16-sink X-clock tree which is as identical as the other one shown in Fig. 10(a). PADJI can insert jumpers to fix antenna violations in the initial X-clock tree, as shown in Fig. 10(b). Moreover, PADJILA integrates layer assignment method and PADJI to decrease the number of inserted jumpers and total required vias, as shown in Fig. 14(c). Comparing with the result as shown in Fig. 14(b) using JILA [20] algorithm, PADJILA gets improvements in terms of delay, power, and total vias.

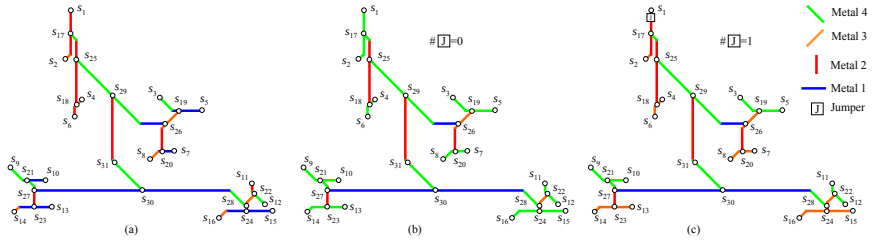


Fig. 14. (a) The initial 16-sink X-clock tree constructed by PMXF. In order to reduce the number of inserted jumpers for fixing antenna violations, the X-clock trees were fixed by (b) JILA [20] and (c) PADJILA algorithms, respectively.

Table III lists the experimental results of PMXF, JILA, and PADJILA algorithms, respectively. Comparing the results of JI and JILA those are respectively shown in Fig. 10(b) and Fig. 14(b), the later algorithm assigned all the 13 wires those were inserted jumpers by the former one with the L_{top} of layer configuration (M4) to decrease the number of inserted jumpers and total vias. From the table, JILA can complete an antenna violation free X-clock tree without jumper insertion. It is not surprising that JILA improved the required total vias, delay, and power consumption compared with JI. However, assigning antenna-critical wires to the L_{top} of layer configuration may result layer

congestion, especially for modern high-density multi-level VLSI routings. Therefore, the proposed PADJILA algorithm can provide several layer candidates to antenna-critical wires for increasing layer flexibility. As the experimental results shown in Table III, PADJILA inserted one jumper, as well as, consumed less total vias than JILA. Moreover, layer assignment method can decrease the number of edge vias. That is the reason PADJILA required less total vias than that of the initial clock tree constructed by PMXF. Next, two sinks s_4 and s_6 in the X-clock trees fixed by PADJI and PADJILA, respectively, are taken as an example to point out the observation and illustrated in Fig. 15.

TABLE III
COMPARISON OF PMXF [21], JILA [20], AND THE PROPOSED PADJILA ALGORITHM WITH L_{MAX}=200μM ON THE GIVEN 16-SINK X-CLOCK TREES.

#Sinks	#Inserted Jumper		Delay (μs)			Skew (μs)			Power (W)			Total vias		
	JILA	PADJILA	PMXF	JILA	PADJILA	PMXF	JILA	PADJILA	PMXF	JILA	PADJILA	PMXF	JILA	PADJILA
16	0	1	0.021500	0.021503	0.021499	0.000009	0.000010	0	0.002083	0.002083	0.002082	92	100	90
Average Ratio	-		-	1.000139	0.999968	-			-	1.000000	0.999519	-	1.086956	0.978260

Figures 15(a) and 15(b) are the local view and cross-section view of the pair of sinks (s_4, s_6) captured from the X-clock tree which was fixed antenna violations with PADJI algorithm. As shown in Fig. 15(b), a jumper is inserted on $e(s_6)$ and needs two extra vias. Sink s_4 needs three NVs to connect poly layer up to M3, as well as s_6 and s_{18} need twos and one, respectively. One EV is inserted on the bending point $P_B(s_4, s_6)$ to connect M2 and M3. Therefore, (s_4, s_6) totally needs nine vias when applying PADJI and $via_{original}$ is set as nine.

As aforementioned, sink s_6 has antenna violation, hence, layer assignment method is required to obtain the layer candidates of $e(s_6)$. In *LayerAssignment* procedure, s_4 and s_{18} are denoted as $B(s_6)$ and $F(s_6)$, respectively. As shown in Fig. 15(a), $e(B(s_6))$.layer is M3 and higher than $e(F(s_6))$.layer, M2. Therefore, *LayerAssignment* procedure obtains that the layer candidates of $e(s_6)$ include M3 and M4. When assigning $e(s_6)$ with M3, the jumper inserted on $e(s_6)$ is assumed to be removed and $via_{modified}$ is obtained as seven, due to s_4, s_6, and s_{18} need three, three, and one NVs, respectively, as well as, the number of EVs is zero. When assigning $e(s_6)$ with M4, $via_{modified}$ is obtained as ten and more than $via_{original}$. Because setting the layer candidate of $e(s_6)$ be M3 can save two vias compared with $via_{original}$, the jumper inserted on $e(s_6)$ is removed and $e(s_6)$ is assigned with M3. Figs. 15(c) and (d) are the local view and cross-section view of the pair of sinks (s_4, s_6) fixed by PADJILA algorithm. Finally, we can proof that layer assignment method not only fixes antenna violations but also decreases the number of inserted jumpers and required vias.

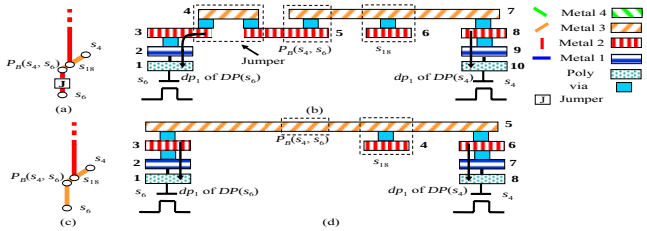


Fig. 15. Take s_4 and s_6 as an example to proof the effectiveness of layer assignment method. (a) The local view of X-clock tree fixed by PADJI where a jumper is inserted on $e(s_6)$, (b) the cross-section view of (a), (c) the local view of X-clock tree fixed by PADJILA where $e(s_6)$ is assigned to M3, and (d) the cross-section view of (c).

When the number of inserted jumpers and required vias are not able be decreased by layer assignment method, jumper insertion should be applied for layer congestion consideration. Two sinks s_1 and s_2 in the X-clock tree fixed by PADJI and PADJILA are taken as an example to proof the observation, respectively. Figures 16(a) and 16(b) are the local view and cross-section view of the pair of sinks (s_1, s_2), respectively, captured from the X-clock tree which applies PADJI algorithm to fix antenna violations. As shown in Fig. 16(b), a jumper is inserted on $e(s_1)$ and needs two extra vias. Sink s_1 are inserted two NVs to connect poly layer up to M2, as well as, s_2 and s_{17} need threes and twos, respectively. One EV is inserted on the bending point $P_B(s_1, s_2)$ to connect M2 and M3. Therefore, (s_1, s_2) totally needs ten vias when applying PADJI and $via_{original}$ is set as ten.

As aforementioned, sink s_1 is antenna-critical and requires layer assignment method to obtain the layer candidates of $e(s_1)$. In the *LayerAssignment* procedure, s_2 and s_{17} are denoted as $B(s_1)$ and $F(s_1)$, respectively. As shown in Fig. 16(a), $e(B(s_1))$.layer is M3 and lower than $e(F(s_1))$.layer, M4, as well as, $F(s_1)$ located on $e(s_1)$. Therefore, *LayerAssignment* procedure obtains that the layer candidates of $e(s_1)$ is M4. When assigning $e(s_1)$ with M4, the jumper inserted on $e(s_1)$ is assumed to be removed and $via_{modified}$ is obtained as eight, due to s_1 and s_2 need four and three NVs, respectively, as well as, an EV is inserted on $P_B(s_1, s_2)$. However, this process results that s_2 is also antenna-critical and another jumper is required for fixing. Meanwhile, $via_{modified}$ is increased from eight to ten and the number of inserter jumpers cannot be decreased. Finally, the layer assignment method is skipped when the ping-pong effect is occurred. Here, the ping-pong effect is defined as that when assigning one of layer candidates may cause another antenna violation and require an extra jumper for fixing. The jumper inserted on $e(s_1)$ is kept. Figures 16(c) and 16(d) are the local view and cross-section view of the pair of sinks (s_1, s_2), respectively, when applying the layer assignment method.

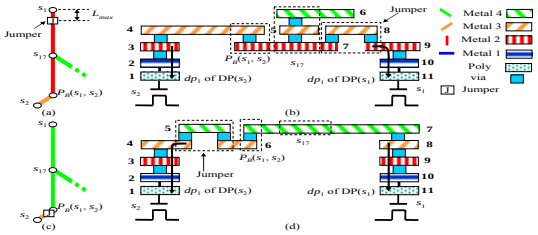


Fig. 16. Take s_1 and s_2 as an example to explain the ping-pong effect of layer assignment method. (a) The local view of X-clock tree fixed by PADJI where a jumper is inserted on $e(s_6)$, (b) the cross-section view of (a), (c) the local view of X-clock tree when applying layer assignment method to assign $e(s_1)$ with M4, and (d) the cross-section view of (c).

After discussing the proposed PADJILA algorithm in detail, the analysis of time complexity is given as follows. As aforementioned, every sink in the given X-clock tree will be examined to determine whether it has antenna violation or not. Hence, the *for* loop runs in $O(n)$. In the *for* loop, the *DischargePath* procedure obtains a set of discharge paths of each sink and the worst case is that each sink contains k paths. Therefore, the procedure runs in $O(k)$. The proposed *LayerAssignment* procedure can obtain the acceptable layer candidates of the edge connecting antenna-critical sink in constant time. Finally, the time complexity of the proposed PADJILA algorithm is $O(n^2)$.

VI. Experimental Results

The proposed PADJI and PADJILA algorithms were implemented with C++ on a Windows machine with 1.7GHz Pentium-4 processor and 512MB memory. The fabrication parameters of ED and FED delay models for 70nm and 130nm process are respectively listed in Table IV. Three sets of benchmarks for evaluating the proposed algorithms and comparing with other works were downloaded from the GSRC bookshelf [28]: (i) Primary1 and Primary2 (MCNC benchmarks [29]), (ii) *r*1-*r*5 (IBM benchmarks [30]), and (iii) s1423, *s*5378, and s15850 (ISCAS89 benchmarks [31]). Table V lists the number of sinks, chip size, and load capacitances of each benchmark.

TABLE IV TECHNOLOGY PARAMETERS FOR THE PROPOSED ALGORITHMS AND OTHER PRIOR WORKS.

Tech. (nm)	Delay Model	w_{min} (nm)	r (Ω/μm)	c_a (fF/μm)	c_f (fF/μm)	D/ln2	E/ln2	F/ln2	F_{clk} (Hz)	V_{dd} (v)	L_{max}(μm)
70	ED	70	1.357	0.00392	-	-	-	-	2.5G	0.75	100
130	FED	130	0.623	0.00598	0.043	1.12673	1.10463	1.04836	100M	1.2	200

TABLE V RELATED INFORMATION OF BENCHMARKS.

Benchmarks		#Sinks	Chip size (μm x μm)		Load capacitance
			Height (μm)	Width (μm)	(*f*F)
MCNC	Primary1	269	6000	6000	500
	Primary2	603	10500	10500	500
IBM	*r*1	267	70000	69984	30 – 80
	*r*2	598	93134	94016	30 – 80
	*r*3	862	98500	97000	30 – 80
	*r*4	1903	126988	126970	30 – 80
	*r*5	3101	145224	142920	30 – 80
ISCAS89	s1423	74	11000	14000	50
	*s*5378	179	13000	13000	50
	*s*15850	597	15000	16000	50

For comparative study, this work first applies PMXF [21] algorithm to construct the initial X-clock trees of all the benchmarks without antenna effect consideration. Next, JI [20] and PADJI algorithms respectively loaded these X-clock trees to detect antenna effects and then inserted jumpers to fix antenna violations. Table VI lists the experimental results of all the benchmarks in the number of inserted jumper, delay, skew, power, and total vias. FED model with 130nm fabrication parameters is used to calculate delay and the upper bound of antenna effect L_{max} is given as 200μm. The ratio is defined as JI/PMXF or PADJI/PMXF. From the table, PADJI inserted 46.12% less jumpers than JI on average. Hence, PADJI had better performance of 0.0932%, 0.0913%, and 46.18% on average in terms of delay, power, and total vias, respectively, compared with JI. Moreover, the proposed wire sizing technique can maintain the zero skew property of given X-clock trees. Comparing with PMXF, PADJI slightly increased 0.0672% in delay, 0.0749% in power, and 17.21% in total vias due to fixing antenna violations with jumper insertion.

TABLE VI COMPARISON OF PMXF [21], JI [20], AND THE PROPOSED PADJI ALGORITHMS BASED ON 130NM FED MODEL AND L_{MAX}=200μM.

Bench marks	#Sinks	#Inserted Jumper		Delay (μs)			Skew (μs)			Power (W)			Total vias		
		JI	PADJI	PMXF	JI	PADJI	PMXF	JI	PADJI	PMXF	JI	PADJI	PMXF	JI	PADJI
Primary1	269	211	79	0.059456	0.059495	0.059483	0.000806	0.000816	0	0.174272	0.174341	0.174322	1194	2014	1352
Primary2	603	353	305	0.254020	0.254135	0.254113	0.000688	0.000694	0	0.412771	0.412895	0.412874	2350	3720	2960
*r*1	267	207	115	0.310696	0.311018	0.310928	0.000360	0.000389	0	0.063340	0.063405	0.063359	1220	2022	1450
*r*2	598	439	256	1.125454	1.126724	1.125910	0.000859	0.000975	0	0.156019	0.156188	0.156067	2846	4646	3358
*r*3	862	642	378	1.803820	1.805865	1.804441	0.001217	0.001301	0	0.210124	0.210368	0.210197	4020	6562	4776
*r*4	1903	1421	815	4.804431	4.810863	4.806244	0.002762	0.003736	0	0.487032	0.487646	0.487208	9158	14866	10788
*r*5	3101	2283	1292	8.585707	8.597276	8.588974	0.003049	0.004162	0	0.809903	0.810941	0.810201	14585	23555	17169
s1423	74	54	20	0.007489	0.007509	0.007492	0.000038	0.000041	0	0.006009	0.006039	0.006028	351	559	391
*s*5378	179	137	70	0.017915	0.017965	0.017930	0.000033	0.000027	0	0.015470	0.015508	0.015490	772	1308	912
*s*15850	597	414	166	0.049893	0.050067	0.050013	0.000124	0.000159	0	0.058674	0.058830	0.058722	2711	4399	3043
Average Ratio		-	0.5388		1.001604	1.000672	-	-	-	-	1.001662	1.000749	-	1.6339	1.1721

The proposed PADJILA algorithm has integrated PADJI and the *LayerAssignment* procedure to reduce the number of inserted jumpers in advance. Table VII lists the experimental results of all the benchmarks constructed and fixed by PMXF, JILA [20], and PADJILA algorithms, respectively. The ratio is defined as JILA/PMXF or PADJILA/PMXF. Comparing Tables VI and VII, the number of inserted jumpers required by JILA and PADJILA can be effectively reduced compared with JI and PADJI. Moreover, PADJILA inserted 48.21% less jumpers than JILA on average and respectively achieved reductions of 0.0242%, 0.0186%, and 20.36%, in delay, power, and total vias, compared with JILA. Comparing with PMXF, PADJILA slightly increases 0.0297% in delay, 0.0536% in power, and 0.87% in total vias. It is obvious that the proposed PADJI and PADJILA algorithms outperformed other methods significantly.

In order to examine the proposed algorithms in the advanced manufacturing process, Table VIII respectively lists the experimental result of PADJI and PADJILA based on 70nm technology. The ratio is defined as PADJILA/PADJI. From the table, PADJILA made improvements of 95.4%, 0.0043%, 0.0019%, and 6.27% on average in the number of inserted jumpers, delay, power, and total vias, respectively. Figure 17(a) shows the full chip and local views of benchmark *r*4 fixed by the proposed PADJI algorithm. Figure 17(b) presents the other routing result fixed by PADJILA algorithm. For fixing antenna violation, the inserted jumpers shown in the local views are represented with the squares labeled in J. Comparing the three local views respectively shown in the right sides of Figs. 17(a) and 17(b), the number of inserted jumpers can be effectively reduced by layer assignment method to improve delay and power consumption. In contrast, the case shown in Fig. 14 is a small scale X-clock routing and its congestions of high metal layers (M3 and M4) are lower than those of benchmarks. Therefore, the decrease degree of total vias listed in Table III is higher than that listed in Table VIII.

TABLE VII COMPARISON OF PMXF [21], JILA [20], AND THE PROPOSED PADJILA ALGORITHMS BASED ON130NM FED MODEL AND L_{MAX}=200μM.

Bench marks	#Sinks	#Inserted Jumper		Delay (μs)			Skew (μs)			Power (W)			Total vias		
		JILA	PADJILA	PMXF	JILA	PADJILA	PMXF	JILA	PADJILA	PMXF	JILA	PADJILA	PMXF	JILA	PADJILA
Primary1	269	12	3	0.059456	0.059472	0.059461	0.000806	0.000813	0	0.174272	0.174299	0.174281	1194	1520	1200
Primary2	603	8	7	0.254020	0.254089	0.254059	0.000688	0.000700	0	0.412771	0.412839	0.412813	2350	3107	2364
*r*1	267	13	7	0.310696	0.310805	0.310784	0.000360	0.000399	0	0.063340	0.06336	0.063355	1220	1470	1234
*r*2	598	22	12	1.125454	1.125812	1.125730	0.000859	0.000909	0	0.156019	0.156061	0.156049	2846	3299	2870
*r*3	862	37	20	1.803820	1.804302	1.804236	0.001217	0.001165	0	0.210124	0.210189	0.210175	4020	4702	4060
*r*4	1903	80	32	4.804431	4.806045	4.805609	0.002762	0.002551	0	0.487032	0.487185	0.487146	9158	10597	9222
*r*5	3101	145	71	8.585707	8.588937	8.588136	0.003049	0.003488	0	0.809903	0.810178	0.810109	14585	17000	14727
s1423	74	0	0	0.007489	0.007494	0.007491	0.000038	0.000041	0	0.006009	0.006029	0.006027	351	399	351
s5378	179	13	8	0.017915	0.017937	0.017926	0.000033	0.000030	0	0.015470	0.015486	0.015480	772	1003	788
s15850	597	32	13	0.049893	0.049958	0.049921	0.000124	0.000143	0	0.058674	0.058732	0.058698	2711	3350	2737
Average Ratio		-	0.5179	-	1.000539	1.000297	-	-	-	-	1.000722	1.000536	-	1.2123	1.0087

TABLE VIII COMPARISON OF THE PROPOSED PADJI AND PADJILA ALGORITHMS BASED ON 70NM ED MODEL AND L_{MAX}=100μM.

Bench marks	#Sinks	#Inserted Jumper		Delay (μs)			Skew (μs)			Power (W)			Total vias		
		PADJI	PADJILA	PMXF	PADJI	PADJILA	PMXF	PADJI	PADJILA	PMXF	PADJI	PADJILA	PMXF	PADJI	PADJILA
Primary1	269	103	6	0.182017	0.182036	0.182026	0.002284	0	0	1.660459	1.660469	1.660469	1081	1287	1193
Primary2	603	332	8	0.598993	0.599016	0.599001	0.004528	0	0	4.028535	4.028721	4.028604	1803	2467	2241
*r*1	267	120	3	0.298881	0.298922	0.298903	0.000297	0	0	0.230762	0.230781	0.230771	1214	1454	1389
*r*2	598	278	14	0.646688	0.646757	0.646733	0.000438	0	0	0.552051	0.552109	0.552090	2708	3264	3115
*r*3	862	405	26	1.210099	1.210222	1.210183	0.000208	0	0	0.808037	0.808115	0.808096	4134	4944	4730
*r*4	1903	864	48	3.409288	3.409644	3.409538	0.000969	0	0	1.961904	1.962100	1.962041	8986	10714	10205
*r*5	3101	1417	78	6.757963	6.758670	6.758445	0.001092	0	0	3.370986	3.371318	3.371221	14608	17442	16553
s1423	74	30	1	0.013947	0.013949	0.013948	0.000029	0	0	0.035684	0.035684	0.035684	359	419	382
s5378	179	73	3	0.029410	0.029414	0.029412	0.000080	0	0	0.102754	0.102764	0.102764	782	928	910
s15850	597	270	15	0.122842	0.122857	0.122856	0.000113	0	0	0.413496	0.413535	0.413535	2600	3140	3112
Average Ratio		-	0.046	-	1.000110	1.000067	-	-	-	-	1.000073	1.000054	-	1.2106	1.1479

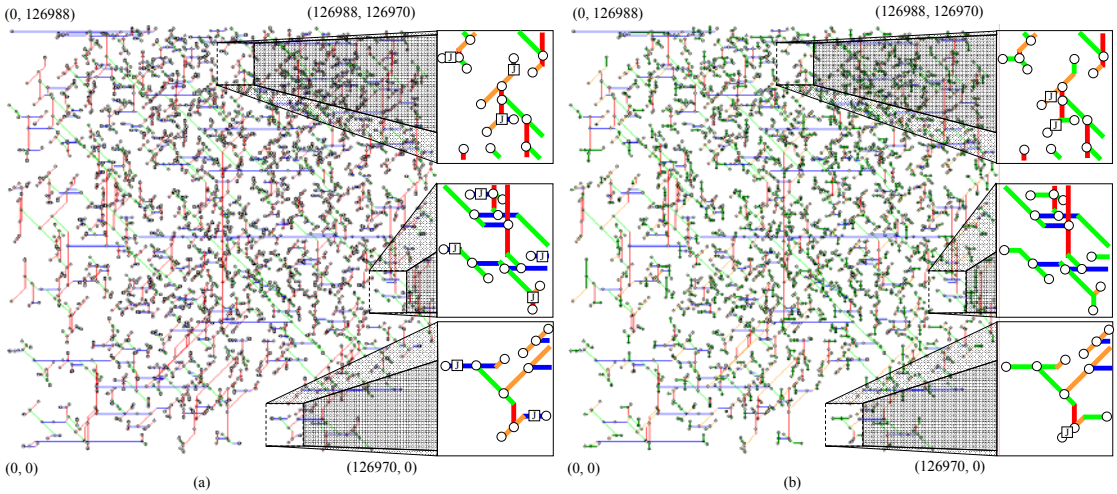


Fig. 17. Full chip views and local views of *r*4 constructed by PMXF and fixed by (a) PADJI and (b) PADJILA.

VII. Conclusion

Antenna effect is one of the DFM (design for manufacturing) problems in nanometer process. This work proposed a discharge-path-based antenna detection algorithm and achieved antenna violations free X-clock routings with jumper insertion algorithm. Moreover, the proposed algorithm integrated layer assignment method to reduce the number of inserted jumpers for clock delay and power consumption improvements. For via timing impact consideration, experimental results revealed that the influences on delay and power consumption are very slight. However, modern complex multi-level VLSI designs wherein the number of vias is on the order of 50 million and even higher. Hence, not only clock tree construction but also jumper insertion and layer assignment for eliminating antenna effect should consider via timing impact seriously. For other DFM issues, the work can be extended to combine with optical proximity correction (OPC) and double via insertion into X-clock routing. For the clock tree optimization, buffers are inserted on the interconnections to improve clock delay. However, the metal wires connecting the input ports of buffers may accumulate sufficient charges to damage the gates of buffers. Therefore, the buffered clock trees should require more jumpers to fix antenna violations compared with bufferless clock trees.

REFERENCES

[1] L. Riviere-Cazaux, K. Lucas, and J. Fitch, "Integration of Design For Manufacturability (DFM) Practices In Design Flows," in *Proc. International Symposium Quality Electronic Design*, March 2005, pp. 102-106.

[2] J. T. Kong, "CAD for Nanometer Silicon Design Challenges and Success," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 12, No. 11, Nov. 2004, pp. 1132-1146.

[3] S. Krishnan, S. Rangan, S. Hattangady, G. Xing, K. Brennan, M. Rodder and S. Ashok, "Assessment of charge-induced damage to ultra-thin gate MOSFETs," in *IEDM Technology Dig*est, 1997, pp. 445-448.

[4] Z. Chen and I. Koren, "Layer reassignment for antenna effect minimization in 3-layer channel routing," in *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, Nov. 1996, pp. 77-85.

[5] *TSMC 0.18μm Design Manual*, Document number TA-10A5-4001, pp. 67-68.

[6] P. H. Chen, S. Malkani, C. M. Peng, and J. Lin, "Fixing antenna problem by dynamic diode dropping and jumper insertion," in *Proc. International Symposium Quality Electronic Design*, March 2000, pp. 275-282.

[7] H. Shirota, T. Sadakane, M. Terai, and K. Okazaki, "A new router for reducing antenna effect in ASIC design," in *Proc. IEEE Custom Integrated Circuits Conference*, May 1998, pp. 601-604.

[8] T. Y. Ho, Y. W. Chang, and S. J. Chen, "Multilevel routing with antenna avoidance," in *Proc. ACM International Symposium Physical Design*, Apr. 2004, pp. 33-40.

[9] T. Y. Ho, Y. W. Chang, and S. J. Chen, "Multilevel routing with jumper insertion for antenna avoidance," *Integration: VLSI* Journal, vol. 29, no. 4, pp. 420-432, Jul. 2006.

[10] B. Y. Su and Y. W. Chang, "An exact jumper insertion algorithm for antenna effect avoidance/fixing," in *Proc. ACM/IEEE Design Automation Conference*, Jun. 2005, pp. 597-602.

[11] B. Y. Su, Y. W. Chang, and J. Hu, "An optimal jumper insertion algorithm for antenna avoidance/fixing on general routing trees with obstacles," in *Proc. ACM International Symposium Physical. De*sign, Apr. 2006, pp. 56-63.

[12] B. Y. Su, Y. W. Chang, and J. Hu, "An exact jumper insertion algorithm for antenna violation avoidance/fixing considering routing obstacles," *IEEE Trans. on Computer-Aided-Design Integrated Circuits and Syst*ems, vol. 26, no. 4, pp. 719-734, Apr. 2007.

[13] B. Y. Su and Y. W. Chang, "An optimal jumper-insertion algorithm for antenna avoidance/fixing," *IEEE Trans. on Computer-Aided Design Integrated Circuits and Syst*ems, vol. 26, no. 10, pp. 1818-1829, Oct. 2007.

[14] D. Wu, J. Hu, and R. Mahapatra, "Coupling aware timing optimization and antenna avoidance in layer assignment," in *Proc. ACM International Symposium Physical Design*, Apr. 2005, pp. 20-27.

[15] D. Wu, J. Hu, and R. Mahapatra, "Antenna avoidance in layer assignment," *IEEE Trans. on Computer.-Aided Design Integrated Circuits and Syst*ems, vol. 25, no. 4, pp. 734-738, Apr. 2006.

[16] L. D. Huang, X. Tang, H. Xiang, D. F. Wong, and I. M. Liu, "A polynomial time optimal diode insertion/routing algorithm for fixing antenna problem," in *Proc. ACM/IEEE Design Automation Test in Europe*., Mar 2002, pp. 470-475.

[17] L. D. Huang, X. Tang, H. Xiang, D. F. Wong, and I. M. Liu, "A polynomial time optimal diode insertion/routing algorithm for fixing antenna problem," *IEEE Trans. on Computer-Aided Design Integrated Circuits and Syst*ems, vol. 23, no. 1, pp. 141-147, Jan. 2004.

[18] Z. W. Jiang and Y. W. Chang, "An optimal simultaneous diode/jumper insertion algorithm for antenna fixing," in *Proc. IEEE/ACM International Conference Computer-Aided De*sign, Nov. 2006, pp. 669-674.

[19] Z. W. Jiang and Y. W. Chang, "An optimal network-flow-based simultaneous diode and jumper insertion algorithm for antenna fixing," *IEEE Trans. on Computer-Aided Design Integrated Circuits and Systems*, vol. 27, no. 6, pp. 1055-1065, June 2008.

[20] C. C. Tsai, F. T. Hsu, C. C. Kuo, J. O. Wu, and T. Y. Lee, "Antenna effect consideration for X-clock routing," in *Proc. IEEE International Conf. on Solid-State and Integrated-Circuit Technology*, Oct. 2008, pp. 2248-2251.

[21] C. C. Tsai, C. C. Kuo, J. O. Wu, T. Y. Lee, and R. S. Hsiao, "X-clock routing based on pattern matching," in *Proc. IEEE International SOC Conf.*, Sept. 2008, pp. 357-360.

[22] S. L. Teig, "The X-architecture: not your father's diagonal wiring," in *Proc. Int. Workshop on System-Level Interconnect Prediction*, Apr. 2002, pp. 33-37.

[23] R. S. Tsay, "Exact zero skew," in *Proc. International Conf. Computer-Aided Design*, Nov. 1991, pp. 336-339.

[24] W. C. Elmore, "The transient response of damped linear network with particular regard to wideband amplifiers," *Journal Applied Physics*, vol. 19, pp. 55-63, 1948.

[25] Arif Ishaq Abou-Seido, Brian Nowak, and Chris Chu, "Fitted Elmore Delay: a simple and accurate interconnect delay model," *IEEE Trans. on VLSI*, vol. 12, no. 7, pp. 691-696, 2004.

[26] T. Y. Ho, Y. W. Chang, S. J. Chen, and D. T. Lee, "Crosstalk- and performance-driven multilevel full-chip routing," *IEEE Trans. on Computer-Aided Design Integrated Circuits and Syst*ems, vol. 24, no. 6, pp. 869-878, June 2005.

[27] C. H. Wang and W. K. Mak, "λ-Geometry clock tree construction with wire length and via minimization," in *Proc. IEEE International Symposium VLSI-DAT*, Apr. 2007, pp. 124-127.

[28] VLSI CAD Laboratory, University of California, San Diego, "Bounded-skew clock tree routing version 1.0," May 2000, http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST/.

[29] M. A. B. Jackson, A. Srinivasan and E. S. Kuh, "Clock routing for high-performance ICs," in *Proc. ACM/IEEE Design Automation Conf.*, June 1990, pp. 573-579.
[30] R. S. Tsay, "An exact zero-skew clock routing algorithm," *IEEE Trans. on Computer-Aided Design Integrated Circuits and Syst*ems, vol. 12, no. 2, pp. 242-249, Feb. 1993.
[31] J. G. Xi, and Wayne W. M. Dai, "Useful-skew clock routing with gate sizing for low power design," in *Proc. ACM/IEEE Design Automation Conf.*, June 1996, pp. 383-388.

研究成果與相關論文發表

1. Chia-Chun Tsai, Jan-Ou Wu, and Trong-Yen Lee, "Maximal Delay Reduction for RLC-Based Multi-source Multi-sink Bus with Repeater Insertion," Accepted by *Circuits, Systems & Signal Processing*, Paper No. 70826, 2009. (SCI, EI)
2. Chia-Chun Tsai, Chin-Yen Lin, Yuh-Shyan Hwang, and Trong-Yen Lee, "The Design of a Li-Ion Battery Charger Based on Multimode LDO Technology," *Journal of Circuits, Systems, and Computers*, Vol. 18, No. 5, pp. 947-963, 2009. (EI)
3. Chia-Chun Tsai, Kai-Wei Hong, and Trong-Yen Lee, "A Bisection-Based Power Reduction Design for CMOS Flash Analog-to-Digital Converters," *Journal of Circuits, Systems, and Computers*, Vol. 18, No. 5, pp. 933-945, 2009. (EI)
4. Chia-Chun Tsai, Huang-Chi Chou, and Trong-Yen Lee, "The Circuit Design of Current-Mode Image Sensor Embedded Smooth Spatial Filter with Flash A/D Converter," *WSEAS Transactions on Circuits and Systems*, Issue 2, Vol. 8, pp. 237-246, Feb. 2009. (EI)
5. Chia-Chun Tsai, Chung-Chieh Kuo, Trong-Yen Lee, Lin-Jeng Gu, and Jan-Ou Wu, "Buffer Insertion and Sizing for X-Architecture Clock Routing," *The 20th VLSI Design/CAD Symposium*, August 2009, Hualein, Taiwan.
6. Chia-Chun Tsai, Chung-Chieh Kuo, Trong-Yen Lee, and Jan-Ou Wu, "X-architecture Clock Tree Construction Associated with Buffer Insertion and Sizing," *The 1st Asia Symposium on Quality Electronic Design*, pp. 298-303, July 15-16, 2009, Kuala Lumpur, Malaysia.
7. Chia-Chun Tsai and Jie-Hou You, "X-Architecture Clock Routing with Considering Voltage-Island Placement," *Information Management and E-Commerce Management Symposium*, Paper no. 26, May 30, 2009, Chiayi, Taiwan.
8. Chia-Chun Tsai, Feng-Tzu Hsu, Chung-Chieh Kuo, Jan-Ou Wu, and Trong-Yen Lee, "X-Clock Tree Construction for Antenna Avoidance," *The 9th International Conference on Solid-State and Integrated-Circuit Technology*, Paper H1.6, October 20-23, 2008, Beijing, China.
9. Chia-Chun Tsai, Chung-Chieh Kuo, Jan-Ou Wu, Trong-Yen Lee, and Rong-Shue Hsiao, "X-Clock Routing Based on Pattern Matching," *The 21st Annual IEEE International SOC Conference*, pp. 357-360, Sept. 17-20, 2008, Newport Beach, CA, USA. (EI)
10. Chia-Chun Tsai, Feng-Tzu Hsu, Chung-Chieh Kuo, Jan-Ou Wu, and Trong-Yen Lee, "Antenna Effect Consideration for X-Clock Routing," *Proc. The 19th VLSI Design/CAD Symposium*, Paper ID:179, Aug. 5-8, 2008, Howard Beach Resort, Kenting, Taiwan.

出席國際會議研討心得報告

南華大學資工系 蔡加春 教授

國科會專題計畫補助：NSC 97-2221-E-343-008，2008/8~2009/7

整合可製造性設計方法應用於 X-結構時脈繞線合成之研究

Integrated Methods of Design for Manufacturability Applied for X-Architecture Clock Routing Synthesis

● 2008 年固態與積體電路技術國際研討會 ICSICT 2008, Oct. 20-23, Beijing, China

固態與積體電路技術國際研討會（ICSICT--International Conference on Solid-State and Integrated-Circuit Technology）是一個高品質又專業的技術研討會，它提供給此領域的業界、學術界及資訊技術系統應用者等經驗交流的機會，引起熱烈的迴響與好評。今年是第九屆 ICSICT 國際研討會（ICSICT 2008），於 2008 年 10 月 20-23 日在大陸北京京儀大飯店舉行(Beijing Jingyi Hotel, Beijing, China)，主辦單位爲 ICSICT Organizing Committee、IEEE Beijing Section、Chinese Institute of Electronics (CIE) 、IEEE Electron Devices Socity (EDS)，協辦單位為 IEEE Solid-State Circuits Society (SSCS), IEEE Circuits and Systems Society, IEEE Hong Kong EDS/SSCS Chapter, IEEE SSCS Beijing Chapter, Japan Society of Applied Physics, Electronics Division of IEE (U.K), URSI Commission D, Institute of Electronics Engineers of Korea, Association of Asia Pacific Physical Societies, IEEE EDS Student Chapter of Peking University。

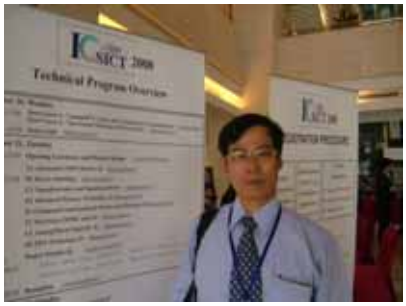
京儀大飯店爲新北京區域剛落成的旅館，大會特別導引如何從北京機場搭機場快軌，再接北京地鐵如何找到京儀大飯店之開會地方。

本年國際研討會計有來自世界28個國家之專家學者，含美洲、亞洲、歐洲與大洋洲等，共投稿與邀稿超過930篇論文，只有400篇論文被接受在會議上口頭發表及256篇論文做海報發表，接受率約70%，這些被接受論文分佈於三個整天與八間meeting rooms同時舉行，包含45 個regular lecture sessions與2個poster sessions。爲：另有五場Keynotes爲：Won-Seong Lee, Samsung Electronics Co., Ltd, Korea 之「Future Memory Technologies」、Rene Penning de Vries, NXP Semiconductors, Netherlands 之「IC Innovations in Automotive」、Masayoshi Esashi,

Tohoku University, Japan 之「Application Oriented MEMS by Open Collaboration」、Mark Bohr, Intel Corporation, USA 之「Using Innovation to Drive Moore's Law」、Chenming Hu, University of California, Berkeley, USA 之「Green Transistor as a Solution to the IC Power Crisis」。及兩場的Pannels爲:『ICs in China: Challenges and Opportunities』，討論者有 Min-Hwa Chi, Vice President, SMIC, China、Guoqiang Dai, Director, China Ministry of Science & Technology、Haiying Li, Vice Dean, Skyworth Group Research Institute, China、Shaojun Wei, Professor, Tsinghua University, China、Ping Wu, CEO, Spreadtrum Communications, China、Tianchun Ye, Director, Institute of Microelectronics, CAS, China、Kevin Zhang, Intel Fellow, Intel Corp., USA等。及『Green Microelectronics』，討論者有Chenming Hu, University of California, Berkeley, USA、Mikael Östling, KTH, Royal Institute of Technology, Sweden、George A. Rozgonyi, North Carolina State University, USA、Mohamad Sawan, Polytechnique University of Montreal, CanadaHsing-Huang Tseng, SEMATECH, USA、Kang L. Wang, University of California, Los Angles, USA、Ko-Min Chang, Freescale Semiconductor, USA等

國內在此會議中總計發表論文超過二十篇，來自各公私立大學及科技大學等，南華大學只有本人參與，藉此國內外學者交流，有助於增加南華大學的曝光率。本人有一篇論文以口頭在會中發表，隸屬於 EDA 領域。

- Chia-Chun Tsai, Feng-Tzu Hsu, Chung-Chieh Kuo, Jan-Ou Wu, and Trong-Yen Lee, "X-Clock Tree Construction for Antenna Avoidance," The 9th International Conference on Solid-State and Integrated-Circuit Technology, Paper H1.6, October 20-23, 2008, Beijing, China.

本研討會有兩個 sessions 爲 EDA 領域，總計 19 篇口頭發表，很值得觀摩與交流。此行參加研討會，與來自世界各地之國際學者相互交流，藉此了解他們研究方向與成果，並帶回大會相關資料與論文光碟片，及與一些國際學者與業界交流經驗，也認識不少各國的研究生，感受他們研究的積極精神與英文表達能力。尤其大陸已有一些如復旦大學、浙江大學、清華大學與北京大學等開始注重 EDA 領域，值得觀察其後續研究與競爭。

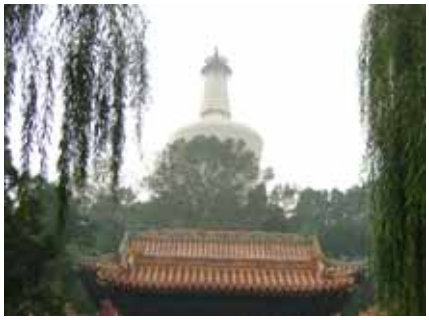
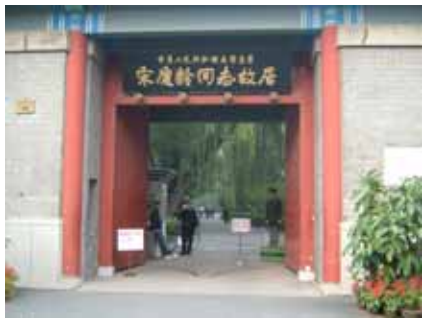





藉此機會參訪一些大陸文化古蹟，王府井大街(嚐北京烤鴨)、天安門廣場、紫禁城(故宮)、景山公園(明思宗殉國處)、北海公園(瓊島之白塔)、什剎海、遊胡

同區、宋慶齡故居(1963-1981)、天壇公園(老年人雜耍自由秀，快樂又健康)、奧運公園(鳥巢、水立方)、香山公園(楓葉風紅)、頤和園(仁壽殿與昆明湖)等。但沒時間前往圓明園、清大校園、北大校園、萬里長城、明十三陵等。






此行之機票、住宿與生活費等經費由國科會計畫 (NSC 計畫名稱：整合可製造性設計方法應用於 X-結構時脈繞線合成之研究，計畫編號：NSC97-2221-E-343－008) 國外差旅費來支付；北京之旅收穫豐碩，感謝國科會補助與南華大學的熱心支持。

● 2009 年亞洲版品質電子設計國際研討會

1st ASQED, July 15-16, 2009, Kuala Lumpur, Malaysia

品質電子設計國際研討會（ISQED---International Symposium on Quality Electronic Design）在美國已舉辦十屆了，它是一個高品質又專業的技術研討會，提供給此領域的業界、學術界及資訊技術系統應用者等經驗交流的機會，引起熱烈的迴響與好評。今年第一次建立亞洲 ISQED（ASQED---Asia Symposium on Quality Electronic Design），於 2009 年 7 月 15-16 日在馬來西亞吉隆坡 **Crowne Plaza Mutiara Kuala Lumpur** 舉行。This symposium will be co-organized by **Malaysian Industry-Government Group for High Technology (MIGHT) and Malaysian Institute of Microsystems (MIMs)** with support from **Malaysian Industrial Development Authority (MIDA).** This symposium also gathers leading industrial and educational entity such as **Intel Microelectronics (M) Sdn. Bhd**), **STATs ChipPAC Malaysia Sdn Bhd**. In continuations of 10 years of successful leadership by ISQED conference in USA, ASQED'09 is aimed at bridging the gap among electronic design tools and processes, integrated circuit technologies, processes & manufacturing, to achieve design quality. The conference emphasizes innovations and the latest developments in IC and system Design, EDA, Semiconductor Process Technology, IC Packaging, Test, and Manufacturing communities. ASQED'09 spans two days, Wednesday through Thursday, in three parallel tracks, hosting over 60 technical presentations, several keynote speakers, tutorials and other informal meetings. Conference proceedings will be published by IEEE and posted in the digital library. ASQED corporate sponsors and supporters include Synopsys, Mentor Graphic, and Cadence Design Systems.

國內在此會議中總計發表論文近十篇，來自各公私立大學等，南華大學只有本人參與，藉此國內外學者交流，有助於增加南華大學的曝光率。本人有一篇論文以口頭在會中發表，隸屬於 EDA 領域。

Chia-Chun Tsai, Chung-Chieh Kuo, Trong-Yen Lee, and Jan-Ou Wu, "X-architecture Clock Tree Construction Associated with Buffer Insertion and Sizing," *The 1st Asia Symposium on Quality Electronic Design*, pp. 298-303, July 15-16, 2009, Kuala Lumpur, Malaysia.

(與 Session Chair Dr. Pan 合影)

藉此機會參訪吉隆坡一些古蹟，如 KL Tower 雙子星、Merdeka Square、Masjid Putrajaya 及 Government Office Building。

原國科會計畫之國外差旅費四萬元已於上次參加 **ICSICT 2008** 用完；此行之機票、住宿與生活費等經費多數由南華大學來補助支付，吉隆坡之旅收穫豐碩，感謝南華大學的補助與熱心支持。