Crosstalk Minimization For Multiple Clock Tree Routing

Ming-Fu Hsiao, Malgorzata Marek-Sadowska*, Sao-Jie Chen
Department of Electrical Engineering, National Taiwan University, Taiwan, R.O.C.
*Department of Electrical and Computer Engineering,
University of California, Santa Barbara, CA 93106, USA

Abstract

Crosstalk noise has been identified as a very important factor for deep submicron chip design. Signals running in parallel on the same layer can experience crosstalk noise. Among all the possible crosstalk noise sources, clock is the most important aggressor as well as victim. Besides, for modern chip design, there are usually more than one clock source, sometimes tens of clock sources. It is important to design the clock topologies for all the clocks running on the same chip to prevent possible crosstalk noise among them. In this paper, we deal with the minimization of inter-clock crosstalk. We propose algorithms to generate clock topology and routing to minimize effective crosstalk. The experimental results show a significant reduction of effective crosstalk compared to that of the conventional clock tree synthesis wherein crosstalk effect is not taken into account.

1. Introduction

With the recent advances in VLSI technology, the device sizes have shrunk below 0.1um. The shrinking of geometries has brought two new major concerns for signal integrity. One is the power and ground noise caused by simultaneous switching circuits. The other problem is the increasing aspect ratio of wires and the decreasing of interconnect spacing which have made coupling capacitance larger than self-capacitance. The ratio of coupling capacitance is reported to be as high as 70%-80% of the total capacitance. For high speed circuits, it has become increasingly important to consider the coupling effect during the layout phase.

Clock synthesis has been an important issue of chip design for more than ten years. To make the design robust and easy to migrate, synchronous design has been the most popular style to achieve this goal. Also, the high integration of chip functions has increased the number of clocks from one to as many as tens. To serve the tradeoff between performance and cost, chip is often designed with hierarchies of clocks to achieve the best performance and cost ratio. For example, consider the on-chip memory. Registers are the highest frequency memory and are synchronized with CPU. But registers can serve only a small amount of the memory requirement. Cache memory is often used as the second choice, other low speed memory such as DRAM the third choice, and so on. Different types of memory use different synchronous circuits, thus increasing the number of clocks. Others, like interface circuits or different functional units can also increase the number of clocks on a single chip. Because clocks are distributed throughout the whole chip and toggle in every cycle with the highest frequency, improper layout could easily introduce too much crosstalk noise on them and cause failure. Coupling can be accumulated over large portions of a tree. Hence, the coupling effect for clock tree may be very significant.

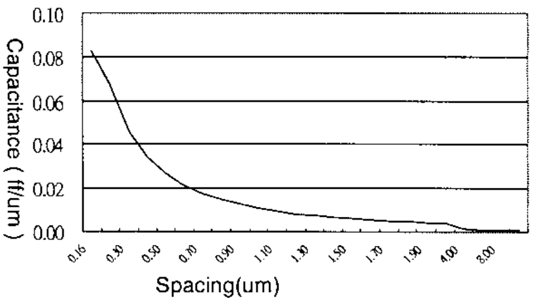


Fig. 1: The relationship of spacing and unit coupling capacitance for a typical 0.13um CMOS process

Failure cases have been reported due to the cross-coupling of clock signals. Therefore, it is important to deal with the cross coupling of clock signals during the clock routing phase.

Since the clock network is always synthesized after placement and before signal routing, it makes sense to minimize the mutual coupling during the clock synthesis phase. The smaller the effective coupling in this phase, the better clock quality will be.

In this paper, we deal with crosstalk minimization for the multiple clock routing problem. As far as we know, ours is the first attempt to attack this problem. The problem formulation is given in the following section, followed by the algorithm to generate the clock topology and routing in Sections 3 and 4. Experimental results and summary are given in section 5.

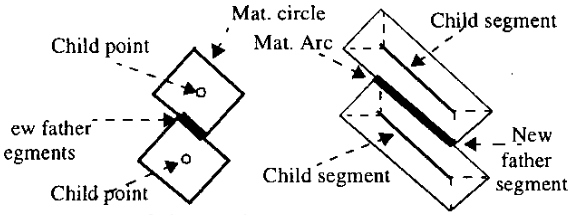
2. Problem Formulation

There are several factors which determine crosstalk noise. Some, such as the victim's driving strength and input capacitance, are related to the circuit design. Some, such as the slew or frequency of aggressor's signals, are related to the waveform of input stimulus. Analytical methods have been established to characterize the behavior of crosstalk noise[2][7]. The only factor affecting crosstalk and related to layout is the coupling capacitance, which is a function of spacing and coupling length. Fig. 1 shows a relationship between the unit coupling capacitance and the spacing for a typical 0.13um CMOS process. One can observe that the unit coupling capacitance drops dramatically with an increase in spacing. When the spacing is over 10um, then the effect of coupling capacitance becomes negligible. The crosstalk noise is *effective* only when the spacing is smaller than 10um. Here we define the effective coupling as follows.

Definition 1: **Effective coupling** is the coupling of two signals running in parallel and whose spacing is smaller than S_{max}. S_{max} is the

The second author acknowledges partial NSF support through grant CCR0098069.

0-7803-7523-8/02/$17.00 ©2002 IEEE



(a) Point to point (b) Segment to segment

Fig. 2: Merging segments for two points and two segments

threshold spacing value below which the effect of the crosstalk is not negligible.

Two signals can couple with each other only when their spacing is closer than S_{max}. The amount of parallel length within this spacing is the effective coupling length. The amount of the accumulated effective coupling length can measure the quality of layout. The longer the accumulated coupling length, the worse the signal will be.

Clocks usually have some constraints on the quality of signals they carry. Such constraints can be expressed in terms of delay, or may be just basic requirements on the noise margin. Crosstalk noise can affect the delay and signal level. So a good router must be able to avoid the accumulation of coupling capacitance, measured as the accumulation of effective coupling length. We define the crosstalk minimization for multiple clock routing problem as follows.

Problem 1: **Crosstalk Minimization for Multiple Clock Trees Routing:** Given a set of clock domains, T_1, T_2,..., T_n, with sink nodes for each, the problem is to construct these clock trees such that: (1) for each clock tree the total effective coupling length is minimized, (2) the zero skew in each clock tree is maintained for the given delay model, (3) the total wire length is minimized.

The second and third goals of this problem are typical of clock tree routing. We add one additional goal on the typical clock tree routing problem for crosstalk effects.

In this paper, we describe a two-phase approach to solve multi-clock routing problem. In the bottom-up phase, we decide the topologies and merging segments simultaneously, level by level, for all clocks. As a cost function we use the overlapping of routing area among these trees. At each level we formulate and solve a minimum cost perfect matching (MCPM) problem. In the second phase, after the topologies have been built, we assign the merging points, level by level, on the merging segments such that the clock trees have minimum effective coupling length. In each level, the problem is formulated as a placement problem which is solved by a simulated annealing approach.

3. Bottom-Up Phase

The collection of points with constant Manhattan distance to a given point forms a Manhattan circle. The shape of the Mahattan circle is a square with a 45 degree-slope. Given two points, the zero skew merging points are the intersection points of two Mahattan circles. The radius of the Mahattan circle depends on the delay model. It is assigned a length to maintain the zero skew. This length is kept and will be used in the top-down phase. The intersection of the Mahattan circles is a segment with slope=1 or slope=-1. It is called the Mahattan arc. It has been proven in [1][5] that for two given merging segments (Mahattan arcs), the zero skew merging points also form a Mahattan arc. An example is given in Fig. 2(b).

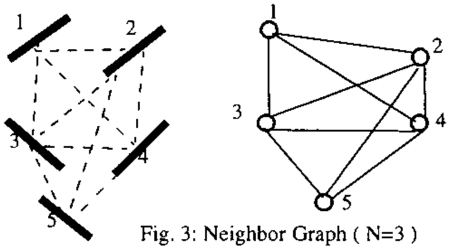


Fig. 3: Neighbor Graph (N=3)

The algorithm starts from a bottom-up phase and constructs the tree topology level by level. At each level, each node (can be a point at the bottom level or a Mahattan arc at the intermediate level) is matched to another node such that the overall cost function is minimized. This is essentially a matching problem. We define the graph in the following sections.

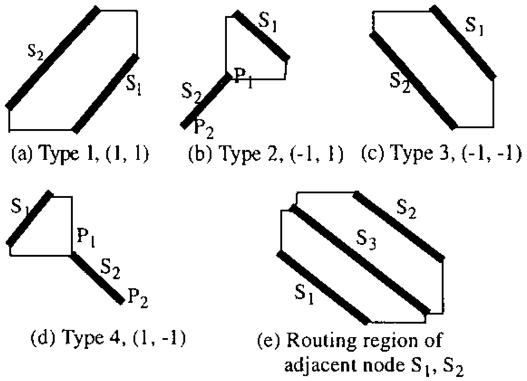
3.1 Neighbor Graph

One of the major goals for a clock routing algorithm is to generate a topology which has the minimum wire length. It makes sense to assume that each node can only be matched to some of its nearest neighbors. A constant N denotes the number of the nearest neighbors for each node.

Definition 2: Neighbor Graph: Each merging segment (or a point for sink nodes) corresponds to a node in the graph. An edge E_{ij} exists between V_i and V_j, if V_j is among the N nearest neighbors of V_i or V_i is one of the N nearest neighbors of V_j. The cost of an edge is defined in 3.3.

Definition 3: Distance Between the Segments: The distance between two segments is the minimum distance between two points on these segments.

An example is given in Fig.3; there, N=3. Some of the nodes might have more than 3 edges. For instance, node 2 is one of the three nearest neighbors of node 5, but node 5 is not one of the three nearest neighbors to node 2. So node 2 has four edges. So does node 4.



(a) Type 1, (1, 1) (b) Type 2, (-1, 1) (c) Type 3, (-1, -1)

(d) Type 4, (1, -1) (e) Routing region of adjacent node S_1, S_2

Fig. 4: Examples of routing regions

3.2 Routing Region

A routing region is defined as a potential routing area of two parent-child segments. Since the slope of merging segments can only be 1 or -1, there are only four types of routing regions that correspond to (1, 1), (-1, -1), (1, -1), and (-1, 1). These four types are illustrated in Fig 4. Note that in (b) and (d) one of the end points (P_2) of the segment doesn't contribute to the routing region because we assume S_1 is the father of S_2. So all the points on S_1 have equal Mahattan distance to P_1, and the distance is smaller than the distance to P_2. Thus only P_1 will be considered as the shortest merging point.

3.3 Edge Cost

An edge E in the neighbor graph implies a new parent merging segment for the two neighbors. For example, in Fig 4(e) if there is an edge between S_1 and S_2, a new merging segment S_3 is introduced corresponding to (S_1, S_2), and forms two routing areas, (S_1,S_3) and (S_2,S_3). The area is denoted by A(E). Suppose there are M trees, the routing area of the ith edge E_{ij} in graph G_j may overlap with the routing area of the kth edge E_{kl} in graph G_l. The intersection area of these two edges is denoted by $RC(E_{ij},E_{kl})$. And $RC(E_{ij},E_{kl})=0$ if j=l, because we are concerned with the overlapping of different clocks. We define the routing collision ratio for an edge as follows:

Definition 4: Routing Collision Ratio: Suppose there are M neighbor graphs G_1, G_2,..., G_M corresponding to M clocks. For an edge E_{ij} in graph G_j, the routing collision ratio of E_{ij} is defined as $ERC = \frac{1}{A(E_{ij})}\sum_k^{j \neq l} RC(E_{ij},E_{kl})$. It is the ratio of the overlapping area and its own area. And then we define the cost of an edge as $C(E) = \alpha \times K \times ERC + \beta \times L$ where α is the weight of the routing collision ratio, β is the weight of the wire length, L is the distance between its two descendant children, and K is the normalization factor for ERC and L. For conventional clock tree synthesis, α is set to 0, and β is set to 1, because the goal is to minimize the total wire length.

3.4 Minimum Cost Perfection Matching

Given M clocks, the topology of each clock is built level by level from bottom to top. At each level, M neighbor graphs are built simultaneously. And minimum cost perfect matching (MCPM) is performed for each tree. Below is the description of the bottom-up process.

```
Procedure Bottom-Up Phase
 while not reached the roots of all clock do
  Build neighbor graph for each clock
   Update the edge cost
   Do MCPM on each tree
  Calculate the parent merging from matching result
    Move one level up
End while
End procedure
```

The bottom-up merging procedure
Figure 6.

The output of the bottom-up phase is a binary tree of merging segments. In each intermediate node the distance to the leaves of its left subtree and the distance to the leaves of its right tree are always kept equal based on a given delay model.

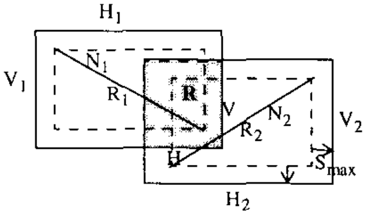


Fig. 5: Overlapping wires

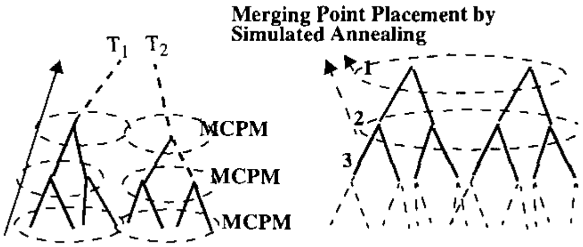
4. Top-Down Phase

After the bottom-up process, the topology is created, and every clock now is a tree of merging segments. The purpose of the top-down phase is to determine the exact location of each merging point such that the overall coupling length can be minimized. The result of the top-down phase is a binary tree with a placed merging point. Any edge on the tree represents a two-pin net. That means the entire tree can be expressed by a set of two-pin nets.

4.1 Cost Function for Top-Down Phase

In general, the coupling capacitance can be decided only after the real wires are placed. For a two-pin net to be routed, a new cost function is necessary at the merging embedding stage to estimate how good it will be after routing. That will help to choose better merging points. The cost function should reflect the expected value of effective coupling length before routing. We will derive the expected value of effective coupling length from a probabilistic point of view. The expected effective coupling length for two-pin net to two-pin net is defined as follows.

For two two-pin nets $N_1 = (P_1(X_0,Y_0), P_2(X_1,Y_1))$, $N_2 = (P_3(X_2,Y_2), P_4(X_3,Y_3))$, if we extend the bounding box of N_1 and N_2 by S_{max} both in horizontal and vertical direction, then N_1 will span a rectangle R_1 with width $|X_1-X_0|+2*S_{max}$ and height $|Y_1-Y_0|+2*S_{max}$, similarly N_2 will span a rectangle R_2 with width $|X_3-X_2|+2*Smax$ and height $|Y_3-Y_2|+2*S_{max}$. We denote the height and width of R_1 by V_1 and H_1, the height and width of R_2 by V_2 and H_2 respectively. And these two rectangles overlap a rectangle R with height V and width H as shown in Fig 5. If we assume N_1 is bounded by R_1, and N_2 by R_2, the expected vertical coupling length will be $V_{ecl} = V \times \frac{H}{H_1} \times \frac{H}{H_2} \times \frac{2 \times Smax}{H}$, where V is the length of the possible vertical coupling, H/H_1 is the probability that a vertical wire of R_1 falls into R, H/H_2 is the probability that a



(a) Bottom-up phase
(b) Top-down phase
Fig.7.

vertical wire of R_2 falls into R, and $2xS_{max}/H$ is the probability that two vertical wires are within S_{max} distance. Similarly, the expected horizontal coupling would be $H_{ecl} = H \times \frac{V}{V_1} \times \frac{V}{V_2} \times \frac{2 \times S_{max}}{V}$. The expected effective coupling length for N_1 and N_2 is $ECL(N_1)=ECL(N_2) = V_{ecl}+H_{ecl}$.

4.2 Placement of Merging Point by Simulated Annealing

Once the tree of segments is built, we use a top-down approach to decide the exact merging points of each internal segment such that the overall expected coupling length among different clock trees is minimized. The process starts from the top level. At each level, we decide the exact merging point of all clock trees at that level by a simulated annealing approach. During the simulated annealing, the moving decision is based on the change of ECL (cost) from the current point to the new point. When a point is moved along a merging segment, its corresponding ECL is recalculated based on the formula given in Section 4.1. The whole top-down process is summarized in Fig8. Fig 7. shows the whole process.

```
Procedure Top-Down Assignment
Initialize the merging point for all trees;
While not reach bottom level for all trees do
  for ( i=0; i< Number of tree; i++)
    Extract segments of tree i at CurrentLevel
  end for
  T = Initial Temperature
  While ( T >= Minimum Temperature ) do
    Move = 0;
    While ( Move < Move_Per_Temperature )
        S = Pick One Segment Radomly at curent level
      $P_{new}$ = Pick one point on S randomly
      if ( ECL($P_{new}$) < ECL($P_{old}$) or
                        P <= $e^{Alpha(ECL New-Old)/T}$
          Move merging point on S from $P_{old}$ to $P_{new}$
    End While
  End While
  Move one level down
End While
```

Fig.8 Top-Down Assignment by Simulated Annealing

5. Experimental Results

We have implemented our algorithm using GNU C++ running under Linux operating system. Since there are no existing benchmarks for this problem, we took five randomly generated examples and used them to test the algorithm. All the examples have two clocks with the same number of sink nodes. To compute ERC, a 2D polygon operation program [4] is used to calculate the intersections of routing regions. In the bottom-up phase, we use the program developed by [3] to solve the MCPM problem.

Suppose there are M clock signals, $T_1, T_2, \ldots, T_M$, with routing area $A_1, A_2, \ldots, A_M$. We measure the ratio of overlapping area with respect to total area $A_1+A_2+\ldots A_M$ at the first stage. For comparison, we use two sets of parameters as described in section 3.3, L(α=0.2 β=0.8), and C(α=0.8 β=0.2). L parameter tends to minimize wire length, and C parameter tends to minimize effective coupling. The bottom-up phase can always significantly decrease this ratio, from 23% to 60% in the tested examples. The result is shown in Table 1. To evaluate the output (Effective Coupling Length L_{eff}) of the second stage, we use the routing heuristic proposed in [6]. In this paper, all two pin connections are laid out using only two patterns, an upper L-bend wire or a lower L-bend wire. The total coupling capacitance is calculated after the clock trees are actually routed. C_{couple} is the summation of inter-clock capacitance and calculated based on 0.13um process characterized in Fig. 1. The result in table 2 shows about 16% to 35% decrease in coupling capacitance without significantly increasing the wire length of the trees. All the five examples are based on a 512x512 grid size.

TABLE 1. Ratio of Overlapping

Ratio	#node	$\alpha = 0.2$ $\beta = 0.8$	$\alpha = 0.8$ $\beta = 0.2$	Dec%
T1024	1024	0.274	0.128	53.3%
T512	512	0.262	0.201	23.3%
T256	256	0.330	0.125	62.1%
T128	128	0.314	0.168	46.5%
T64	64	0.281	0.187	33.5%

TABLE 2. C_{couple}(C for α=0.8, β=0.2 L for α=0.2,β=0.8)

	L_{eff}	Length	C_{couple}FF	Dec%
T1024C	54042	113251	834.36	25.8%
T1024L	67736	113349	1124.29	
T512C	23804	37693	421.94	27.2%
T512L	25566	38103	579.19	
T256C	7580	26824	184.11	16.8%
T256L	9276	26275	221.28	
T128C	4346	19324	131.12	25.2%
T128L	6436	18835	175.34	
T64C	3258	13160	39.46	35.4%
T64L	4438	12958	61.06	

Reference

[1] T.H. Chao, Y.C. Hsu, J.M. Ho, K.D. Boese, and A.B. Khang, "Zero Skew Clock Routing With Minimal Wirelength", IEEE Trans. CAS, Vol.39, No.11, NOV 1992, pp.799-814.

[2] L.H Chen, and M. Marek-Sadowska, "Aggressor Alignment for Worst-Case Crosstalk Noise", IEEE Trans. Computer-Aided Design, Vol.20, No.5, May 2001, pp.612-621.

[3] W. Cook and A. Rohe, "Computing minimum-weight perfect matchings", INFORMS Journal on Computing, Vol. 11, 1999, pp. 138-148.

[4] K. Holwerda , "http://www.xs4all.nl/~kholwerd/bool.html"

[5] D. J.-H. Huang, A. B. Kahng, and C.-W. A. Tsao, "On the Bounded-Skew Clock and Steiner Routing Problems", Proc. 32nd ACM/IEEE Design Automation Conf., San Francisco, June 1995, pp. 508-513.

[6] R. Kastner, Bozorgzadeh, E.; Sarrafzadeh, M. "Coupling Aware Routing", ASIC/SOC Conference, 2000. Proceedings. 13th Annual IEEE International , 2000, pp. 392-396.

[7] A. Vittal, L.H. Chen, M. Marek-Sadowska, K.P. Wang, and S. Yang, "Crosstalk in VLSI Interconnections", IEEE Trans. Computer-Aided Design, Vol.18, No.12, Dec. 1999, pp.1817-1824.