# Clock Routing Synthesis for Nanometer IC Design

**Xianzhe Jin and Kwangki Ryoo**, *Member, KIMICS*

***Abstract***—Clock skew modeling is important in the performance evaluation and prediction of clock distribution network and it is one of the major constraints for high-speed operation of synchronous integrated circuits. In clock routing synthesis, it is necessary to reduce the clock skew under the specified skew bound, while minimizing the cost such as total wire length and delay. In this paper, a new efficient bounded clock skew routing method is described, which generalizes the well-known bounded skew tree method by allowing loops, i.e., link-edges can be inserted to a clock tree when they are beneficial to reduce the clock skew and/or the wire length. Furthermore, routing topology construction and wire sizing is used to reduce clock delay.

***Index Terms***—Routing, Topology, Clock, Synthesis

## I. INTRODUCTION

The evolution of VLSI chips toward larger die sizes and faster clock speeds makes clock distribution an increasingly important issue and the reduced feature size of VLSI devices makes possible to design complex deep submicron integrated circuits [1]. In deep submicron technology, interconnect delay represents the major portion of the total delay [2]. As the clock frequency of digital systems approaches 1GHz, low-skew clock routing becomes an important issue, since the critical speed is limited by the critical path delay and clock skew [3]. The clock skew is the maximum clock misalignment, i.e., the maximum difference in the clock arrival times. Recently, many clock routing techniques have been reported. They can be classified into two typical sets of approaches, which are based on the "balanced-tree method (BTM)[4-11]" and "mesh method [12-13]".

There have been many researches on the clock routing algorithm regarding BTM. In BTM, a clock signal is routed as a tree so that the delay times of clock signal are balanced. H-tree [4] has been recognized for a long time as a technique to reduce the skew in synchronous systems. As shown in Fig. 1(a), H-tree works well to reduce the skew for regular structures such as systolic arrays. But, asymmetric distributions of clock pins are common in general cases and the H-tree may not be effective for clock routing. Moreover, the total wire length can be excessively long, impacting critical area, power consumption, and clock rates for large circuits.

The generalization of H-tree that hierarchically divides into median and connects the mean points like Fig. 1(b) is proposed in [5]. A further improvement is done by bottom-up pairwise connection, which constructs a perfect length balanced clock tree[6] as shown in Fig. 1(c). The Fig. 1(d) shows deferred merge embedding. It calculates the wire segments that satisfy the given skew bound. However, all these heuristics focus only on the wire length balanced clock tree. Although wire length is perfectly balanced, clock skew remains non-zero because the timing delay is affected by load capacitance, interconnect resistance, and capacitance according to the routing patterns. To improve this problem, exact zero skew algorithms using Elmore delay model is proposed [7]. In [8] and [9], the merging segment(the line segment having zero skew) is introduced and the interconnect point is determined on the minimum distance position of the segment in order to minimize wire length. Generally, the more the clock skew increases, the more the total wire length is reduced [10][11].

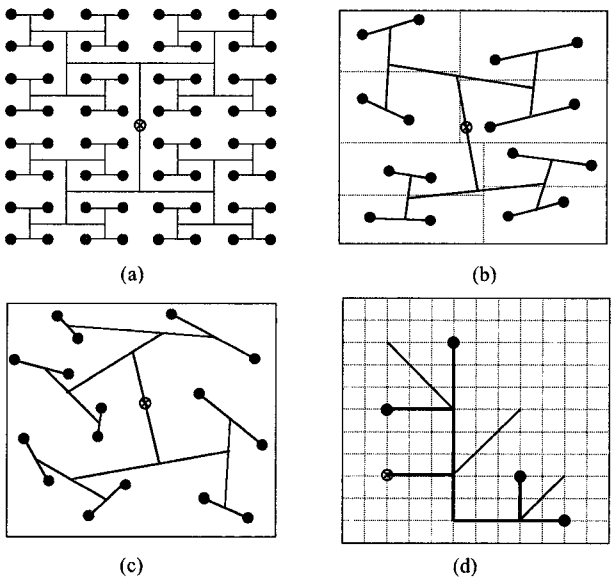


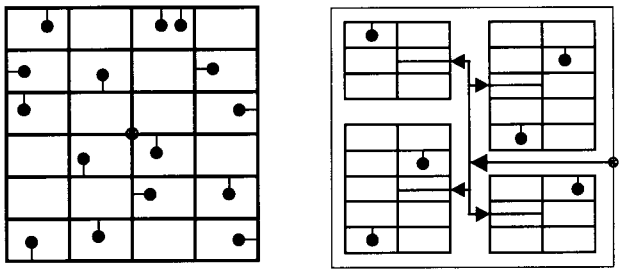

Fig. 1 Balanced tree method (a) H-Tree (b) method of means and median (c) geometric matching (d) deferred merge embedding.

Manuscript received August 5, 2008; revised October 15, 2008. Xianzhe Jin and Kwangki Ryoo is with the Department of Information and Communication Engineering, Hanbat National University, Daejeon, 305-719, Korea (Tel: +82-42-821-1710, Fax: +82-42-821-1595, Email: kkryoo@hanbat.ac.kr)

If the clock skew of the synchronous system is within the 5% range of clock period, the circuit operates correctly in the real world, which is a very tight design constraint. To limit the clock skew, a bounded skew clock routing algorithm is proposed in [10] and [11]. The merging region, the set of inner points that satisfy the bound of clock skew, is calculated and the minimum distance interconnection is found.

Recently, several clock routing techniques using mesh topology have been proposed. In the fixed mesh method (FMM)[12], the clock net is routed as a fixed mesh driven by a large buffer [12]. FMM is easy to route and only one track is required in each channel, which means the routing area is predictable, as shown in Fig. 2(a). But, FMM has disadvantages in that it needs large routing resources and it consumes too much power.

The balanced mesh method (BMM)[13] incorporates the advantages of BTM and FMM(Fig 2(b)). However, these routing techniques based on the mesh method produce large total wire length, need many or big buffers, and dissipate large power.



(a) (b)

Fig. 2 Mesh method (a) FMM (b) BMM

In this paper, we present a bounded clock skew optimization algorithm using the generalized graph topology. A new skew reduction technique by adding and removing a link-edge on the routing topology is proposed. The proposed algorithm combines the advantages of BTM and BMM. The balanced routing tree is constructed and link-edges are iteratively added to reduce the clock skew.

Also, in order to reduce source-to-sink delay, routing topology construction and wire sizing are performed by using a new width selection function, is performed.

## II. DELAY MODEL

As the feature size decreases to the submicron dimension, the wire resistance is no longer negligible. In order to consider both wire resistance and capacitance, the interconnect is modeled as an RC tree. In the lumped RC model, R refers to the resistance of driver and C refers to the sum of the total interconnect capacitance and the total loading capacitance. This model assumes that wire resistance is negligible, which is generally true for designs with feature sizes of 1.2μm. But under the deep submicron IC design, distributed RC delay model must be used.

As in most previous research on interconnect layout optimization, we adopt the Elmore delay model. For a wire e, let $l_e$, $w_e$, $c_e$, and $r_e$ denote the length, width, capacitance, and resistance, respectively. The $r_e$ and $c_e$ is formulated as follows.

$$r_e=\frac{r_o l_e}{w_e},\ c_e=(c_o w_e+c_f)l_e$$

where, $r_o$ is resistance of unit length, $c_o$ and $c_f$ are area capacitance and fringing capacitance for unit area and length, respectively.

Also let e denote the wire entering the node i from its parent. The delay of wire segment, D(e) is formulated as follows.

$$D(e)=r_e\cdot(\frac{c_e}{2}+c(T_i))$$

where, $T_i$ is the subtree rooted at node i and $c(T_i)$ is the capacitance of $T_i$ . The equivalent π-model for a wire $e_i$ is shown in Fig. 3. Under the Elmore delay model, the signal delay from source s to node i in an RC tree is calculated by using the following equation.

$$D(s,i)=\sum_{k\in path(s,i)} r_{e_k}\cdot c(T_k)$$

where, path(s,i) is the unique path from the source s to the node i in an RC tree, $e_k$ is the wire entering the node k, $r_{ek}$ is the resistance of $e_k$, and $c(T_k)$ is the total capacitance of the subtree rooted at node k.

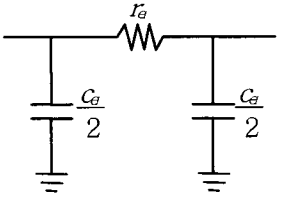



Fig. 3 Equivalent π-model for a wire

## III. PROPOSED ALGORITHMS

### *A. Link-edge insertion by HSPICE*

The main idea of proposed clock skew optimization algorithm is based on the link-edge insertion technique. A link-edge is inserted repeatedly between two nodes of a clock tree/graph whose delay difference is "large" and the distance between them is "small". When the appropriate edge is inserted, the delay of a node that has the longer delay is decreased and the clock skew can be reduced.

As shown in Fig. 4, if a link-edge is inserted between two nodes in the clock tree, a loop is constructed and the delay of a node that has the larger delay is decreased and

the delay of a node that has the smaller delay is increased, if the length of the link-edge is "short". As a result, the clock skew is decreased. For the delay calculation of a node in the clock tree, equivalent π-model is used. In Fig. 5, the simulation models before and after a link-edge insertion are shown.

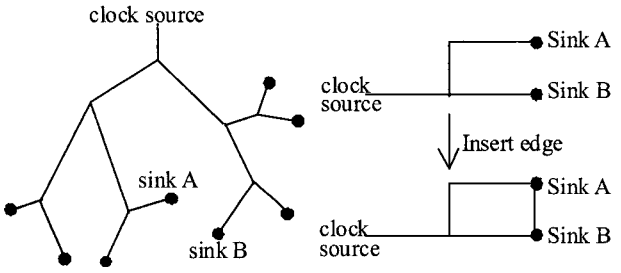



Fig. 4 Link-edge insertion

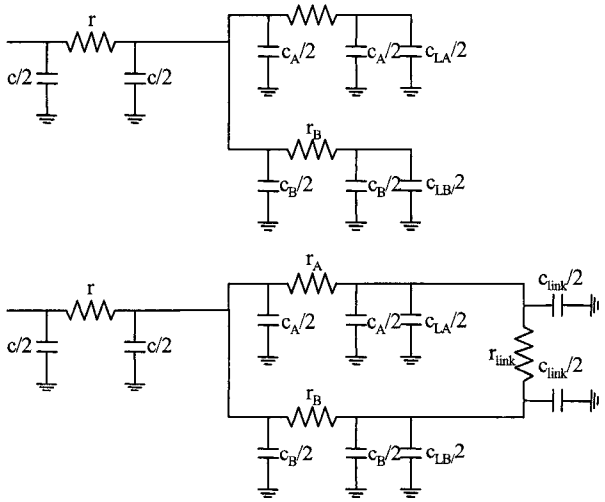



Fig. 5 π-model before/after a link-edge insertion

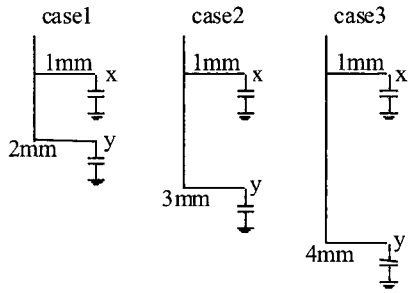



Fig. 6 Three cases of test examples

The HSPICE simulations on three cases of test circuits shown in Fig. 6 are performed in order to prove the effectiveness of the link-edge insertion. For the simulation, clock frequency of 100MHz, load capacitances of load 0.0525pF, unit resistance of 0.03Ω/m, unit capacitance of 0.0002 pF/cm are used. Assume that the length of a sink x is 1mm and the lengths of y are 2mm, 3mm, and 4mm, respectively, as shown in Fig. 6.

In each of the three cases, a link-edge which has the wire length of 1mm or 2mm is inserted and HSPICE simulation is performed in order to calculate the delay of the sink x and y. After the link-edge of length 1mm or 2mm is inserted for the third test circuit, the clock skew is reduced by 76.6% and 53.5% respectively. The comparisons of time delays are shown in Table 1 and the changes of the clock skews are shown in Table 2 after link-edge insertion.

Table 1 The comparison of timing delays

| | length of link | delay | case 1 | case 2 | case 3 |
|---|---|---|---|---|---|
| before link | no link | $T_x$ | 6.95 | 7.13 | 7.01 |
| | | $T_y$ | 24.40 | 52.07 | 89.60 |
| after link | 1mm | $T_x$ | 18.26 | 22.98 | 27.14 |
| | | $T_y$ | 23.69 | 35.30 | 46.41 |
| | 2mm | $T_x$ | 22.80 | 26.29 | 30.98 |
| | | $T_y$ | 34.97 | 51.24 | 69.38 |

From the Table 1, one may confirm that the delay of sink y is reduced and the delay of sink x is increased. But it must be noted that the delay of sink y is increased in case when the length of the link-edge is long. Therefore it is inefficient when the distance between two nodes is relatively long.

Table 2 The comparison of clock skews

| | length of link | case 1 | case 2 | case 3 |
|---|---|---|---|---|
| before link | no link | 17.454 (1.000) | 42.940 (1.000) | 82.587 (1.000) |
| after link | 1mm | 5.430 (0.311) | 12.319 (0.287) | 19.280 (0.234) |
| | 2mm | 12.171 (0.697) | 24.953 (0.581) | 38.397 (0.465) |

From the above results, the link-edge insertion technique is an effective method to reduce the clock skew when the delay difference is large.

### *B. Link-edge insertion algorithm*

From the previous section, we confirm that the effect of link-edge insertion is desirable when the delay difference between two nodes is large and the distance between two nodes is small. Two nodes to be linked are selected by the link-cost function which is the measure of the linking effect. The link-cost function used in the proposed algorithm is defined as follows:

$$LinkCost(i,j)=\frac{Distance(i,j)}{DelayDifference(i,j)}$$

where, i and j is the indices of nodes in the clock tree. The value of the link-cost function is small if

the distance is small and delay difference is large. Two nodes having the smallest value of link-cost have the larger possibility to be linked. The proposed link-edge insertion algorithm is shown in Fig. 7. At first, sort the delay of nodes in decreasing order and construct the node set, SMD(the set of maximal delay). The number of nodes in SMD is determined by a user-defined parameter. Because the effect of linking is not desirable when the distance between two nodes is long, the nodes which are located in the outer region of radius r from the selected node i are neglected. Experimentally, the value of radius r is determined to be the 10-20% of chip dimension. In the outer while loop, it selects a node i that has the maximum delay in the SMD and the find the node set, SIR(the set of inner region nodes) whose elements are located in the inner region of radius r from the selected node i.

```
Find node set SMD;
While (SMD is not empty){
  Select a node i that has the maximum delay in SMD;
  Remove node i from the SMD;
  Find node set SIR whose element is in the inner
  region of radius r from node i;
  While (SIR is not empty){
      Select a node j that has the min linkCost(i,j);
      Remove node j from SIR;
      Insert link-edge between node i and node j;
      Delete an edge on the loop containing edge e_ij;
      If(clock skew is reduced){
        Relocate merging point;
        Update SMD and break;
      }Else
        Undo insertion of link-edge;
  }/*while*/
}/*while*/
```

Fig. 7 Link-edge insertion algorithm

In the inner while loop, it selects a node j that has the minimum *link-cost(i,j)* and inserts *link-edge(i,j)* into the clock tree. When the edge insertion causes a loop, it is not possible to use Elmore delay model. Therefore HSPICE simulation is used. In some cases, the effect of linking is reduced as the resistance and capacitance are increased by edge insertion itself. In order to prevent resistance and capacitance from increasing, an edge on the loop containing the link-edge $e_{ij}$ is deleted. The edge to be deleted is chosen to produce the smallest delay difference between the two end points. When the clock skew is not reduced by link-edge insertion or deletion, the inner loop is repeated for the next element of the *SIR*. If *SIR* is empty, the outer loop is repeated on the next element of *SMD* until *SMD* is empty.

### *C. Relocation of merging point*

During link-edge insertion and deletion, relocation of merging point is performed to minimize the total wire length. The merging point is relocated within a given skew bound, as shown in Fig. 8. The distance of displacement is calculated as follows. In Fig. 9, assume that the wire length is reduced if merging point is moved to node1 and the distance between node1 and node2 is $l$, then the position of zero skew point $x$ and the displacement $x'$ under the given skew bound are calculated by the following equation.

$$x=\frac{r_o l(\frac{c_o l}{2}+c_2)+t_2-t_1}{r_o l(c_1+c_2+c_o l)}, \quad x'=\frac{r_o l(\frac{c_o l}{2}+c_2)+t_2-t_1-B}{r_o l(c_1+c_2+c_o l)}$$

where, $t_1$ is the delay from node 1 to its sink, $t_2$ is the delay from node 2 to its sink, and $c_1$ and $c_2$ are the capacitances of subtrees at node 1 and node 2, respectively. B is the given skew bound.

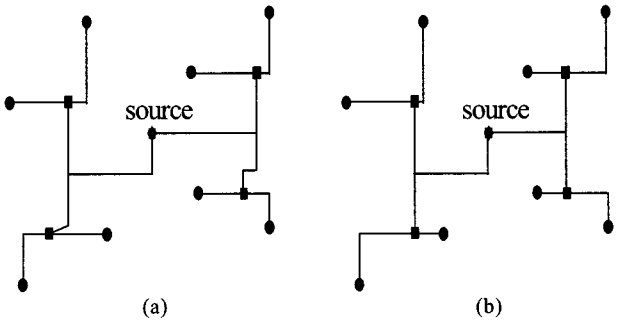



Fig. 8 Relocation of merging point (a) before (b) after the relocation of merging point

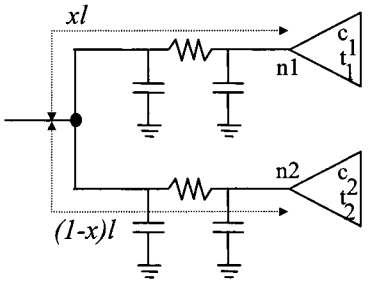



Fig. 9 Calculation of merging point

Therefore the maximum displacement of merging point is $\Delta x = x - x'$. Moving the merging point toward the direction of reduction can reduce the wire length.

The proposed algorithm is implemented on an IBM compatible PC by using the ANSI C programming language. The benchmark test data r1, r2, and r3 are used to evaluate our algorithm for the skew bounds in the range of 10-1000 ps. In order to satisfy the given skew bound, our algorithm starts with the 10% larger skew bound. As shown in Table 3, the results are obtained by both the link-edge insertion and

merging point relocation and finally, the total wire length is reduced by 3%, on the average, than that of the balanced skew tree [11].

Table 3 The comparison of total wire length

| Skew bound | | 10 | 100 | 1000 |
|---|---|---|---|---|
| r1 | BST | 1087703 | 926205 | 793498 |
| | Ours | 1070183 | 913127 | 774062 |
| r2 | BST | 2155481 | 2201023 | 1839665 |
| | Ours | 2132178 | 2171096 | 1768748 |
| r3 | BST | 2789321 | 2515178 | 2506399 |
| | Ours | 2750372 | 2445204 | 2453266 |

### *D. Routing topology construction*

In this section, the routing topology construction problem for the delay minimization is described. One of the well known paper on the bounded skew problem is by J. Cong[11]. In order to construct a routing tree in [11], two nodes that have the shortest distance are merged iteratively and the subtree capacitance between the two nodes is ignored. Considering only the distance between two nodes may unbalance the final routing tree. If the difference of capacitance is large, the region satisfying the given skew bound may not exist and detouring wire must be added to decrease the difference of capacitance.

As a result, the total wire length and the propagation delay are increased. In the proposed routing topology construction algorithm, the distance between the two nodes, subtree capacitance, criticality of sink and the distance between the source and the merging region are considered. The maximization of slack time at each sink is set as an objective function. As the slack time is maximized, actual arrival time at a sink is decreased, the clock period can be reduced, and it is possible to design clock distribution network in high-performance ICs. In our algorithm, iterative merging scheme is applied to construct a binary routing tree. In order to select two nodes to be merged, the merging cost function is defined as follows.

$$MergingCost(i,j)=\frac{Dist(i,j)+\alpha\cdot Capa(i,j)+\lambda(i,j)}{MergingPoint(i,j)}$$

where, Dist(i,j) is the Manhattan distance between the two nodes, i and j. In the case when subtree contains many nodes, it has the minimum distance from a node to the minimum rectangle that contains all nodes in the subtrees. Capa(i,j) is the sum of capacitance of node i and j because subtree i and j are connected parallel on the ground. $\lambda$(i,j) has the maximum value among the criticality values in subtree i and subtree j. MergingPoint(i,j) is the shortest distance between the clock source and the minimum rectangle that contains all nodes in subtree i and j. Because two nodes that have minimum merging cost are selected, two nodes that have short distance and small sum of capacitances are merged firstly.

If the node having large capacitance is merged at first, the level of that node is large and the propagation delay from the source becomes large. Therefore, the node having long distance from the source is merged first. This is because the wire length may be increased in order to interconnect the node that is far from the source if the node near the source is merged first.

Table 4 Parameters used in experiment

| Circuit | IC1 | IC2 | IC3 |
|---|---|---|---|
| Technology[μm] | 2.0 | 1.2 | 0.5 |
| $R_d$[Ω] | 164 | 212.1 | 270.0 |
| $R_o$[Ω/μm] | 0.033 | 0.073 | 0.112 |
| $C_o$[F/μm] | 0.019 | 0.022 | 0.039 |
| $C_i$[F] | 5.7 | 7.06 | 1.0 |
| Size[mm] | 10x10 | 10x10 | 10x10 |

Table 5 Comparison of delay and length (n=5)

| Circuit | Comp. | IC1 | IC2 | IC3 |
|---|---|---|---|---|
| Maximum Delay [ns] | I1S | 0.267 | 0.393 | 0.262 |
| | Ours | 0.114 | 0.174 | 0.249 |
| Wire Length [μm] | I1S | 1.48 | 1.48 | 1.48 |
| | Ours | 1.64 | 1.64 | 1.64 |

Table 6 Comparison of delay and length (n=9)

| Circuit | Comp. | IC1 | IC2 | IC3 |
|---|---|---|---|---|
| Maximum Delay [ns] | I1S | 1.028 | 0.664 | 0.444 |
| | Ours | 0.722 | 0.544 | 0.443 |
| Wire Length [μm] | I1S | 2.18 | 2.18 | 2.18 |
| | Ours | 2.499 | 2.499 | 2.499 |

The proposed algorithm is compared with iterative 1-Steiner algorithm (I1S). The parameters used in the experiment are as shown in Table 4. $R_d$ is driver resistance, $R_o$ and $C_o$ is the unit resistance and capacitance respectively. $C_i$ is the load capacitance of sink. From the Tables 5 and 6, the propagation delay is decreased by 43% and total wire length is increased by 12%.

### *E. Wire sizing*

In the deep sub-micron design, adjusting the wire width can minimize the propagation delay. For the clock trees, we have developed an algorithm to minimize delay times, while satisfying the given skew. In this section, the wire width design to increase the speed of a circuit is suggested. It reduces the delay from the source to the sink by adjusting the wire segment in the given clock distribution tree, while it doesn't greatly increase the area due to the increased wire width. The proposed

algorithm is as follow; First, the algorithm calculates the delay from the root node to each sink in the routing tree for the given routing topology. After the longest path, $R_m$, is determined, the wire widths of the segment in the longest path are increased. Let the longest path be (r, $a_1$, $a_2$, ..., $a_n$). Here r is the root, and $a_n$ is a leaf node. For partial paths (r, $a_1$, $a_2$, ..., $a_k$, k = 1,2,...,n), the segment gains are computed. The wire width of the maximum segment gain is increased. The wire segment is selected by using the segment gain function;

$$SegmentGain(s_i)=\frac{\Delta D-\alpha\cdot L(s_i)}{\Delta W}$$

where, $\Delta D$ is the reduction of the delay when increasing the width of $s_i$ by $\Delta W$ and L is the wire length of the segment $s_i$. As the wire length becomes longer, the chip area due to the wire increases and the capacitance due to the adjustment of the wire width also increases. Therefore, negative sign is attached to the second term. The constant $\alpha$ is the weight of the wire length on the variation of the delay time and $\Delta W$ is the variation of the wire width. When selecting a wire segment, the segment that has the maximum value of the segment gain function is selected in order to adjust the wire width. In other words, when increasing the unit wire width, the segment that reduces the delay time as much as possible and that has a small wire length is selected as a candidate segment. If the delay is not improved or the given skew bound is not satisfied during the iterative procedure, the wire sizing is finished.

In the experiment, 50~500 pins are randomly placed within the 10 x 10 $mm^2$ square region. For the experiments, the resistance of source gate of 270Ω, wire resistance per unit length of 0.12Ω/μm, the wire capacitance per unit area of 0.04 fF/μm, the fringing capacitance of 0.15 fF/μm, and the minimum wire width of 0.5μm are used. The load capacitance of a sink has the values between 0.05 pF and 0.15 pF. Assume that wire width has one of the three values which values of which are w, 2w, and 3w, where w is the minimum wire width. The skew range is set to be 10 ps and the pins of 50, 100, 200, and 500 are randomly placed. The results after adjusting the wire width are shown in Table 7. The results are the average value of the delays of the experiments in which the sink positions are randomly generated 10 times. We find that the delay is reduced by 12.84%, on the average, by using wire sizing.

Table 7 Comparison of delay

| Pin | Initial delay | Final delay | Ratio |
|---|---|---|---|
| 50 | 343.2 | 312.41 | -8.97% |
| 100 | 818.5 | 716.08 | -12.5% |
| 200 | 1196.3 | 1053.91 | -11.9% |
| 500 | 1923.7 | 1576.86 | -18.2% |

### *F. Overall algorithm*

The proposed topology construction, link-edge insertion, merging point relocation, and wire sizing algorithms are combined into an integrated program. The overall design flow of the proposed clock routing system is as shown in Fig. 10. The inputs of routing topology construction are two files. One file contains the position of each clock pin, the capacitance of each sink, and the criticality of sink. The other file contains parameters such as chip dimension, source position, unit capacitance, unit resistance, and fringing capacitance. The output of this stage is the clock routing tree that satisfies a given clock skew. Our algorithm starts with 10% larger initial skew. It is because the total wire length is reduced as the clock skew is increased. The proposed routing topology construction algorithm controls the order of merging sequence by considering the pin position, the criticality of sink, and the capacitance of subtree.

In the next stage (link-edge insertion and merging point relocation stage), the clock skew is decreased to the given skew bound by link-edge insertion. It moves a merging point within given skew bound from zero-skew point to the direction of reducing the total wire length. Finally, a new wire sizing algorithm is proposed to minimize the delay by adjusting the width of wire segments.

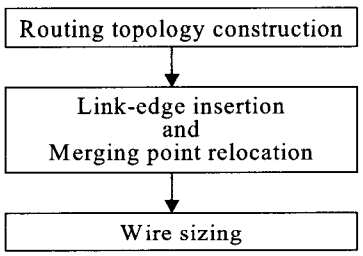



Fig. 10 Overall design flow

Table 8 shows the experimental results of our program which combines topology construction, link-edge insertion, and wire sizing. The numbers of inserted link-edges are 8, 13, 23, and 32 for rl, r2, r3, and r4, respectively. The timing delay is reduced by 13.93% on the average than that of the bounded skew clock routing method proposed in [11].

Table 8 Comparison of delay

| Circuit | Cong[11] | Ours | Ratio |
|---|---|---|---|
| rl | 1635.48 | 1445.43 | -11.62% |
| r2 | 2562.74 | 2194.21 | -14.38% |
| r3 | 3284.02 | 2888.95 | -12.03% |
| r4 | 6041.36 | 4969.62 | -17.74% |

The final routing tree of test data r2 is as shown in Fig. 11. Our program reports the circuit name, parameter file, delay time, and total wire length in the right side of the window.

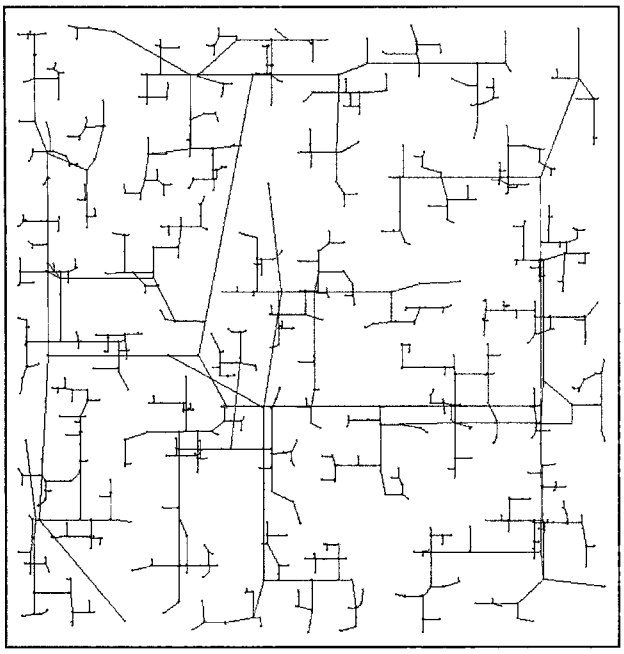


Fig. 11 Final routing tree of r2

## VI. CONCLUSIONS

In this paper, we have presented a new efficient algorithm for bounded clock skew routing using link-edge insertion. The effect of the link-edge insertion is proved through HSPICE simulation.

From the experiments on the benchmark test circuits, it is shown that our results satisfy the given skew bound and the total wire lengths are reduced. The proposed algorithm constructs a new clock routing topology which is a generalized graph model, while most previous methods use only the tree-structured routing topology. Also, in order to reduce the source-to-sink delay, wire sizing using a new width selection function is performed. For the practical application, we also plan to work on buffer insertion and clock routing optimization for low power consumption.

## ACKNOWLEDGMENT

This work was partly supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD)(KRF-2004-003-D00259) and 2003 Hanbat National University research fund. The related CAD tools were supported by IDEC.

## REFERENCES

[1] Xiaohong Jiang and Susumu Horiguchiu, "Statistical Skew Modeling for General Clock Distribution Networks in Presence of Process Variations," IEEE Trans. On VLSI Systems, Vol. 9, No. 5, pp. 704-717, Oct 2001.

[2] Eby G. Friedman, "Clock Distribution Networks in Synchronous Digital Integrated Circuits," Proceedings of the IEEE, Vol. 89, pp. 665-692, May 2001.

[3] Vojin G. Oklobdzija, "Multi-GHz Systems Clocking," 5th International Conference on Application Specific Integrated Circuits, Vol. 2. pp. 701-706, Oct 2003.

[4] Hiroki Matsutani, Michihiro Koibuchi, and Hideharu Amano, "Performance, Cost, and Energy Evaluation of Fat H-tree: A Cost-Efficient Tree-Based On-Chip Network," IEEE International Parallel and Distributed Processing Symposium, pp. 1-10, Mar 2007.

[5] Michael A. B. Jackson, Arvind Srinivasan, and E. S. Kuh, "Clock Routing for High-Performance ICs," Proc. ACM/IEEE Design Automation Conference, pp.573-579, 1990.

[6] Andrew B. Kahng, Jason Cong, and Gabriel Robins, "High-Performance Clock Routing Based on Recursive Geometric Matching," Proc. ACM/IEEE Design Automation Conference, pp.322-327, 1991.

[7] Ren-Song Tsay, "Exact Zero Skew," Proc. IEEE Int'l Conf. on Computer-Aided Design, pp.336-339, 1991.

[8] Kenneth D. Boese and Andrew B. Kahng, "Zero-Skew Clock Routing Trees with Minimum Wire Length," Proc. IEEE 5th Int'l ASIC Conf., pp.1.1.1-1.1.5, 1992.

[9] Masato Edahiro, "A Clustering-Based Optimization Algorithm in Zero-Skew Routings," Proc. ACM/IEEE Design Automation Conference, pp.612-616, 1993.

[10] Dennis J.-H. Huang, Andrew B. Kahng, and Chung-Wen Albert Tsao, "On the Bounded-Skew Clock and Steiner Routing Problem," Proc. ACM/IEEE Design Automation Conference, pp.508-513, 1995.

[11] Jason Cong, Andrew B. Kahng, Cheng-Kok Koh, and C.-W. Albert Tsao, "Bounded-Skew Clock and Steiner Routing Under Elmore Delay," Proc. IEEE Int'l Conf. on Computer-Aided Design, pp.66-71, 1995.

[12] D. W. Dobberpuhl, R. T. Witek, R. Allmon, R. Anglin, D. Bertucci, S. Britton, L. Chao, R. A. Conrad, D. E. Dever, B. Gieseke, S. M. N. Hassoun, G. W. Hoeppner, K. Kuchler, M. Ladd, B. M. Leary, L. Madden, E. J. MeLellan, D. R. Meyer, J. Montanaro, D. A. Priore, V. Rajagopalan, S. Samudrala, and S. Santhanam, "A 200-MHz 64-bit dual-issue CMOS micro-processor," IEEE Journal of Solid-State Circuits, pp.1555-1567, 1992

[13] Hidenori Sato, Hiroaki Matsuda, and Akira Onozawa, "A Balanced Mesh Clock Routing Technique for Performance Improvement," IEICE Transaction Fundamentals, Vol. E80-A, No. 8, August 1997.

[14] Kaushik Ravindran, Andreas Kuehlmann, and Ellen Sentivich, "Multi-Domain Clock Skew Scheduling," International Conference on Computer Aided Design, pp. 801-808, Nov 2003.

[15] M. B. I. Reaz, Nowshad Amin, M. I. Ibrahimy, F. Mohd-Yashin, and A. Mohammad, "Zero Skew Clock Routing For Fast Clock Tree Generation," Canadian Conference on Electrical and Computer Engineering, pp. 23-28, May 2008.

[16] Chia-Ming Chang, Shih-Hsu Huang, Yuan-Kai Ho, Jia-Zong Lin, Hsin-Po Wang, and Yu-Sheng Lu, "Type-Matching Clock Tree for Zero Skew Clock Gating," 45th ACM/IEEE Design Automation Conference, pp. 714-719, Jun 2008.

**Xianzhe Jin** was born in Yanji, China in 1982. He received a B.S. degree in Information and Communication Engineering from Paichai University, Taejeon, Korea in 2007 and will receive M.S. degree in Information and Communication Engineering from Hanbat National University in 2009. His current interests include clock routing synthesis, image processing codec design and SoC platform design and verification.

**Kwangki Ryoo** is an Associate Professor of the Department of Information and Communication Engineering at Hanbat National University. He was born in Konju, South Korea in 1964. He studied at Hanyang University, Seoul, Korea where he received his B.S., M.S. and PhD in Electronic Engineering in 1986, 1988, and 2000, respectively. From 1991 to 1994, he was the full-time lecturer in the department of Electronic Engineering at Korea Military Academy (KMA), Seoul, Korea. From 2000 to 2002, he worked at System IC Design Team at Electronics and Telecommunications Research Institute (ETRI), Taejeon, Korea. His current research interests include clock synthesis, system chip design, SoC platform design and verification, hardware/software co-design/verification, and multimedia codec design.