A Novel Timing-Driven Global Routing Algorithm Considering Coupling Effects for High Performance Circuit Design[*]

Jingyu Xu, Xianlong Hong, Tong Jing, Yici Cai

Dept. of Computer Science & Technology, Tsinghua Univ.,
Beijing, 100084, P. R. China

Jun Gu

Dept. of Computer Science, Hong Kong Univ. of S & T,
Hong Kong, P. R. China

Abstract-- As the CMOS technology enters the very deep submicron era, inter-wire coupling capacitance becomes the dominant part of load capacitance. The coupling effects have brought new challenges to routing algorithms on both delay estimation and optimization. In this paper, we propose a timing-driven global routing algorithm with consideration of coupling effects. The two-phase algorithm based on timing-relax method includes a heuristic Steiner tree algorithm and an optimization algorithm. Experimental results are given to demonstrate the efficiency and accuracy of the algorithm.

I. INTRODUCTION

The exponential reduction in scaling of feature size continues to push forward very-deep-submicron(VDSM) technology. The shrinking of geometries brings two new major concerns for chip performance. One is the power and ground noise caused by simultaneous switching circuits. The other is the increasing aspect ratio of wires and the decreasing of interconnect spacing, which have made the inter-wire coupling capacitance the dominant part of load capacitance. For high performance circuit design, ignoring such coupling capacitance may lead to significant deviations between actual and nominal timing responses, power consumptions and functional behaviors.

Previous works[1-4] have various contributions to timing optimization for standard cell(SC) global routing based on conventional interconnect delay metrics[5,6]. These algorithms are straightforward and efficient regarding specific test cases, while with recent advances in VDSM technology, the capability of timing optimization has limitations: the adopted delay models fail to take the signal input slew into account, and the optimization process inaccurately ignores the coupling effects.

Increasing concern has been raised regarding the coupling effects issues. In [7], global routing with crosstalk constraints has been studied. [8,9] introduced channel routing algorithms to minimize crosstalk. Global interconnect sizing and spacing with consideration of coupling capacitance are addressed in [10]. These previous approaches mainly fall into two categories. In the first category, the objective is minimizing crosstalk effects. Timing constraints are not emphasized. In the second, coupling capacitance is estimated for optimal wire sizing and spacing while no topological optimization are carried out. Among these various approaches, there is still an absence of quantificational and efficient measurement of the effect of coupling on the interconnect delay to guide the routing process. For high performance circuit global routing, efficient and accurate delay optimization methods are needed

to meet the maximum timing constraint [11].

In this paper, we tackle the timing optimization problem by applying a new timing-relax method that can be divided into two phases—an initial routing tree construction algorithm and an optimization algorithm. Advanced delay estimation methods are adopted to include the lateral coupling capacitance between neighboring wires for delay calculation. Besides, coupling effects are also utilized as a heuristic throughout the timing optimization.

II. PROBLEM FORMULATION

The definition of global routing graph (GRG) can be found in [16]. Without loss of generality, we tackle the timing optimization problem by minimizing the longest path delay from any input port to any output port of a circuit.

$$\text{Let } u_{ij} = \begin{cases} 1, \text{when net } n_i \text{ passes edge } e_j \\ 0, \text{ elsewise} \end{cases}$$

The timing-driven global routing problem is formulated as:

Minimize $\max_i delay(P_i) \quad i \in \{1, 2, ..., m\}$

Subject to $f_j = \sum_{i=1}^{N_n} u_{ij} \le c_j$

where m is the number of paths and N_n the total number of nets. $delay(P_i)$ denotes the delay of path P_i composed of gates and net wires. Let f_j be the total demand of the nets using edge e_j. f_j should be no greater than the edge capacity c_j.

III. TIMING ANALYSIS

A. Wire-Load-Estimation Model

For wire-load estimation, we apply an accurate wire-load model that is useful for timing and noise analysis in an early design stage[12]. It is derived by simulation of metal wires in different layers to obtain the first cut parasitic numbers and then by curve fitting them to the model. The largest error in estimation of parasitics of a metal wire using this model is within 5% of the field solvers to measure these parasitics.

With specified GRG capacity and number of used tracks, an estimated wire spacing and other geometrical information are the input of the wire load model. All capacitance around a given conductor can be extracted including the coupling capacitance. Considering the switching behavior of wires, we specify miller factor to decouple coupling capacitance according to switching activities.

B. Interconnect Delay Model

The conventional interconnect delay metrics[5,6] can not model parasitic effects and resistive shielding effects accurately. Further more, these metrics can not take signal input rise/fall time into account or propagate signal transition time during delay evaluation. We apply more advanced delay estimation methods. The built-in delay evaluation engine uses
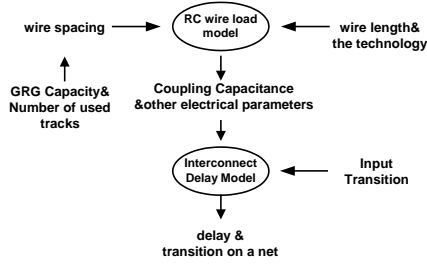
Fig.1. Delay calculation considering coupling effects

congruence transformation technique [13] to reduce the order of a large RC net-list while stability and passivity can be guaranteed. To be an efficient timing analysis engine with good trade-off between accuracy and speed, an adaptive method is used to control the order of net-list reduction. The resulting reduced order macro-model can be used for incremental delay estimation. In benchmark testing, the accuracy can be within 1% of SPICE simulation. Fig.1 shows the delay calculation method of the algorithm utilizing above models. The resulting delay value takes the coupling effects into consideration.

We use table-lookup model for gate delay estimation. [14] makes a useful attempt on applying table-lookup in performance-driven routing by constructing lookup table from SPICE data. In this paper, the lookup tables are all from industrial circuit library.

IV. GLOBAL ROUTING ALGORITHM

In our timing-relax method, we first try to construct optimal timing performance routing trees as the initial solution. Then, an optimization algorithm is proposed to relax non-critical regions to satisfy routing demands and utilizes coupling effect as a heuristic to optimize the circuit delay.

A. The Initial Timing-Driven Steiner Tree Algorithm

The initial solution is generated with two main concerns. (1)Timing performance of the initial solution is quite important. (2)A time consuming initial solution is not desirable. With above considerations, a heuristic algorithm is proposed based on the idea of CFD algorithm[15]. In Elmore delay model, the upper-bound of delay from source s to sink t is given by the following expression:

$$T(s,t) \leq T_D(s,t) = R(s,t)\sum_{k\in T}C_k = (R_s + rL(s,t))(C_z + cW) \quad (1)$$

where $L(s, t)$ is the wire length from s to t, and W is the total wire length of tree T. Since $T_D(s, t)$ is a function of L and W, a heuristic is proposed to minimize L and W simultaneously.

Definition 1 ITDT(Initial Timing-Driven Steiner Tree) algorithm constructs a Steiner tree T for a given set of pins on global routing graph to minimize $T_D(s, t)$.

Definition 2 Active Node is the current node for generating new edges. The initial set of active nodes is $v_i \in V$.

Let $p(v_i, v_j)$ be the distance between pin v_i and v_j. For a given pin r, a Compact Weight (CW) $d(r, v_j)$ is calculated to estimate the relative position of r with other active nodes, which is formulated as:

$$d(r,v_j) = \sum_{r \neq v_j, v_j \in R} p(r,v_j)^{-1} \quad (2)$$

There are 4 possible directions in the Manhattan space for each active node to generate an edge during Steiner tree construction. For each direction, CW is calculated to determine which direction yields the minimal W.

A Source Related Weight(SRW) $w(r,s)$ is used to denote the weight of generating directions related with $L(s,t)$. For a given pin r, $w(r,s)$ is defined as:

$$w(r,s) = \frac{m}{L(s,r)} \quad (3)$$

where m is the size of the set of active nodes, and $L(s,r)$ is the distance between source s and sink r. SRW encourages the node to grow towards the source.

The combined weight $cbw(r)$, is defined to contribute to the minimization of W and $L(s,t)$ simultaneously. The direction with a larger value of $cbw(r)$ attracts the edge generating of the active node.

$$cbw(r) = a_1d + a_2w = a_1 \sum_{r \neq v_j, v_j \in R} p(r,v_j)^{-1} + a_2 \frac{m}{L(s,r)} \quad (4)$$

Let R denote the set of active nodes. The heuristic timing driven tree construction algorithm is described in Fig.2.

ITDT_algorithm( )
$T \rightarrow R$;
While $|R| \neq 1$ *do*
    For(each r in R) do
        Calculate cbw(r) of each generating direction;
        Generate an edge according to cbw(r) and update R;
        If(two sub trees touch), merge sub trees;
        Select new active node and update R;
If(cycle found), break cycle;

Fig. 2. Heuristic Timing-driven routing tree algorithm

Considering the longest path delay of the circuit, constructing minimum delay tree for each net respectively may not necessarily yield the best timing for the circuit. We apply a short-term iteration algorithm to approach the best solution. The pseudo-code is shown in Fig.3.

Initial_solution_optimization_algorithm( )
1) For(each net) do
    Let the sink pin t_i yielding maximum L(s,t) be the target critical node t.
    *ITDT_algorithm(t);/*find best timing tree with respect to delay(s, t)*/*
2) Check each path and find the longest delay path P_t.
*3) dp = pathdelay(P_t), Improve = 0;/*the temporary longest path delay*/*
4) For(each net_i on P_t) do
    If(critical node $t_i \notin P_t$) and (net ID not marked) do
        Get the actual end pin t_a, $t_a \in V(net_i)$ and $t_a \in P_t$
        For $\forall v \notin path(s,t_a)$, find the biggest sub-tree rooted at v;
        *Minimize the total capacitance of Steiner tree below v;/*use*
ITDT_algorithm( ), and set a_1=1, a_2=0/*
        Update longest delay path and calculate new dp;
        If(Δdp≥0), set: Improve =0, restore the initial tree of net_i;
        Else set: Improve =1, mark the net ID;
5) If(Improve == 0), stop; else, go step2), iterate;

Fig. 3. Optimization algorithm for the initial routing solution

When minimizing the total capacitance of Steiner tree below a node v to reduce $delay(s, t_a)$ in *step 4*, we either increase the wire spacing to reduce the inter-wire coupling capacitance or construct minimum wire length sub-tree to reduce the ground capacitance. The proof for effectiveness of the optimization algorithm in *step 4* can be found in [16].

B. Timing Optimization
(1). The Timing Optimization Strategy

Based on the initial solution, we optimize the network topology of the circuit to adjust most congested areas. To keep good timing performance of the initial solution, the most critical paths should be kept in the original topology. Fig.4 is a simple example circuit. The critical path covers 12 nets, with thick solid line indicating the wire segments on the

Fig.4. Timing relax.

path. We first build up a "forbidden net list" from the critical path for rerouting. A ratio t is calculated to determine which net should be inserted into the "forbidden net list". For a given net_i on the critical path, t_i is the proportion of delay contributed by net_i to total path delay. When threshold t is set to zero, all the nets on critical path will be inserted into the forbidden net list. Otherwise, only those have great impact on total path delay are selected. They are called *forbidden nets* since they are seldom detoured in rerouting.

If resource conflict occurs among forbidden nets and un-forbidden nets, for example, in Fig.4, there are wire segments congested between node a and node b, we detour un-forbidden nets to let them give up these wire segments. Note that forbidden nets and un-forbidden nets are transferable. As shown in Fig.4, when the delay of node a to b is the biggest among any path before node c, *net0* has higher priority on the resource. But if we detour *net1* too much that the delay of node e to g is greater than that of a to b, the critical path changes and *net1* is inserted to "forbidden net list" while *net0* is pulled out. We use two methods to deal with this case. First, the delay redundancy of the un-forbidden nets is calculated to guide the rerouting. Second, timing information should be updated in time for the sake of finding new un-forbidden nets. In Fig.4, the nets in dash line are example of un-forbidden nets.

(2). Timing Optimization Considering Coupling Effects

The optimization algorithm is based on coupling effects transference, which directs the coupling effects to transfer to areas that are not critical for circuit delay. This method is applied simultaneously with the congestion optimization.

Definition 3 The extended congestion(EC) of a segment is the combination of coupling and congestion evaluation.

To get a more realistic estimation of load capacitance to meet max timing constraint, we define the coupling weight with the worst case consideration(i.e. wire above and below switch in opposite directions). The extended congestion weight of segment i is given by the following formulae:

$$\widetilde{w}_i = \alpha_1 w_{congi} + \alpha_2 w_{coupi} \tag{5}$$

$$w_{congi} = (f_i + 0.5\varepsilon)/(c_i + \varepsilon) \tag{6}$$

Fig.5. Detour critical path.

where w_{congi} is contributed merely by congestion of the segment and w_{coupi} by coupling effects. ε is a small constant.

As the coupling capacitance is inversely proportional to the spacing between neighboring wires, the coupling weight can be defined as:

$$w_{coupi} = \frac{C_{ci}}{C_{mi}} \tag{7}$$

where coupling capacitance C_{ci} depends on spacing and wire length of the segment, C_{mi} is the maximum coupling capacitance calculated under minimum spacing condition.

For a given GRG segment with fixed routing capacity, reducing the number of used tracks, wire spacing becomes larger and coupling capacitance gets smaller. Consequently delay is reduced. With the intention to optimize the circuit delay, we naturally note that adjusting the route on the critical path would result in a smaller circuit delay. The EC weight of segment i on the longest delay path is given by the following:

$$\widetilde{w}_i = \alpha_1 w_{congi} + \mu\alpha_2 w_{coupi}, \quad \mu > 1 \tag{8}$$

Since $\mu\alpha_2 > \alpha_2$, the extended congestion weight of segment on critical path is magnified. Thus when a net has alternative routes available in rerouting, it is more likely to choose the GRG segment on non-critical path because it has comparatively lower "congestion". Take Fig.5 as an example. The critical path(in dash line) is from PI to PO, covering 16 segments denoted by $s1$ to $s16$. A net with two sink pins t1, t2 is finding the route from source s to t1. It has several alternatives including via segment $s8$, $s9$ and $s10$. According to the algorithm, segment $s8$, $s9$ and $s10$ tend to have higher weight for being on the critical path. As shown in Fig.5, s finally chooses to connect t1 via segment $s17$, $s18$ and $s19$ after comparison on the congestion cost of each route. The reduced routing demand on critical path has transferred to non-critical paths, which may increase their path delay instead. However, the coupling effects transference is directed to facilitate the total delay, and consequently guarantees the algorithm space of further delay optimization.

The core algorithm for optimization is a 3-step (step3, 4, 5 in Fig.6) iteration algorithm using rip-up and reroute method.

Optimization_algorithm( )
1. *Construct initial timing tree for all the nets.*
2. *Evaluate global congestion of GRG.*
3. *Determine the longest delay path. (omitted for lack of space)*
4. *Simultaneous random optimization.*
    a) for each segment S_j of net_i: calculate congestion & coupling weight,
    b) trace back on longest delay path P_t, for each S_j of net_i on P_t: recalculate weighted coupling of S_j,
    c) select congested nets, reroute congested segment.
    d) after rerouting of all selected nets, refresh the weight of segments.
    e) evaluate overall routablity. if $E_{cg}(1.0) = \phi$, return.
    f) else, do step 3, starting with the best solution, go to 4(a), iterate.
5. *Sequential random optimization. Evaluate overall routablity.*
If the constraints are not satisfied after iterations of step 5, do the outmost iteration of step3-4-5 again.

Fig.6. Pseudo-code of the optimization algorithm.

By simultaneous random optimization of step4, we can approach some local optimum quickly but roughly. Then, starting with this solution, iteration of step5 is to find better route by accurate rerouting.

V. EXPERIMENTAL RESULTS

We have implemented the timing-driven global routing

algorithm on a Sun Enterprise 450 in C language and tested it with three industrial circuits extracted from microprocessor design. They are under $0.13\mu m$ technology and provided with corresponding look-up tables containing the gate timing arc information. Note that the routing resource constraints had all been satisfied in the following experimental results unless specified. Two sets of experimental data are provided. The first set of data (given in subsection A) is the comparison on timing optimization capability of three methods. The resulting delay values are all actual delay considering worst-case coupling effects. The second (given in subsection B) is the comparison of nominal delay taking no coupling effects into account with the actual delay in subsection A.

A. Comparison on Circuit Delay

Table I and Table II compare the experimental results of three different global routing methods. In method T(T: timing optimization), we skip the initial optimal routing tree construction phase and employ normal optimization algorithm merely. Star-topology routing tree is used as the initial tree, which is experimentally proved to be beneficial in reducing delay. In method IT(I: Initial optimal routing tree construction), we first apply our initial routing tree construction algorithm. Then, optimization is applied but no coupling directed optimization is carried out. Method ITC(C: coupling as a heuristic guiding optimization) is our two-phase timing driven algorithm considering coupling effects.

Table I gives the comparison of method T and IT. We can see from Table I that our initial routing tree construction algorithm has enabled the results of method IT a remarkably improved timing performance with significant speed-ups of the routing process. Table II compares the delay performance and run time of method IT and ITC. Applying our coupling directed delay optimization algorithm, method ITC has made an up to 12% improvement over the timing performance of method IT within very slightly increased run time.

TABLE I. COMPARISON ON TOTAL DELAY AND RUN TIME

Testcase	Number of Nets	Method T Delay(ns)	Method IT Delay(ns)	% Delay Optimized	Method T CPU Time(s)	Method IT CPU Time(s)
GDC	1568	12.2438	5.4832	55.22%	146	41
BIU	943	14.8035	8.0667	45.51%	65	14
PAT	4674	27.5655	19.6060	28.87%	155	45

TABLE II. COMPARISON ON TOTAL DELAY AND RUN TIME

Testcase	Number of Nets	Method IT Delay(ns)	Method ITC Delay(ns)	% Delay Optimized	Method IT CPU Time(s)	Method ITC CPU Time(s)
GDC	1568	5.4832	4.8501	11.55%	41	49
BIU	943	8.0667	7.4334	7.85%	14	16
PAT	4674	19.6060	18.0553	7.91%	45	51

B. Comparison on Nominal Delay and Actual Delay

To see how coupling effects affect the accuracy of timing optimization, we applied the three methods above again without considering coupling effects in delay estimation and optimization. Table III gives the comparison on the nominal delay and actual delay value. The column with "-" indicates the delay obtained with no consideration of coupling. "+" indicates the actual delay value obtained in subsection A. We

can see that in worst-case, the delay introduced by coupling can be up to 44% of the actual delay, which implies that simply not considering coupling effects may lead to great deviation between the nominal and actual delay value and consequently any optimization based on it will be inaccurate.

VI. CONCLUSIONS

We have presented a new timing-driven global routing algorithm for high performance circuit design. By taking coupling effects into account and utilizing it as a heuristic in guiding the optimization process, a remarkable improvement on delay performance has been achieved. Experimental results also show that, having good trade-off between accuracy and speed, the presented algorithm is very efficient for high performance circuit design.

REFERENCES

[1] J. Huang, X. L. Hong, C. K. Cheng, E. S. Kuh, "An Efficient Timing-Driven Global Routing Algorithm", *In: Proceedings of ACM/IEEE DAC*, Dallas, Texas, pp.596-600, 1993.
[2] X. L. Hong, T. X. Xue, J. Huang, C. K. Cheng, E. S. kuh, "TIGER: An Efficient Timing-Driven Global Router for Gate Array and Standard Cell Layout Design", *IEEE Trans. on CAD*, 16(11), pp.1323-1330, 1997.
[3] J. Hu, S. S. Sapatnekar, "A Timing-constrained Algorithm for Simultaneous Global Routing of Multiple Nets", *In: Proceedings of IEEE/ACM ICCAD*, San Jose, CA, pp.99-103, 2000.
[4] T. Jing, X. L. Hong, H. Y. Bao, Y. C. Cai, J. Y. Xu *et al*, "A Novel and Efficient Timing-Driven Global Router for Standard Cell Layout Design Based on Critical Network Concept", *In: Proceedings of IEEE ISCAS*, Scottsdale, Arizona, USA, pp.I165-I168, 2002.
[5] W. C. Elmore, "The Transient Response of Lumped Linear Networks with Particular Regard to Wideband Amplifiers", *J. of Applied Physics*, 19(1): pp.55-59, 1948.
[6] T. Sakurai, "Approximation of Wiring Delay in MOSFET LSI", *IEEE J. of SSC*, 18(4): pp.418-426, 1983.
[7] H.Zhou, and D.F.wang, "Global Routing with Crosstalk Constraints", *IEEE Trans. on CAD*, Vol. 18, No. 11, pp. 1683-1688, 1999.
[8] S.S.Sapatnekar, "A Timing Model Incorporating the Effect of Crosstalk on Delay and its Application to Optimal Channel Routing", *IEEE Trans. on CAD*, Vol. 19, No. 5, pp.550-559, 2000.
[9] K.C.Hsu, Y.C.Lin, P.X.Chiu, and T.M.Hsieh, "Minimum Crosstalk Channel Routing with Dogleg", *In: Proceedings of IEEE ISCAS*, Geneva, pp.73-76, 2000.
[10] J. Cong, L. He, C. K. Koh, Z. G. Pan, "Global interconnect sizing and spacing with consideration of coupling capacitance", *In: Proceedings of IEEE/ACM ICCAD*, San Jose, CA, USA, pp.628-633, 1997.
[11] T. Jing, X. L. Hong, Y. C. Cai, H. Y. Bao, J. Y. Xu, "The Key Technologies and Related Research Work of Performance-Driven Global Routing", *J. of Software*, 12(5): pp.677-688, 2001.
[12] X. D. Yang, A Reduced Order Modeling and Analysis for VLSI RLC Interconnect, Ph.D. thesis, CSE Dept., UCSD, USA, May, 2000.
[13] A. Odabasioglu, M.Celik, L.T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm", *In: Proceedings of ACM/IEEE ICCAD*, San Jose, CA, USA, pp.58-65, 1997.
[14] J. Lillis and P. Buch, "Table-Lookup Methods for Improved Performance-Driven Routing", *In: Proceedings of ACM/IEEE DAC*, San Francisco, USA, pp.368-373, 1998.
[15] X. L. Hong, "A Performance-Driven Steiner Tree Algorithm Using Constructed Force Directed Approach for Global Routing", *Chinese Journal of Semiconductors*, 16(3): pp. 218-223, 1995
[16] J. Y. Xu, X. L. Hong, T. Jing, Y. C. Cai, J. Gu, "An Efficient Hierarchical Timing-Driven Steiner Tree Algorithm for Global Routing", *In IEEE/ACM ASP-DAC*, Bangalore, India, pp.473-478, 2002.

TABLE III. COMPARISON ON NOMINAL DELAY AND ACTUAL DELAY

Test case	Method T Delay(ns)		% Delay deviation	Method IT Delay(ns)		% Delay deviation	Method ITC Delay(ns)		% Delay deviation
	-	+		-	+		-	+	
GDC	7.7279	12.2438	36.87%	4.0176	5.4832	26.73%	3.9209	4.8501	17.16%
BIU	9.7786	14.8035	33.94%	6.4929	8.0667	19.51%	6.3920	7.4334	14.01%
PAT	15.3715	27.5655	44.24%	14.7341	19.6060	24.85%	14.6451	18.0553	18.89%