Path-Based Timing Optimization by Buffer Insertion with Accurate Delay Model*

Zhang Yiqian, Zhou Qiang, Hong Xianlong and Cai Yici

(*Depertment of Computer Science & Technology, Tsinghua University, Beijing* 100084, *China*)

Abstract: An algorithm of path-based timing optimization by buffer insertion is presented. The algorithm adopts a high order model to estimate interconnect delay and a nonlinear delay model based on look-up table for gate delay estimation. And heuristic method of buffer insertion is presented to reduce delay. The algorithm is tested by industral circuit case. Experimental results show that the algorithm can optimize the timing of circuit efficiently and the timing constraint is satisfied.

Key words: buffer insertion; timing optimization; interconnect planning; routing algorithm

EEACC: 2570 **CCACC**: 7410D

CLC number: TN47 **Document code**: A **Article ID**: 0253-4177(2004)05-0520-06

1 Introduction

With progress in VLSI sub-micron technology, interconnect delay has become a dominant factor in integrated circuits. Interconnect optimization has become a critical step in the super performance design of VLSI. Some interconnect optimization techniques, such as topology optimization, device sizing, buffer sizing, and wire sizing, have gained widespread acceptance in deep sub-micron design. In particular, buffer insertion techniques have been successful in reducing interconnect delay.

Buffer insertion has been an active area of study in recent years. Closed formed solutions have been proposed in Refs. [1~ 4] for inserting buffers on a 2-pin net. The authors of Ref. [5] inserted buffers on a tree by iteratively finding the best buffer location. Van Ginneken[6] proposed a dynamic programming algorithm which finds the optimal solution under the Elmore wire delay model and a linear gate delay model. Several variants of this algorithm have been proposed[1,7,8]. These algorithms above use both simplified gate and wire delay models. However, it is well known that both the Elmore delay model and the linear gate model are inaccurate. Furthermore, both models are not aware of the input waveform, which become increasingly important in today´s deep sub-micron design. Therefore, the optimal solution under these simple models may be inferior. Thus, the author of Ref. [9] extends Van Ginneken´s algorithm by using both accurate interconnect and gate delay models, and the signal waveform is considered in buffer insertion.

All these algorithms above are timing opti-

* Project supported by National High Technology Research and Development Program of China(No. 2002AA1Z1460), and National Natural Science Foundation of China(No. 60176016)

Zhang Yiqian female, was born in 1979, PhD candidate. Her research interests focus on interconnect routing algorithm.

Zhou Qiang male, was born in 1961, associate professor. His research interest is VLSI layout algorithm and system integration.

Hong Xianlong male, was born in 1940, professor. His research interests include layout algorithms and systems.

Received 27 July 2003, revised manuscript received 7 December 2003

mization aiming at a single net, start at the receivers and move towards the root of the net. And each receiver has a user-specified required arrival time. The required departure time of the root is optimized by buffer insertion.

But in real circuit design, a timing requirement is generally specified as a delay bound for paths between registers and inputs/outputs. For net-based timing optimization, a technique is used in advance to assign overall timing constraint on the path into the nets that form the path. It is clear that the required performance is guaranteed if the delay time of every net satisfies the constraint. However, it may be too strict to require all nets to be optimized within the given fixed delay bound since this may cause needless power and area cost.

In this paper, an algorithm of path-based timing optimization by buffer insertion with accurate gate and interconnect delay computation is proposed.

2 Path-based timing analysis

In this section, path-based timing analysis technique and an accurate delay model are used to calculate gate and interconnect delay.

2.1 Delay calculation

Our algorithm applies a high order delay model to estimate interconnect delay and uses model order reduction technique to speed up the delay calculation for RC interconnect[10]. Further more, the coupling effects are taken into consideration in the calculation of capacitance of nets and interconnect delay.

We use a nonlinear delay model based on look-up table for gate delay estimation. The propagation and transition tables are provided in the industrial circuit library. The propagation delay is computed by performing table look-up and interpolation within the propagation table that is a two-dimensional table indexed by total output capacitance and input transition time. The total output capacitance is the sum of pin and wire capacitance in the net connected with the output pin of the gate. The transition time is computed by performing simple linear interpolation within the transition table that is a one-dimensional table indexed by the total output capacitance.

The calculation for net delay and cell delay is shown in Fig. 1.

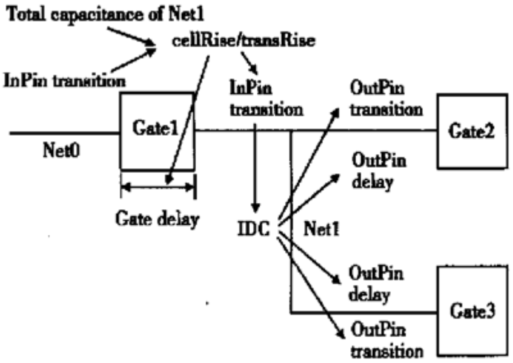

Fig. 1 Calculation of net and cell delay

The nonlinear delay model stores delay information in the technology library in the form of lookup tables. Cell delay can be computed by performing table lookup and interpolation in the propagation and transition tables provided in the library.

In the library, delay information is given in the following tables: Rise propagation, Fall propagation, Rise transition, and Fall transition. The fall propagation tables are two-dimensional tables indexed by total output capacitance and input transition time. The total output capacitance is the sum of pin and wire capacitance on the net connected with the output pin of the gate. The transition time is computed by evaluating the transition table. In this case, the table is a one-dimensional table based on total output capacitance. By performing simple linear interpolation within the table, the transition time is determined.

2.2 Method of path-based timing analysis

The procedure of the path-based timing analysis is illustrated as follows.

Figure 2 shows an example of logic network, in which PI is the input pin, PO is the output pin, A, B, and C are the gates of the circuit, $A_i(1)$ and $A_i(2)$ are the input pins of Gate A, $A_o(1)$ is the output pin of Gate A, and so forth. And s/t is added as the imaginary input/output pin of the circuit so that the logic circuit is transformed into a logic network.

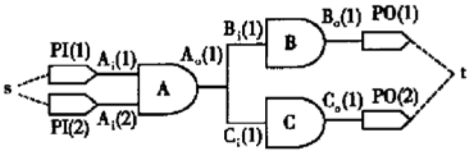

Fig. 2 A logic network

Figure 3 shows a timing graph, which is constructed according to the logic network, in which nodes in the graph represent pins in the logic network, and edges between nodes represent two types of connections: one shown by double-line arrowhead is interconnect relationships between a driver pin and its fanout, the other shown by single-line arrowhead is timing relationships between an input and an output pin of a gate. The weight of every edge is the interconnect delay or the gate delay calculated using the delay model described in section 2.1. As shown in Fig. 3, the weight d_{13} is the interconnect delay value from node 1 to node 3.

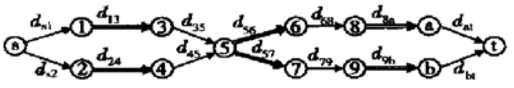

Fig. 3 A timing graph

Let $t_E(k)$ be the earliest arrival time when the signal propagates from node s to node k. It can be calculated by the following formula:

$$t_E(k)=\begin{cases}0, & k=\mathrm{s}\\ \max(t_E(i)+\mathrm{delay}(i,k)\,|\,(i,k)\in \boldsymbol{E}), & \text{others}\end{cases}$$

$t_E(\mathrm{t})$ is the total delay of the circuit, and it determines the speed of the circuit. It is denoted by T_E. T_E should not exceed a required delay value T_L. In order to satisfy the design requirement, the signal should arrive at node i before $t_L(i)$, here, $t_L(i)$ denotes the latest arrival time of node i. It can be calculated by the formula below:

$$t_L(i)=\begin{cases}T_L, & i=\mathrm{t}\\ \min(t_L(k)-\mathrm{delay}(i,k)\,|\,(i,k)\in \boldsymbol{E}), & \text{others}\end{cases}$$

Path-based timing analysis is performing so that the paths that violate the timing constraint T_L are found. Our algorithm first calculates the earliest arrival time of every node in the timing graph from PI to PO, then calculates the latest arrival time of every node from PO to PI. If the earliest arrival time of a node is later than the latest arrival time of it, the node is taken as a constraint-violated node, which belongs to a constraint-violated path. Thus, all the constraint-violated paths can be determined according to those constraint-violated nodes.

2.3 Strategy of optimization

For the paths that violate the timing constraint, some nets in the paths are chosen to be optimized by buffer insertion illustrated as follows.

Let all the paths that do not satisfy the timing constraint be $P_1, P_2, \cdots, P_i$, and let the nets in these paths be $N_1, N_2, \cdots, N_j$.

For each net N_t, $t\in\{1, 2, \cdots, j\}$, there are k constraint-violated paths including the net N_t: $P_{t_1}, P_{t_2}, \cdots, P_{t_k}$, $\{P_{t_1}, P_{t_2}, \cdots, P_{t_k}\}\subseteq\{P_1, P_2, \cdots, P_i\}$.

Then, $\{N'_1, N'_2, \cdots, N'_j\}$ is obtained by sorting $\{N_1, N_2, \cdots, N_j\}$ according to the value of k decreasingly.

Finally, the former m nets $N'_1, N'_2, \cdots, N'_m$ ($m\leqslant j$) are chosen from $\{N'_1, N'_2, \cdots, N'_j\}$ so that all the constraint-violated paths can be optimized.

From the procedure above, we can see that the more constraint-violated paths the net belongs to, the higher priority the net is given to insert buffer. Therefore, more "critical" nets are chosen to be inserted buffers so that the total number of buffers can be as small as possible during the timing optimization.

3 Heuristic method of buffer insertion

Since the interconnect delay is calculated by a high order model, buffer locations cannot be determined by simple analytical function. Therefore, heuristic method is adopted to determine buffer location in our algorithm.

(1) For the path from the source s to the critical sink t that belongs to the critical path, the midpoint of the path, the one fourth point of the path, and the third fourth point of the path, …, will be in turn chosen to insert buffers, that is, if there has been a buffer inserted in the last iteration in the midpoint of the path, the one fourth point will be chosen in this iteration, and so forth.

(2) For other branches in the net, if a branch is comparable to the path from s to t according to our measurement:

$l/l_0 \geqslant C_0$

where l is the length of the branch; l_0 is the length of the path from s to t; C_0 is the threshold value, C_0 is valued by 0.8 in our algorithm, then a buffer will be inserted to decouple this branch. Thus, the source gate can drive the critical sink and a smaller additional load due to buffered long noncritical branches, and the delay from s to t can be diminished.

As shown in Fig. 4, to optimize the delay time from A_1 to C_1, one buffer is inserted in the path from A_1 to C_1. To optimize the delay time from A_2 to C_2, one buffer is inserted in the path from A_2 to C_2, and another buffer is needed in the F_2 so that the long branch F_2B_2 can be decoupled. Similarly, three buffers are needed in order to optimize the delay time from A_3 to C_3.

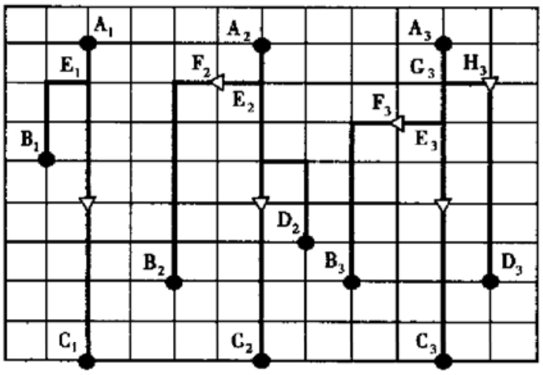

Fig. 4 Strategy of buffer insertion

4 Description of algorithm

As shown in Fig. 5, the path-based timing optimization algorithm contains three key steps. Our algorithm first performs path-based timing analysis for the whole circuit by the method shown in section 2.2 using the delay calculation method presented in section 2.1, and all the constraint-violated paths can be determined consequently. Then, some "critical" nets in the paths are chosen to be optimized using the optimization strategy shown in section 2.3. Finally, buffer insertion is adopted for the nets one by one using the buffer insertion method presented in section 3. The procedure iterates until all the paths satisfy the given timing constraint of the circuit.

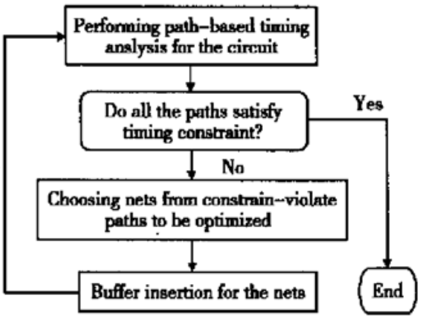
Performing path-based timing analysis for the circuit
Do all the paths satisfy timing constraint?
Yes
No
Choosing nets from constrain-violate paths to be optimized
Buffer insertion for the nets
End

Fig. 5 Algorithm flow

In the algorithm, the "critical" nets influencing the timing of the circuit are selected by the strategy of choosing nets from the constraint-violated paths, and the heuristic method of buffer insertion can efficiently reduce delay for each net. Thus. after several iterations. the timing of the whole circuit can be optimized efficiently, and the timing constraint will be satisfied. Because the buffer insertion method is simple, the computing complexity of the algorithm depends on the complexity of path-based analysis algorithm, as normal in polynormial.

5 Experimental results

Our algorithm is implemented by language C on SUN Enterprise E450. An industrial circuit, GDC which has 1728 paths is tested.

The experimental results are shown in Table 1. In the first column of Table 1, the different specified timing constraints are listed. The related numbers of paths that do not satisfy timing constraint are in column 2. The corresponding numbers of nets we processed and numbers of inserted buffers in timing optimization procedure are listed in columns 3 and 4. And in the last column, the related running times are given. For loose timing constraints in the first 3 rows of Table 1, the number of processed nets is equal to the number of inserted buffers. That means there is only one buffer inserted in each processed net. For tight timing constraints in the last 2 rows, the number of inserted buffers is greater than the number of processed nets. That means there are some nets inserted more than one buffer.

Experimental results show that our algorithm can optimize the delay of the circuit efficiently so that the specified timing constraint can be satisfied, and the running time of the algorithm is accepted.

Table 1 Timing optimization for GDC

Timing constraint /ns	Number of paths	Number of nets	Number of buffers	Running time/s
0. 85	279	43	43	63
0. 75	401	46	46	63
0. 68	589	113	113	85
0. 60	720	320	322	136
0. 56	764	357	396	395

6 Conclusion

In this paper, an algorithm of path-based timing optimization by buffer insertion with accurate gate and interconnect delay computation is proposed, and heuristic method of buffer insertion is adopted. Experimental results show that our algorithm can optimize the timing of circuit efficiently, and the running time of the algorithm is accepted. We will improve our algorithm in optimizing buffer location, buffer size, and number for each processed net in the future.

References

[1] Alpert C J, Devgan A. Wire segmenting for improved buffer insertion. IEEE/ACM DAC, 1997: 588

[2] Chu C C N, Wong D F. Closed form solution to simultaneous buffer insertion/sizing and wire sizing. International Symposium on Physical Design, 1997: 192

[3] Dhar S, Franklin M A. Optimum buffer circuits for driving long uniform lines. IEEE J Solid-State Circuits, 1991, 26(1): 32

[4] Wang Yibo, Cai Yici, Hong Xianlong. An interconnect optimization algorithm in SOC layout design. Chinese Journal of Simiconductors, 2003, 24(5): 550(in Chinese) [王一博, 蔡懿慈, 洪先龙. SOC 布图设计中的互连优化算法. 半导体学报, 2003, 24(5): 550]

[5] Lin S, Marek-Sadowska M. A fast and efficient algorithm for determining fanout trees in large networks. Proc Euro Conf Design Automation, 1991: 539

[6] Van Ginneken L P P P. Buffer placement in distributed RC-tree networks for minimal elmore delay. Int Symp Circuits and Systems, 1990: 865

[7] Lillis J, Cheng C K, Lin T T Y. Optimal wire sizing and buffer insertion for low power and a generalized delay model. IEEE J Solid-State Circuits, 1996, 31(3): 437

[8] Okamoto T, Cong J. Interconnect layout optimization by simultaneous steiner tree construction and buffer insertion. ACM/SIGDA Physical Design Workshop, 1996: 1

[9] Menezes N, Chen C P. Spec-based repeater insertion and wire sizing for on-chip interconnect. Proc of the 12th Intl Conf on VLSI Design, 1999: 476

[10] Odabasioglu A. Celik M. Pileggi L T. PRIMA: passive reduced-order interconnect macromodeling algorithm. In: Proceedings of ACM/IEEE ICCAD, 1997: 58

采用精确时延模型基于路径的缓冲器插入时延优化*

张轶谦　周　强　洪先龙　蔡懿慈

(清华大学计算机科学与技术系, 北京　100084)

摘要: 提出了一种基于路径的缓冲器插入时延优化算法, 算法采用高阶模型估计连线时延, 用基于查表的非线性时延模型估计门延迟. 在基于路径的时延分析基础上, 提出了缓冲器插入的时延优化启发式算法. 工业测试实例实验表明, 该算法能够有效地优化电路时延, 满足时延约束.

关键词: 缓冲器插入; 时延优化; 互连规划; 布线算法

EEACC: 2570　　**CCACC**: 7410D

中图分类号: TN47　　**文献标识码**: A　　**文章编号**: 0253-4177(2004)05-0520-06

* 国家高技术研究发展计划(批准号: 2002AA1Z1460)和国家自然科学基金(批准号: 60176016)项目资助

张轶谦　女, 1979 年出生, 博士研究生, 从事互连规划布线算法研究.

周　强　男, 1961 年出生, 副教授, 目前研究方向为 VLSI 布图理论与算法, 系统集成.

洪先龙　男, 1940 年出生, 教授, 博士生导师, 研究方向为 VLSI 布图理论与算法.

2003-07-27 收到, 2003-12-07 定稿