

## A New Clock-Driven ECO Placement Algorithm Based on Table-Lookup Delay Model\*

Liu Yi<sup>1</sup>, Hong Xianlong<sup>1</sup>, Cai Yici<sup>1</sup>, Wu Weimin<sup>1</sup> and Chung-Wen Albert Tsao<sup>2</sup>

(1 Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

(2 Celestry Design Technologies, Inc., San Jose CA 95112, USA)

**Abstract:** A new clock-driven ECO placement algorithm is presented for standard-cell layout design based on the table-lookup delay model. It considers useful clock skew information in the placement stage. It also modifies the positions of cells locally to make better preparation for the clock routing. Experimental results show that with little influence to other circuit performance, the algorithm can improve permissible skew range distribution evidently.

**Key words:** clock-driven; incremental placement; useful skew; permissible skew range

**EEACC:** 2570A      **CCACC:** 7410D

**CLC number:** TN47

**Document code:** A

**Article ID:** 0253-4177(2002)05-0473-06

### 1 Introduction

Due to the continuing reduction of feature size under deep-submicron technology, interconnect delay has become more significant than active device delay. High performance clock design is rapidly becoming a critical problem. Several techniques, such as clock scheduling<sup>[1]</sup> (solved by linear programming), clock routing algorithms (from zero-skew<sup>[2]</sup> to bounded-skew<sup>[3]</sup>), and simultaneously skew scheduling and clock tree routing<sup>[4,5]</sup>, have been developed.

Clock skew is manifested by a lead/lag relationship between the clock signals that control a local data path. A local path is composed of two sequentially adjacent registers and a combinational logic between them. The UST-BP algorithm in Reference[4] derives positive and negative skew bounds respectively. With allowable negative skew, the circuit can run at a clock period that is less than the critical logic path delay. This allows a

larger timing constraint to minimize the circuit area or power.

The problem of clock-skew optimization is typically addressed after placement. But if the skew information can be considered earlier, the better tree structure of clock routing may be obtained. Reference [6] has already begun to bring the clock-skew minimization problem into the placement domain, but it deals with zero skew only.

In this paper, we make the first attempt to use incremental placement, or ECO (engineering change order) placement<sup>[7,8]</sup> as a post-layout optimization method to improve the useful skews of clock. To deal with standard cell layout<sup>[9]</sup>, which is prevalent in the industry, we propose a new clock-driven placement algorithm. The advantage of our approach is that it introduces a new cell moving strategy and it can quickly adjust positions of the cells, enlarging legitimate positive and negative skew ranges. The experimental results show that it can ameliorate the distribution of permissible skew ranges significantly, with little damage to other de-

\* Project supported by National Natural Science Foundation of China (No. 60076016) and 973 National Key Project (No. G1998030411)

Received 18 September 2001, revised manuscript received 28 November 2001

©2002 The Chinese Institute of Electronics

sign objectives.

## 2 Preliminaries

We use  $S = \{s_1, s_2, \dots, s_n\}$  to denote the set of clock pins in the circuit, with  $s_i$  being the clock pin of flip-flop  $FF_i$ . A pair of flip-flops is sequentially adjacent when only some combinational logic exists between them. Figure 1 shows two sequentially adjacent flip-flops, with  $FF_{01}$  feeding data to  $FF_{02}$ .

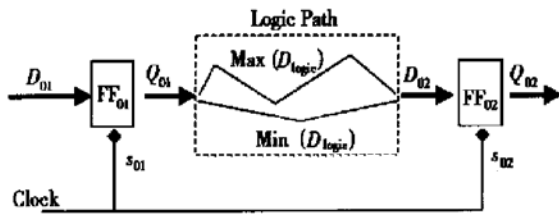


Fig. 1 A local data path in synchronous circuit

Clock signals may arrive at clock pins  $s_i$  and  $s_j$  at different time  $t_i$  and  $t_j$ , respectively. Clock skew between  $s_i$  and  $s_j$  is defined to be  $\text{skew}_{ij} = t_i - t_j$ . If clock signal arrives at  $s_{01}$  earlier, we call it negative skew; on the other hand, if  $s_{02}$  gets the clock signal first, the skew is defined as positive skew. With negative skews between certain clock sinks, e. g. sinks in critical logic paths, gate sizing can be allowed to reduce power consumption further<sup>[4]</sup>. That is the reason we call the negative skews useful skews.

To avoid double-clocking and zero-clocking in clock operation, skews can not be too excessive. If the maximum and minimum delay of each combinational logic block ( $D_{\text{logic}}$ ) are known, a region of valid clock skew called permissible skew range can be assigned to each local data path<sup>[10]</sup>. We must constrain each skew in its permissible range, by the following inequalities:

$$\begin{aligned} -\min(D_{\text{logic}}) - D_{\text{ff}} + D_{\text{hold}} &\leq \text{skew}_{1,2}(t_{01} - t_{02}) \\ &\leq C_p - \max(D_{\text{logic}}) - D_{\text{ff}} - D_{\text{setup}} \end{aligned} \quad (1)$$

where  $C_p$  is the time of clock period,  $D_{\text{ff}}$  is the propagation delay through the flip-flop,  $D_{\text{setup}}$  and  $D_{\text{hold}}$  are time for data to remain stable before and after the clock triggers respectively.

## 3 Table-lookup delay model

### 3.1 Delay table

In order to calculate the path delay more accurately, we adopt table-lookup delay model to do timing analysis. The nonlinear delay model stores vendor-specific delay information in the technology library in the form of table-lookup. We use two parameters to describe a signal: transition time (denoted as  $\delta$ ) and transfer delay (denoted as  $D$ ). As shown in Fig. 2, transition time is the time required for the signal to change its voltage state, and transfer delay is the timing difference between the signal and original input.

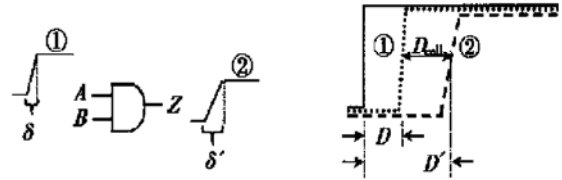


Fig. 2 Signal passing through a cell

There are altogether two delay tables for every cell: CDT (cell delay table) and TDT (transition delay table). They are both two-dimensional tables indexed by input  $\delta$  and output capacitance. While data signals pass a cell, CDT is used to calculate the transfer delay  $D_{\text{cell}}$  and TDT is for the change of transition time  $\delta'$ .

### 3.2 Cell delay calculation

The total path delay,  $D_{\text{total}}$  is the sum of every cell delay and net delay on the path:  $D_{\text{total}} = \sum D_{\text{cell}} + \sum D_{\text{connect}}$ . A required maximum delay,  $D_{\text{req}}$  is set to each controlled path, and the slack  $S$  is defined as  $S = D_{\text{req}} - D_{\text{total}}$ . A path with negative slack value is called a critical path.

Cell delay,  $D_{\text{cell}}$ , is the delay contributed by cell itself. It is roughly a nonlinear incremental function with parameters  $\delta$  and  $C$ . We use 4-point interpolation for calculating  $D_{\text{cell}}$  at certain combination of  $\delta$  and  $C$ . Suppose in CDT, the values used for transi-

tion time indexing are  $(\delta_0, \delta_1, \dots, \delta_j)$  and the values for output capacity indexing are  $(C_0, C_1, \dots, C_k)$ . Let  $D_{jk}$  be the value of delay when transition time is  $\delta_j$  and total output capacity is  $C_k$  in the table. We define  $D_{jk}$  as the following quadric function:

$$D_{jk} = a\delta_j C_k + b\delta_j + cC_k + d \quad (2)$$

For simplicity, assume we can find  $j$  and  $k$ , so that  $\delta_j < \delta_{j+1}$  and  $C_k < C_{k+1}$ . Parameters  $a, b, c$  and  $d$  can be determined from the four table values  $D_{jk}, D_{(j+1)k}, D_{j(k+1)}$  and  $D_{(j+1)(k+1)}$ . Then we can get the value of cell delay according to formula (2).

### 3.3 Connect delay calculation

The connect delay,  $D_{\text{connect}}$ , is the delay on the interconnect net. Since we have not known the pattern of the routing tree of nets in placement, we use an average load model of the nets and calculate connect delay as follows:

$$D_{\text{connect}} = \frac{R_{\text{wire}}}{N} \left( \frac{C_{\text{wire}}}{N} + C_{\text{pin}} \right) \quad (3)$$

where  $N$  is the number of fanin terminals of the net;  $R_{\text{wire}}$  and  $C_{\text{wire}}$  are the resistance and capacity of the wire, respectively;  $C_{\text{pin}}$  is the sum of capacity of all fanin pins on the net.

## 4 Algorithm description

### 4.1 Timing analysis

To perform clock-driven incremental placement, we must first find out the information of local path delay. Permissible skew ranges can be calculated through inequality (1), after we get values of delay of the longest and shortest path between adjacent flip-flop pairs. This task can be accomplished by the Bellman-Ford algorithm<sup>[11]</sup>.

A simple synchronous circuit can be modeled as a finite directed graph: timing analysis graph  $G(V, E)$ , which has  $|V|$  nodes and  $|E|$  edges. Each node in the graph,  $v_j \in V$ , is associated with a functional cell (such as flip-flop, circuit input, circuit output, or logic gate). Each edge in the graph,  $e_{ij} \in E$ , represents a physical connection

between vertices  $v_i$  and  $v_j$ .

We first sort nodes according to their topology order (when nodes are topologically sorted in timing analysis graph, there will be no edge going from a node  $v$  to another node  $u$  where  $v$  comes after  $u$  in the sorted order). By introducing data structure—tag, we record that which flip-flops are in front of a cell, and the most/least time it takes for the signal to reach that cell. Transferring tags dynamically in the topology sequence, we can obtain all the information of local path delay. Both topologically sorting nodes and transferring tags take  $O(|V| + |E|)$  iterations.

### 4.2 Force judgement

In order to improve clock permissible skew ranges, we must first make it clear how cell positions infect local path skew bound. Here, by borrowing the idea of the force directed interchange method, we treat every cell as if they were in some fictitious force field. The local longest path and shortest path between adjacent flip-flops will impose forces on combinational cells, which impel the cells to enlarge permissible skew range.

To increase positive skew range, from inequality (1), we know that the longest path will impose a force on cells intending to shrink the path; while the shortest path want to expand itself to enlarge negative skew range. After considering all the paths going through the cell, we will get its moving direction. What to do next is moving cells.

### 4.3 Cells adjustment

As shown in Fig. 3, suppose cell A wants to move in bottom right direction due to the

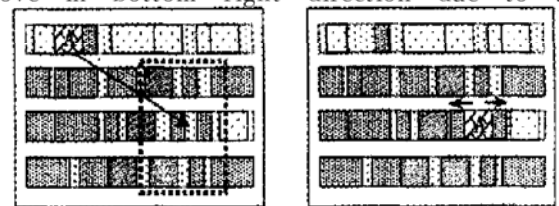


Fig. 3 Adjustment of cell (a) Decide select range; (b) Push other cells

forces. The algorithm first decides a select range from which the most suitable position will be chosen, then put cell A in its optimal location. Cell A will push other cells of that row to both sides, just like dissolving in that row. The original position of cell A becomes blank, which is ready for other cells to move in.

Being ECO placement, it wants to move as few cells as possible. In Fig. 3, suppose there are  $L$  cells on the left of A, denoted as  $C_{l1}, C_{l2}, \dots, C_{lL}$  from right to left. There is space  $G_{lp}$  between cell  $C_{l(p-1)}$  and  $C_{lp}$ ,  $ML_p$  is total space got when we push to the  $p$ th cell. Similarly, there are  $R$  cells  $C_{r1}, C_{r2}, \dots, C_{rR}$  on the right of A from left to right. We also denote  $G_{rp}$  and  $MR_p$ . The task is how to push the cells

until there is enough room to insert cell A. Hence, the problem can be expressed as the following integer program:

$$\begin{aligned} & \text{Minimize} && l + r \\ & \text{subject to:} && 0 \leq l \leq L, \\ & && 0 \leq r \leq R, \text{ both } l \text{ and } r \text{ are integer} \\ & && ML_l + MR_r \geq \text{Width}(A) \end{aligned} \quad (4)$$

We set up a space matrix  $M_{(L+1)(R+1)}$ , whose element  $(l, r)$  is:  $m_{lr} = ML_l + MR_r$ . If cell A can be put in such cell row, we can find a solution  $(p, q)$  in the diagonal of matrix  $M$  at first. This is an upper bound of the moving cells number:  $N_u = p + q$ . Next, change current solution  $(l, r)$  to next solution  $(l', r')$  through left and right branches iteratively according to the following steps:

Repeat	left branch
$l' = \begin{cases} l & m_{lr} \geq \text{Width}(A) \\ l+1 & m_{lr} < \text{Width}(A) \end{cases}$	
$r' = \begin{cases} r-1 & m_{lr} \geq \text{Width}(A) \\ r-1 & m_{lr} < \text{Width}(A) \text{ and } l' + r > N_u \\ r & m_{lr} < \text{Width}(A) \text{ and } l' + r \leq N_u \end{cases}$	
Until $(r' < 0 \text{ or } l' > R)$	

Repeat	right branch
$r' = \begin{cases} r & m_{lr} \geq \text{Width}(A) \\ r+1 & m_{lr} < \text{Width}(A) \end{cases}$	
$l' = \begin{cases} l-1 & m_{lr} \geq \text{Width}(A) \\ l-1 & m_{lr} < \text{Width}(A) \text{ and } r' + l > N_u \\ l & m_{lr} < \text{Width}(A) \text{ and } r' + l \leq N_u \end{cases}$	
Until $(l' < 0 \text{ or } r' > R)$	

Suppose there are  $n$  cells in that row, we take at most  $(n+1/4n)$  iterations to find the optimal solution with minimum number of moved cell. So the complexity of inserting one cell is  $O(n)$ .

## 5 Experimental result

The clock-driven ECO placement algorithm has been implemented in language C on a SUN Enterprise 450 workstation running on Solaris Operating System. We have run our program on two industrial test cases with 4000 to 40000 cells. Some statistical information is listed in Table 1.

Table 1 Characteristics of test cases

Circuits	Rows	Cells	Flip-flops	Nets	Local paths
Table	48	4596	127	4738	62
Design	164	44283	1947	44943	155748

Experimental results are summarized in

Table 2. There is evident improvement for the minimum and average permissible skew range, especially for the negative skew range. So that it provides a better foundation for the useful skew clock routing. Meanwhile it does little spoil to the timing characteristics. Certainly, cells' movement will make total wire length grow, but the increment is quite small.

In order to observe the changing trend of permissible skew ranges more clearly, we listed all the local paths between adjacent flip-flop pairs in test case——Design. As shown in Table 3, all the 155748 local paths were divided into 7 portions. They were sorted by the values of negative permissible skew range. The values were measured by the  $C_p$  (clock period) value. After optimization, some local paths got larger permissible skew ranges and belong to other portions. We collected statistics on the change number of local paths in each portion.

Table 2 Improvement of permissible skew range

Circuit		Table		Design	
Performance		Original	Optimized	Original	Optimized
Negative skew range/ns	Minimum	0.2088	0.2427 (+ 16.2%)	0.3539	0.3704 (+ 4.66%)
	Average	0.9020	0.9166 (+ 1.62%)	7.6409	7.7058 (+ 0.85%)
Total skew range/ns	Minimum	3.0000	3.0000	14.8679	14.8683
	Average	3.0000	3.0000	27.5519	27.5623
Timing	Critical paths	769	766	72	72
	Min slack/ns	-2.0463	-2.0433	-4.7030	-4.7660
Total wire length/ $10^6\mu\text{m}$		8.1846	8.2387 (+ 0.66%)	10.877	10.928 (+ 0.47%)
Run time/s		-	1.03	-	198.24

Table 3 Permissible skew range distribution in circuit Design

Negative skew range	Original	Optimized	Change $\Delta$
0~ $1/7C_p$	10203	8781	-1422
$1/7C_p \sim 2/7C_p$	47395	48292	897
$2/7C_p \sim 3/7C_p$	44253	44554	301
$3/7C_p \sim 4/7C_p$	35878	36045	167
$4/7C_p \sim 5/7C_p$	15820	15892	72
$5/7C_p \sim 6/7C_p$	2155	2140	-15
$6/7C_p \sim 7/7C_p$	44	44	0
Total	155748	155748	-

From the statistical data we can see that local paths with small skew ranges move downwards clearly after adjustment. That means not only the worst permissible skew range is enlarged, but also narrow skew ranges have been improved globally.

## 6 Concluding remarks

In this paper, we presented a new clock-driven incremental placement algorithm. It finds the key cells' through timing analysis. It uses the concept of force to decide cells moving direction, and it moves cells to enlarge permissible skew ranges between adjacent flip-flops. Results obtained from experiments have demonstrated that making better preparation for useful clock routing, the algorithm can improve skew range distributions effectively.

## References

- [1] Fishburn J P. Clock skew optimization. IEEE Trans Comput, 1990, 39(7): 945
- [2] Chao Tinghai, Hsu Yuchin, Ho Janming, et al. Zero skew clock routing with minimum wirelength. IEEE Trans Circuits Syst II -Analog and Digital Signal Processing, 1992, 39(11): 799
- [3] Huang D J-H, Kahng A B, Tsao C-W A. On the bounded-skew clock and Steiner routing problems. Proceeding Of 32<sup>nd</sup> Design Automation Conference, USA, 1995: 508
- [4] Xi J G, Dai W W-M. Useful-skew clock routing with gate sizing for low power design. Proceeding Of 33<sup>rd</sup> Design Automation Conference, 1996: 383
- [5] Tsao Chung-Wen Albert, Koh Cheng-Kok, et al. UST/DME: a clock tree router for general skew constraints. IEEE/ACM International Conference on Computer-Aided Design. 2000: 400
- [6] Natesan V, Bhatia D. Clock-skew constrained cell placement. Proceedings of the IEEE International Conference on VLSI Design, 1995: 146
- [7] Choy C-S, Cheung T-S, Wong K-K. Incremental layout placement modification algorithms. IEEE Trans Comput-Aided Des Integr Circuits Syst, 1996, 15(4): 437
- [8] Yu Hong, Yao Bo, Hong Xianlong, et al. ECOP: a row-partition based incremental placement algorithm for standard cell layout design. Chinese Journal of Semiconductors, 2001, 22(1): 96(in Chinese)[于泓, 姚波, 洪先龙, 等. ECOP: 一种基于单元行划分的标准单元模式增量布局算法. 半导体学报, 2001, 22(1): 96]
- [9] Hou Wenting, Yu Hong, Hong Xianlong, et al. A new congestion-driven placement algorithm based on cell inflation. Chinese Journal of Semiconductors, 2001, 22(3): 275
- [10] Kourtev I S, Friedman E G. Synthesis of clock tree topologies to implement nonzero clock skew schedule. IEEE Proceedings: Circuits, Devices and Systems. 1999, 146(6): 321
- [11] Gerez S H. Algorithms for VLSI design automation. Wiley, New York, 1999: 95

## 一种新的基于查找表的时钟性能驱动的增量式布局算法\*

刘 毅<sup>1</sup> 洪先龙<sup>1</sup> 蔡懿慈<sup>1</sup> 吴为民<sup>1</sup> Chung-Wen Albert Tsao<sup>2</sup>

(1 清华大学计算机科学与技术系, 北京 100084, 中国)

(2 Celestry Design Technologies 公司, San Jose CA 95112, 美国)

**摘要:** 提出了一种新的时钟性能驱动的增量式布局算法, 它针对目前工业界较为流行的标准单元布局, 应用查找表模型来计算延迟. 由于在布局阶段较早地考虑到时钟信息, 可以通过调整单元位置, 更有利于后续的有效偏差时钟布线和偏差优化问题. 来自于工业界的测试用例结果表明, 该算法可以有效地改善合理偏差范围的分布, 而对电路的其它性能影响很小.

**关键词:** 时钟驱动; 增量式布局; 有用偏差; 合理偏差范围

**EEACC:** 2570A      **CCACC:** 7410D

**中图分类号:** TN47      **文献标识码:** A      **文章编号:** 0253-4177(2002)05-0473-06

\* 国家自然科学基金(批准号: 60076016)及国家“九七三”重点研究发展规划(编号: G1998030411)资助项目

2001-09-18 收到, 2001-11-28 定稿

©2002 中国电子学会