Optimal Spacing and Capacitance Padding for General Clock Structures [*]

Yu-Min Lee Hing Yin Lai Charlie Chung-Ping Chen
Department of Electrical and Computer Engineering
University of Wisconsin at Madison
Madison, WI 53706
Tel: 1-608-265-3789 Tel: 1-608-265-1145
yu-min@cae.wisc.edu laih@cae.wisc.edu chen@engr.wisc.edu

ABSTRACT

Clock-tuning has been classified as important but tough tasks due to the non-convex nature caused by the skew requirements. As a result, all existing mathematical programming approaches are often trapped at local minimum and have no guarantee of obtaining global optimal solution.

In this paper, we present optimal clock tuning algorithms which effectively apply capacitance-padding to reduce clock skew, power, and delay for general clock topologies. Capacitance-padding can be achieved by wire-spacing, wire-splitting, wire-padding and transistor-padding. We show that under the Elmore delay model, capacitance-padding can be formulated as a linear programming problem and solved with great efficiency. Capacitance-padding can also be used as a post processing step for any non-zero-skew clock tree or mesh structure to achieve timing closure. Experiment results on several practical industry examples show that our algorithms are extremely efficient. Problems with over 6000 variables can be optimally tuned within 1 minute on a PC with 500-MHZ Intel Pentium III processor.

I. INTRODUCTION

Delay, skew, and power are the most important concerns in current VLSI clock-tree design. With the increasing complexity of synchronous ASICs, clock skew and clock-signal delay have become important factors in determining circuit performance [1, 3, 7, 15]. As shown in [7], the clock-signal delay has great impact on system-level skew and thus is an important consideration in clock-tree design. As reported in [5], power dissipation occupied 40% of the overall chip's power dissipation. Therefore, it is essential to carefully design clock to simultaneously consider delay, skew, and power.

Clock-tuning has been shown an effectively way to enhance clock skew, delay and power [2]. Due to the non-convex nature of this problem caused by the skew requirements which require signal arrival time within mindelay and maxdelay constraints, all existing mathematical programming approaches are often trapped into local minimum and can not guarantee to obtain global optimal solution.

Moreover, existing algorithms suffer long runtime and large storage requirements for large scale problems. For example, [10, 15] convert the skew minimization problem into the least-square minimization problem and the runtime and storage are proportional to cubic and quadratic in the problem size.

In this paper, we present optimal clock tuning algorithms by capacitance-padding. Capacitance-padding is achieved by wire-spacing, wire-splitting, wire-padding, and transistor-padding. Capacitance padding can effectively reduce clock skew and delay for general clock topologies including trees as well as mesh structures without changing clock topologies. We first show that this problem can be formulated as a linear programming problem and hence the optimal solution is guaranteed. We then propose a two-stage approach which minimizes the maximum delay at the first stage and apply capacitance padding to further minimize skew at the second stage. Our algorithm can also be used to explore skew-delay-power trade-off relationship. Experiment results on several practical industry examples show that our algorithms are extremely efficient. Problems with over 6000 variables can be optimally tuned within 1 minute on a PC with 500-MHZ Pentium III processor.

II. PRELIMINARIES

By the modified nodal analysis, the system equations of general linear circuits can be expressed as follows

$$C\dot{x} \quad = \quad -Gx + bu, \tag{1}$$

where x represents the state variables, G is the conductance matrix, C is the susceptance matrix (includes capacitors and inductors), and the term bu represents excitation from independent sources. Applying the Laplace transform to Equation (1) and assuming zero initial conditions (i.e. $x(0) = 0$), we get

$$sCX(s) \quad = \quad -GX + bU(s), \tag{2}$$

where $X(s)$ and $U(s)$ denote the Laplace transform of x and u, respectively. After rearranging the terms of the above equation, we get the impulse response of the system as follows

$$X(s) \quad = \quad (sC + G)^{-1}b.$$

Let $A = -G^{-1}C$, we can rewrite the above equation as

$$X(s) \quad = \quad (I - sA)^{-1}G^{-1}b. \tag{3}$$

---

[*] This work was partially supported by the Texas Advanced Research Program under Grant No. 003658459.

Let $v = G^{-1}b$, the AWE method expands the above equation at $s = 0$ or $s = \infty$ as

$$X(s) \quad = \quad \sum_{i=0}^{\infty} A^i v s^i.$$

By setting $m_i = A^i v$, the AWE method uses the following recurrent relation to iteratively compute higher order moments from lower order moments.

$$\begin{aligned} m_0 &= v \ = G^{-1}b \\ m_i &= Am_{i-1}. \end{aligned}$$

Note that it only needs to perform LU decomposition of G once. The rests of the computation are on repeatedly solving the above linear equations. To get more insight of the AWE method, we have the following observations. By the definition of A, we know

$$\begin{aligned} Gm_0 &= b \\ Gm_1 &= -Cm_0, \qquad (4) \\ Gm_i &= -Cm_{i-1}, \ \ i \geq 2. \qquad (5) \end{aligned}$$

Hence, to calculate m_q, we substitute each capacitor with a current source value Cm_{q-1} and then solve for voltages and currents of the original circuit while keeping resistors and conductors unchanged. It has been shown that (negated) m_1 is the Elmore delay [16] and m_0 is the DC solution.

There are several ways to adjust capacitance values such as spacing, padding, transistor-padding, and splitting. As shown in Figure 1, wire spacing increases or decreases the spaces between wires to reduce or increase the coupling capacitance between them. Wire-padding and transistor-padding attach wires and transistors to a wire in order to increase the capacitance load as shown in Figure 2. As shown in Figure 3, wire-splitting (split the wire into several narrower wires) changes the total capacitance value while preserving the original resistance. We call those capacitance adjusting techniques capacitance padding. The values of padding capacitance can be fine tuned continuously since wire sizes, space, and transistor width can be adjusted continuously. Note that capacitance padding will not change the clock topology or even wire width and resistance. We can simply use the empty routing space to perform spacing and capacitance padding. In this way, there is no penalty for the routing at all.

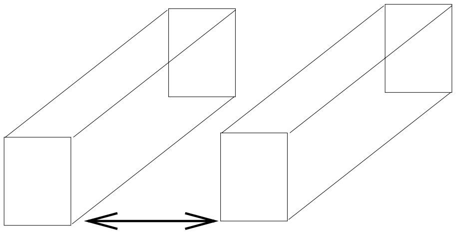

Fig. 1. Wire Spacing.

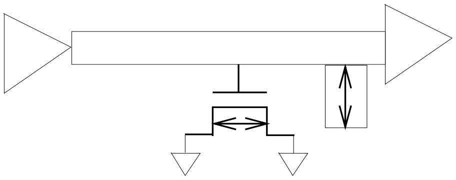

Fig. 2. Wire and Transistor Padding.

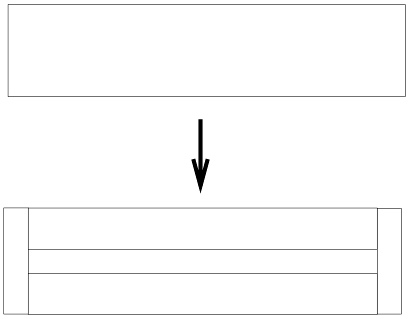

Fig. 3. Wire Spliting.

Capacitance padding is different from wire-snaking since wire-snaking changes both resistance and capacitance but capacitance padding changes capacitance only. As a result, the net delay changed by capacitance padding can be formulated as a linear programming problem while the delay changed by wire-snaking can not.

In this paper, we assume the upper bound and lower bound of the capacitances are given. Those values can be obtained by any capacitance extraction method.

There are two major components of power dissipation in the CMOS circuits, namely, static dissipation (due to leakage current) and dynamic dissipation (due to charging and discharging of load capacitances [capacitive dissipation], and switching transient current [short-circuit dissipation]). Given a clock tree, its power dissipation can be computed as follows [14]:

$$P \quad = \quad f(C_{tot}V_{dd}^2 + V_{dd}\int_0^{T_c} i_s(t)dt),$$

where C_{tot} is the total capacitance of the tree, f is the clock frequency, T_c is the cycle time, and i_s is the short-circuit current. We consider only the capacitive dissipation in this paper, since the capacitive dissipation usually dominates the power dissipation in practical applications [4]. Hence we have

$$P \quad \approx \quad fC_{tot}V_{dd}^2.$$

Clock skew is defined as the maximum difference in the delays from the clock source to clock sinks; that is, the skew of a clock tree, $S = \max_{i,j}|D_i - D_j|$.

In this paper, we are targeting to solve the optimal spacing and capacitance padding for clock tree to reduce skew, power, and delay. This problem can be formulated as follows:

- *CSCP: Clock Spacing and Capacitance Padding Problem* Given: A clock tree T with the source N_0 and sinks $\{N_1, N_2, \ldots, N_s\}$, wire segments $\{w_1, w_2, \ldots, w_n\}$, buffers $\{w_0, w_{n+1}, w_{n+2}, \ldots, w_{n+m}\}$, upper bounds $\{U_0, U_1, \ldots, U_{n+m}\}$, and lower bounds $\{L_0, L_1, \ldots, L_{n+m}\}$ for the capacitance $c_0, c_1, \ldots c_{n+m}$.
  Objective: Find $\mathbf{c}$ that minimizes $\max_{1 \le i \le s} D_i$, S, P, and A.

III. ALGORITHMS

In this section, we first present our algorithms for solving optimal spacing and capacitance padding for skew minimization problem and then demonstrate how to obtain skew, delay, and power trade-off relationships.

A. Optimal Spacing and Capacitance Padding

We formulate the optimal spacing and capacitance padding for clock tree to minimize skew problem as follows:

$$\begin{aligned} \mathcal{M} : Minimize \quad & D_{max} - D_{min} \\ Subject\ to \quad & D_{min} \le D_i(\mathbf{c}) \le D_{max},\ 1 \le i \le s, \\ & L_i \le c_i \le U_i,\ 0 \le i \le n+m. \end{aligned}$$

Note that D_{max} and D_{min} are variables we introduced to minimize clock skew. Problem M contains two sets of constraints. The first set of constraints ensure that every sink satisfies its skew (maxdelay and mindelay) constraints and the second set of constraints makes sure that the capacitance value of every device is within feasible region.

Since Elmore delay is the first (negated) moment of a node, the Elmore delay of any point can be obtained from Equation (4). From Equation (4), we know that by keeping resistance matrix G fixed, m_1 is only a linear function in terms of susceptance matrix C. If only C can be adjusted, this problem is a linear programming problem which uses C to adjust m_1. Unfortunately, we can not include Equation (5) into our problem formulation since Cm_1, the right hand side of Equation (5), are no longer linear terms.

We rewrite Problem $\mathcal{M}$ under the Elmore delay model as follows:

$$\begin{aligned} \mathcal{M}' : Minimize \quad & D_{max} - D_{min} \\ Subject\ to \quad & D_{min} \le -m_1^i \le D_{max},\ 1 \le i \le s, \\ & Gm_1 = -Cm_0, \\ & L_i \le c_i \le U_i,\ 0 \le i \le n+m. \end{aligned}$$

Noted that problem M' is a linear programming problem since G and m_0 are all constants, and the objective function is a linear function in terms of C.

We can also explore the skew, delay, and power trade-off relationships by assigning weight to each factor and iteratively solving the following problem to search the desired solutions.

$$\begin{aligned} \mathcal{M}'' : Minimize \quad & \alpha D_{max} + \beta(D_{max} - D_{min}) + \gamma \sum_{i=0}^{n+m} c_i \\ Subject\ to \quad & D_{min} \le -m_1^i \le D_{max},\ 1 \le i \le s, \\ & Gm_1 = -Cm_0, \\ & L_i \le c_i \le U_i,\ 0 \le i \le n+m. \end{aligned}$$

[2] shows that simultaneous wire-sizing and buffer-sizing can significantly reduce delay and skew. To further reduce the clock skew, we can apply spacing and capacitance padding as a post-processing step for the delay-optimized solution. The experimental results show that this two-stage approach can significantly reduce clock skew.

B. More Accurate Delay Model

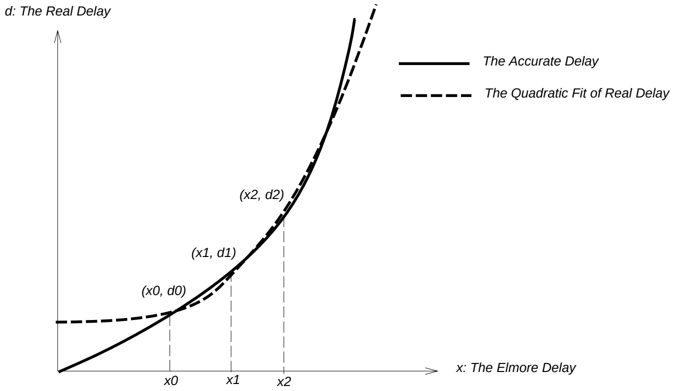

Fig. 4. The quadratic model of real delay in terms of Elmore delay.

To improve of the accuracy of Elmore delay model, it is possible to use higher order moments to get the more accurate delay and sensitivity. On the other hand, the computation effort for the exact sensitivity for each parameter is computational expensive [9]. Hence, we propose to use a fudge-factor approach to map Elmore delay to the exact delay while still using Elmore delay as a sensitivity information. In this way, we can iteratively update the Elmore delay targets for each sink to compensate the errors. In particular, we sequentially use a quadratic function $a_k x^2 + b_k x + c_k$ as our approximation function, where a_k, b_k, and c_k are the respective new coefficients for next iteration and new x is the Elmore delay target. These coefficients will be updated iteratively to fit the accurate delays in a quadratic form. Let d_0, d_1, and d_2 be the accurate delays respective to three Elmore delays x_0, x_1, and x_2 as shown in Figure 4, where a, b, and c are coefficients which satisfy the following equations.

$$\begin{aligned} d_0 &= a x_0^2 + b x_0 + c \\ d_1 &= a x_1^2 + b x_1 + c \\ d_2 &= a x_2^2 + b x_2 + c. \end{aligned}$$

We solve the above equations and get:

$$
\begin{aligned}
a &= \frac{\frac{d_1-d_0}{x_1-x_0} - \frac{d_2-d_0}{x_2-x_0}}{x_1 - x_2} \\
b &= \frac{d_1 - d_0}{x_1 - x_0} - a(x_1 + x_0) \\
c &= d_0 - a x_0^2 - b x_0.
\end{aligned}
$$

Given the target delay D and fitted coefficients a, b, and c, we can find the target Elmore delay x for the next iteration from the following equation

$$ax^2 + bx + c \quad = \quad D.$$

After solving the above equation, we obtain $x = \frac{-b \pm \sqrt{b^2-4a(c-D)}}{2a}$. After setting up the new target delay in terms of Elmore delay, we call the linear programming subroutine to find the optimal solution. The above process is repeated until the program converges.

IV. EXPERIMENTAL RESULTS

We implement and test our algorithm on the five circuits $r1$–$r5$ used in [13] on a PC with 500MHZ Pentium III microprocessor. The per micron resistance and capacitance used are $3m\Omega$ and $0.02fF$, respectively. The lower and upper bounds for wire widths are $1\mu m$ and $10\mu m$, respectively. We use both spacing and capacitance padding techniques to minimize skew. Table I lists the names of the circuits, numbers of wire segments in the circuits, delays, and skews requirements. The skews are verified by SPICE simulations. We first run our algorithms on the original clock tree to minimize skew or only reduce 50% skew. The experimental results show that our algorithm reduces 99.46% skew with only 16.49% delay penalty in average. On the other hand, if we only want to minimize 50% skew then the delay actually can be reduced about 17.904% in average. All the runtimes are under 1 minute. We also test two-stage approach which first runs optimal wire-sizing to minimize the maximum delay followed by spacing and capacitance padding to further reduce skew. In this way, both skews and delays are significantly reduced. Figure 5 shows the trade-off relationship between delay and skew.

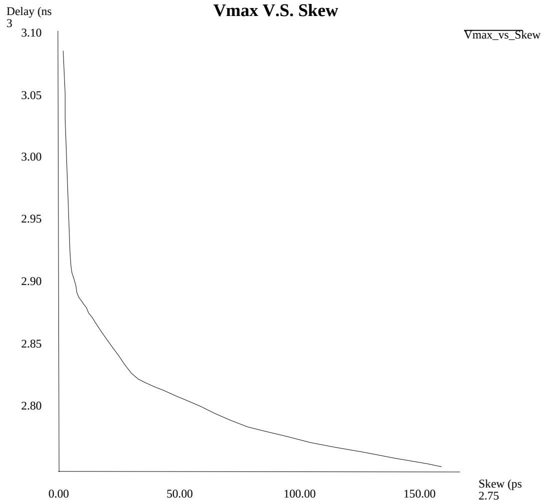

Fig. 5. Delay v.s. skew tradeoff.

REFERENCES

[1] H. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley Pub. Company Inc., 1990.

[2] C.P. Chen, Y.W Chang, and D. F. Wong, “Fast performance-driven optimization for buffered clock trees based on Lagrangian relaxation” *DAC*, 1996.

[3] J. Chung and C.-K. Cheng, “Skew sensitivity minimization of buffered clock tree,” *Proc. ICCAD*, pp. 280–283, 1994.

[4] J. Cong and K.-S. Leung, “Optimal wiresizing under elmore delay model,” *IEEE TCAD* 14(3), pp. 321–336, 1995.

[5] D. Dobberpuhl and R. Witek, “A 200MHz 64B dual-issue CMOS microprocessor,” *Proc. IEEE ISSCC*, pp. 106–107, 1992.

[6] W. C. Elmore, “The transient response of damped linear networks with particular regard to wide band amplifiers,” *J. Applied Physics*, 19(1), 1948.

[7] M. A. B. Jackson, et al., “Clock routing for high performance ICs,” *Proc. DAC*, pp. 573–579, June 1990.

[8] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley Pub. Company Inc., 1984.

[9] N. Menezes, R. Baldick, and L. T. Pillage, “A sequential quadratic programming approach to concurrent gate and wire-sizing,” *Proc. ICCAD*, 1995.

[10] N. Menezes, S. Pullela, F. Dartu, and L. T. Pillage, “RC interconnect syntheses—a moment fitting approach,” *Proc. ICCAD*, Nov. 1994.

[11] S. Pullela, N. Menezes, and L. T. Pillage, “Reliable non-zero skew clock trees using wire width optimization,” *Proc. 30th ACM/IEEE Design Automation Conference*, pp. 165–170, June 1993.

[12] S. S. Sapatnekar, “RC interconnect optimization under the Elmore delay model,” *Proc. ICCAD*, pp. 387–391, 1994.

[13] R.-S. Tasy, “Exact zero skew,” *IEEE TCAD*, 1993.

[14] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI design—A Systems Perspective*, Addison-Wesley Pub. Company Inc., 1993.

TABLE I

EXPERIMENTAL RESULTS FOR A) ONE STAGE MAXIMUM SKEW REDUCTION B) ONE STAGE 50% SKEW REDUCTION
C) TWO STAGES MAXIMUM SKEW REDUCTION.

One Stage Maximum Skew Reduction

Ckt	#	Delay (ns)			Skew (ps)			Runtime
	Nodes	Initial	Final	Reduce%	Initial	Final	Reduce%	(sec)
r1	533	0.775	0.776	-0.13	64	0.188	99.71	0.71
r2	1195	2.108	2.628	-19.78	221	0.939	99.58	3.90
r3	1723	3.376	3.087	8.56	154	2.324	98.49	2.17
r4	3805	9.087	12.022	-24.41	716	1.845	99.74	9.62
r5	6201	15.864	20.154	-27.04	974	2.360	99.76	21.13
Average	-	-	-	-16.49	-	-	99.46	-

One Stage 50% Skew Reduction

Ckt	#	Delay (ns)			Skew (ps)			Runtime
	Nodes	Initial	Final	Reduce%	Initial	Final	Reduce%	(sec)
r1	533	0.775	0.551	28.90	64	31	51.56	0.16
r2	1195	2.108	1.649	21.77	221	110	49.88	0.65
r3	1723	3.376	2.785	17.50	154	80	48.05	1.42
r4	3805	9.087	7.777	14.42	716	246	65.64	4.57
r5	6201	15.864	14.765	6.93	974	490	49.69	10.08
Average	-	-	-	17.904	-	-	52.964	-

Two Stages Maximum Skew Reduction

Ckt	#	Delay (ns)			Skew (ps)			Runtime
	Nodes	Initial	Final	Reduce%	Initial	Final	Reduce%	(sec)
r1	533	0.775	0.159	79.48	64	0.004	99.99	1.17
r2	1195	2.108	0.372	82.35	221	0.059	99.97	1.13
r3	1723	3.376	0.501	85.16	154	9.609	93.76	1.18
r4	3805	9.087	1.573	82.69	716	3.338	99.53	5.36
r5	6201	15.864	2.834	82.14	974	1.134	99.88	9.86
Avg	-	-	-	82.364	-	-	97.850	-

[15] Q. Zhu, W. W.-M. Dai, and J. G. Xi, “Optimal sizing of high-speed clock networks based on distributed RC and lossy transmission line models,” *Proc. ICCAD*, pp. 628–633, 1993.

[16] L. T. Pillege, and R. A. Rohrer “Asymptotic waveform evaluation for timing analysis”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, April, 1990.