A Unified Theory for Mixed CMOS/BiCMOS Buffer Optimization

Takayasu Sakurai, *Member, IEEE*

***Abstract*—A simple yet realistic gate sizing theory is presented to optimize delay of a cascaded gate buffer. The theory is based on the fact that CMOS/BiCMOS gate delay is linearly dependent on fan-out f, that is, the delay can be expressed as $Af + B$, where A and B are coefficients. The optimum fan-out f_{OPT} is shown to be approximated as $e + B/1.5A$ for a gate chain. The theory covers various BiCMOS/CMOS gate types such as NAND's and NOR's in a unified framework. The existence of spurious capacitance is shown to increase the size of all transistors compared with the case without the spurious capacitance.**

I. Introduction

GATE sizing is one of the key techniques to improve speed of VLSI's. A simple theory [1] shows that the delay of cascaded gates is minimized when the size of the gates is tapered by a factor of $e(= 2.718)$. This principle is widely used in the initial design stage of memory VLSI's [2], macro designs, and ASIC's. This rule, however, is too simplistic in the sense that it neglects zero fan-out delay of logic gates and hence the accuracy is not practical. Since the theory does not cover various gate types, it is not applicable to cascaded gate buffers including mixed BiCMOS/CMOS gates and including various gate types such as inverters, NAND's, and NOR's. On the other hand, there are CAD tools to do the gate sizing [3] but the tools are based on the *RC* equivalent of MOSFET circuits and are not applicable to BiCMOS circuits. Moreover, CAD tools use numerical programming, which is not intuitive to designers.

In this paper, a new speed optimization theory is presented which is simple yet realistic and covers various BiCMOS/CMOS gate types within a unified framework. First, in Section II, notations are summarized and a delay model for logic gates used in this paper is described. In Sections III, IV, and V, the following cases are treated sequentially: the case where the number of gates in a buffer is fixed, the case where all gates in a buffer are the same type, and the case where certain logic gates are buffered by cascaded inverters. These cases cover most of the cases met in the real VLSI designs. The effect of spurious capacitances are considered in Section VI.

Manuscript received December 1, 1991; revised March 30, 1992.

The author is with the Semiconductor Device Engineering Laboratory, Toshiba Corporation, 1 Komukai-Toshiba-cho, Saiwai-ku, Kawasaki 210, Japan.

IEEE Log Number 9200924.

II. Delay Model of Logic Gates

In this paper, t_{pHL} (t_{pLH}) is defined as the delay from the time when a specific input crosses $0.5V_{DD}$ to the time when the falling (rising) output crosses $0.5V_{DD}$, and t_{pd} as $(t_{pHL} + t_{pLH})/2$. t_{p*} signifies either t_{pHL}, t_{pLH}, or t_{pd}. The size of a gate, w, is measured by size units (SU's). The input capacitance of one SU gate is equal to the input capacitance of a certain gate, for example, the CMOS two-input NAND gate in this paper. The input capacitance of one SU gate is called one load unit (LU) and all capacitances are measured in LU's.

Fig. 1 shows measured delay characteristics of CMOS/BiNMOS gates together with gate symbols used in this paper. t_{p*} can be well approximated by $Af + B$ even for BiNMOS gates as seen from Fig. 1, where A and B are constants and f signifies fan-out. The BiNMOS gate shown in Fig. 1 was first introduced in [7]. The measurement is carried out using a 0.5-μm BiCMOS channelless gate array (Fig. 2) [7], [8]. Although BiNMOS gates are used in the measurement, the present theory can be applicable to ordinary BiCMOS logic gates, too.

Considering that t_{p*} can be written as a linear combination of input transition term and a load capacitance term [4]–[6], A's and B's for a gate can be calculated from SPICE simulation by using the circuit and the method shown in Fig. 3. The output of the gate under test (node 2) is connected to one typical gate of 1 SU which acts as a monitor and one more typical gate of $f - 1$ SU which acts as a load capacitor. In this paper, the typical gate is chosen to be a two-input NAND gate. The input of the gate under test (node 1) is driven by a step voltage. Delays from node 1 to node 2 ($1 \rightarrow 2$ delay), and delay from node 1 to node 3 ($1 \rightarrow 3$ delay) are simulated with varying load capacitance (f). Suppose that $1 \rightarrow 3$ delay can be expressed as $p + Af$ and $1 \rightarrow 2$ delay can be expressed as $q + rf$. Then A is obtained from the slope of $1 \rightarrow 3$ delay dependence on f. Next, B can be calculated as $B = qA/r$. The detailed derivation is found in the Appendix.

By using this method, the effect of output voltage slope on the delay of the subsequent gate can be approximately incorporated in A and B. If node 2 is very capacitive, it slows down the voltage response of node 2 and this in turn slows down the gate response of the next gate ("monitor" in the case of Fig. 3). So the capacitive load on node 2 not only degrades the performance of the gate under test, but it further degrades the speed characteristics of the sub-

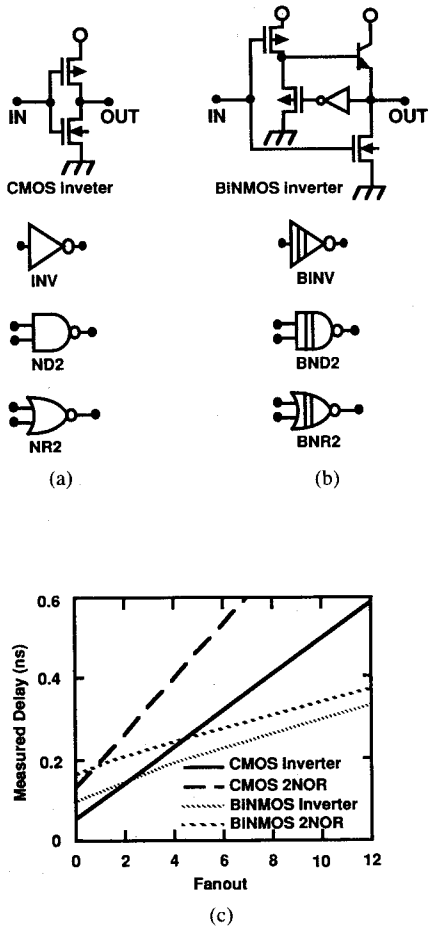


Fig. 1. Notations for (a) CMOS and (b) BiCMOS gates, and (c) measured delay dependence on fan-out for various CMOS/BiNMOS gates.

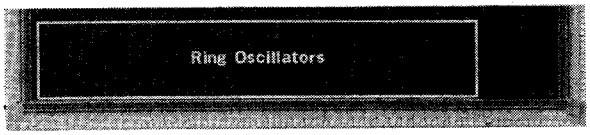


Fig. 2. Chip microphotograph of test structures on a 0.5-μm BiCMOS gate array.

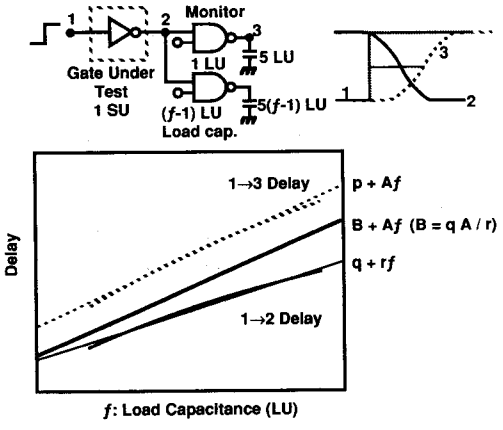


Fig. 3. Circuit and method for calculating A's and B's.

sequent gate. Since the subsequent gate is not always a two-input NAND gate, there is an approximation but an overall approximation is much better than just neglecting the slope-dependent delay effect. The explanation of the slope-dependent delay effect is found in [5] and [6]. The validity of the above-mentioned method can be verified by the overall success in the following sections. A's and B's for various gates in 0.5-μm BiCMOS technology are tabulated in Table I, as an example. Values for A's and B's can be found for example in the document supplied by ASIC vendors, though the calculation method for these constants might be different from organization to organization. The method described in Fig. 3 is not related to any specific organization.

III. Case 1: When the Number of Gates in a Buffer is Fixed

By using the delay model described in the previous section, the total delay D of the gate chain can be written as

$$D = \left(B_0 + A_0 \frac{w_1}{w_0}\right) + \left(B_1 + A_1 \frac{w_2}{w_1}\right) + \cdots + \left(B_{n-1} + A_{n-1} \frac{w_n}{w_{n-1}}\right). \quad (1)$$

w_i's are sizes of gates measured in SU's. By differentiating with respect to w_i's, it is easy to show that D is minimized when

$$A_0 \frac{w_1}{w_0} = A_1 \frac{w_2}{w_1} = \cdots = A_{n-1} \frac{w_n}{w_{n-1}} \equiv \tau_A. \quad (2)$$

By multiplying all the terms, we get

$$\tau_A^n = A_0 A_1 \cdots A_{n-1} \frac{w_n}{w_0} \rightarrow$$

$$\tau_A = \sqrt[n]{Y}\sqrt[n]{A_0 A_1 \cdots A_{n-1}}, \qquad Y \equiv \frac{w_n}{w_0}. \quad (3)$$

Therefore, the total delay D is rewritten as

$$D = \sum_{i=0}^{n-1} (B_i) + n\tau_A = \sum_{i=0}^{n-1} (B_i) + n(A_0 A_1 \cdots A_{n-1} Y)^{1/n}. \quad (4)$$

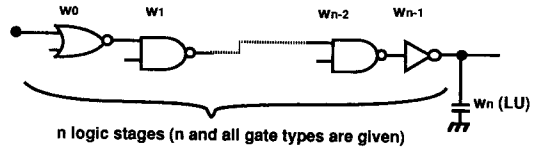
Formula (3) shows a strategy of gate delay optimization for a cascaded gate buffer. When Y is given, τ_A can be calculated, since all A_i's are known. Then, the size of gates, w_i, can be calculated by using (2) as follows:

$$w_1 = \tau_A \frac{w_0}{A_0}, \quad w_2 = \tau_A \frac{w_1}{A_1}, \cdots, w_i = \tau_A \frac{w_{i-1}}{A_{i-1}}, \cdots, w_{n-1} = \tau_A \frac{w_{n-2}}{A_{n-2}}. \quad (5)$$

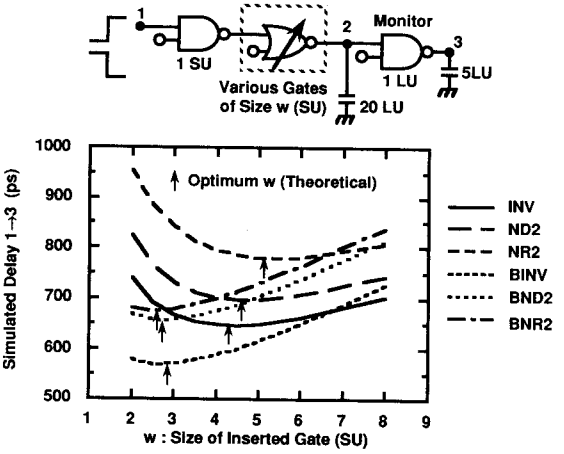
An application example of this theory is shown in Fig. 4. The theoretical values based on (1)–(3) are indicated by arrows, which show good agreement with SPICE simulation. In this example, t_{pd} is optimized but the optimization is also possible for the rising or falling input case

TABLE I
CALCULATED A AND B FOR 0.5-μm CMOS/BiCMOS GATES

Cell Name	Description	t_{pLH}			t_{pHL}			t_{pd}		
		A (ps)	B (ps)	f_{OPT}	A (ps)	B (ps)	f_{OPT}	A (ps)	B (ps)	f_{OPT}
INV	CMOS Inverter	42.4	37.9	3.3	21.0	33.0	3.7	31.7	35.5	3.5
ND2	CMOS 2NAND	42.1	53.0	3.5	33.3	68.5	4.1	37.7	60.8	3.8
NR2	CMOS 2NOR	72.6	124.2	3.8	21.1	57.8	4.5	46.9	91.0	4.0
BINV	BiNMOS Inv.	11.4	93.7	8.2	19.2	32.1	3.8	15.3	62.9	5.5
BND2	BiNMOS 2NAND	10.6	104.9	9.3	18.2	90.4	6.0	14.4	97.7	7.2
BNR2	BiNMOS 2NOR	12.3	178.7	12.4	13.0	36.9	4.6	12.7	107.8	8.4



(a)



(b)

Fig. 4. (a) Notations and (b) an example for size optimization of a mixed buffer.

in a similar manner, and it will give a different set of optimum gate sizes.

IV. Case 2: When All Gates in a Buffer are the Same Type

When all gates in a buffer are the same type, $A_0 = A_1 = \cdots = A_{n-1} = A$ and $B_0 = B_1 = \cdots = B_{n-1} = B$ hold. All tapering factors f_i, such as w_1/w_0, w_2/w_1, $\cdots$, w_n/w_{n-1} are the same and can be written as f. Consequently, D can be simplified to

$$D = \left(B + A\frac{w_1}{w_0}\right) + \left(B + A\frac{w_2}{w_1}\right) + \cdots + \left(B + A\frac{w_n}{w_{n-1}}\right) = n(B + Af). \quad (6)$$

Since $f^n = w_n/w_0 = Y$, n equals $\ln(Y)/\ln(f)$. Then the total delay D is expressed as follows:

$$D = n(B + A\sqrt[n]{Y}) = (B + Af)\frac{\ln Y}{\ln f}. \quad (7)$$

By differentiating in terms of f, the condition to give minimum delay is derived. f should be a root of the following equation:

$$\frac{f}{e}\ln\frac{f}{e} = \frac{B}{eA}. \quad (8)$$

It is impossible to analytically solve this transcendental equation but approximate formulas for the optimum f, f_{OPT}, are expressed as follows:

$$f_{OPT} \approx \frac{1}{2}\left(e^2 + 3\frac{B}{A}\right)\Big/\ln\frac{1}{2}\left(e^2 + \frac{B}{A}\right) \quad (9)$$

or

$$f_{OPT} \approx e + \frac{B}{1.5A}. \quad (10)$$

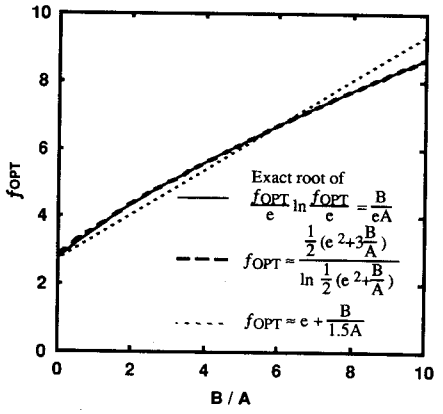


Fig. 5. Exact f_{OPT} and approximate formulas.

A graph of the exact root and the approximate formula is shown in Fig. 5. The expression $f_{OPT} \approx e + B/1.5A$ is quite simple and useful in the initial design stage. When B is set equal to zero, the f_{OPT} becomes e, which coincides with the simple theory [1]. An example for this case is shown in Fig. 6, where good agreement is observed between the theory and SPICE simulation.

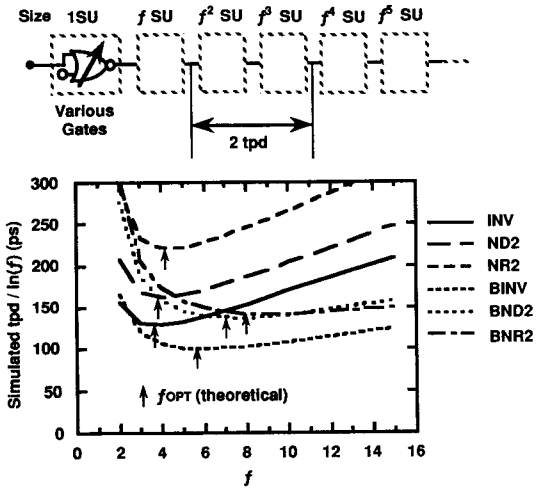


Fig. 6. Simulated delay dependence on f and the theoretical f_{OPT} (indicated by arrows).

V. Case 3: When Certain Logic Gates are Buffered by Cascaded Inverters

We often encounter the case where a certain combination of logic gates is buffered by cascaded CMOS/BiCMOS inverters as is shown in Fig. 7. In this case, the last k terms in the delay expression (1) have the same value:

$$D = \sum_{i=0}^{m-1} \left(B_i + A_i \frac{w_{i+1}}{w_i} \right) + \sum_{i=m}^{m+k-1} \left(B_m + A_m \frac{w_{i+1}}{w_i} \right). \tag{11}$$

By differentiating in terms of w_i, D is rewritten as follows, which corresponds to (4):

$$D = \sum_{i=0}^{m-1} B_i + kB_m + (m + k)(A_0 \cdots A_{m-1} A_m^k Y)^{1/m+k}. \tag{12}$$

Here, it is possible to introduce τ_A:

$$\tau_A = (A_0 \cdots A_{m-1} A_m^k Y)^{1/m+k} \tag{13}$$

and differentiate D in terms of k. The condition to minimize the delay is

$$\frac{\partial D}{\partial k} = B_m + \tau_A + (\ln A_m - \ln \tau_A)\tau_A = 0. \tag{14}$$

If a quantity τ_A/A_m is written as f_m, the condition for f_m to minimize the total delay is

$$\frac{f_m}{e} \ln \frac{f_m}{e} = \frac{B_m}{eA_m}. \tag{15}$$

This equation is identical to (8) and the root of the equation, f_{mOPT}, can be calculated by either (9) or (10). The optimum number of buffer stages, k_{OPT}, can be calculated

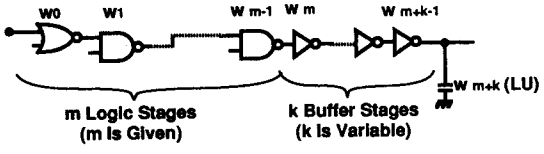


Fig. 7. Notation for buffered logic gates.

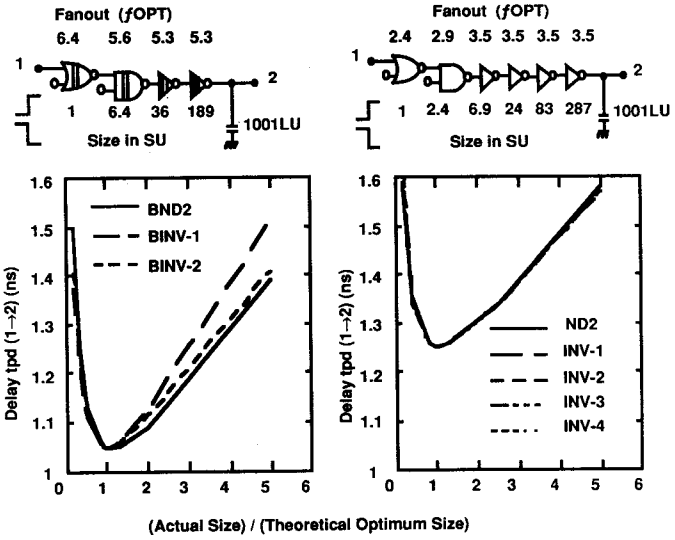


Fig. 8. An example of buffer optimization for CMOS and BiNMOS case.

by solving (13) in terms of k, and k_{OPT} is given as follows:

$$k_{OPT} = \frac{\ln \left(\frac{A_0}{A_m} \cdots \frac{A_{m-1}}{A_m} \right) + \ln Y}{\ln f_{mOPT}} - m. \tag{16}$$

Now that the number of gate stages gets determined and all gate types are given, the case is reduced to case 1 in Section III. An example of the optimization is shown in Fig. 8. In this case, $Y = 1001$, and k_{OPT}'s for CMOS and BiNMOS case are calculated as 1.9 and 3.7, respectively. It is observed that when the gate sizings deviate from the theoretical values, the total delay is deteriorated. It is also seen that the speed advantage of BiCMOS gates is about 20% over CMOS gates after the optimization in this example.

VI. Consideration on Spurious Capacitances

Spurious capacitances as shown in Fig. 9 are usually small and negligible compared with gate capacitances in a buffer, but there are cases where the spurious capacitances are not negligible. The effect of such capacitances on the optimization procedure is considered qualitatively in this section. The total delay D for this case is written as

$$D = \left(B_0 + A_0 \frac{w_1' + C_1}{w_0} \right) + \left(B_1 + A_1 \frac{w_2' + C_2}{w_1'} \right) + \cdots + \left(B_{n-1} + A_{n-1} \frac{w_n}{w_{n-1}'} \right) \tag{17}$$

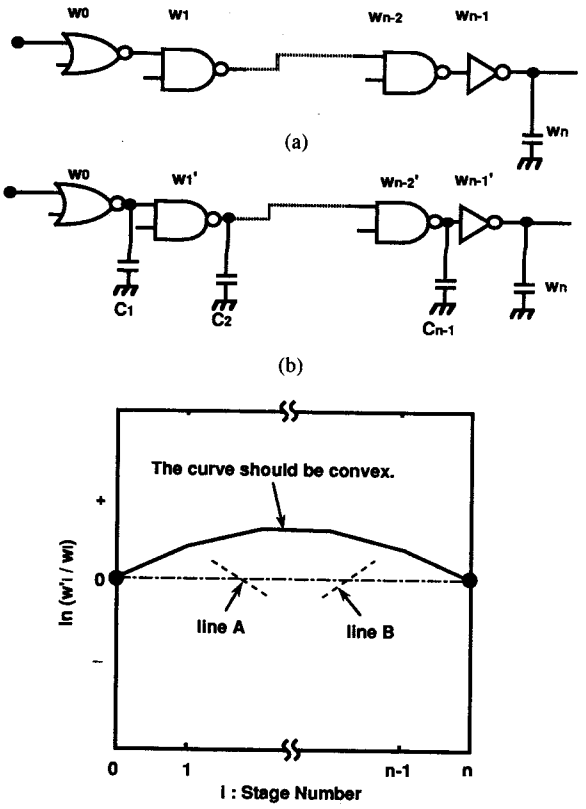


Fig. 9. Effect of spurious capacitances. (a) A buffer without spurious capacitances. (b) A buffer with spurious capacitances.

where C_i is the spurious capacitance on node i measured in LU's. By differentiating D with respect to w_i', we have

$$A_{i-2}\frac{w_{i-1}'}{w_{i-2}'} = A_{i-1}\frac{w_i' + C_i}{w_{i-1}'} > A_{i-1}\frac{w_i'}{w_{i-1}'} \quad (i = 2, \cdots, n-1). \tag{18}$$

Consequently, the following relations hold:

$$A_0\frac{w_1'}{w_0} > A_1\frac{w_2'}{w_1'} > \cdots > A_{n-1}\frac{w_n}{w_{n-1}'}. \tag{19}$$

On the other hand, in the case without the spurious capacitances, the following equations hold which are the same as (2):

$$A_0\frac{w_1}{w_0} = A_1\frac{w_2}{w_1} = \cdots = A_{n-1}\frac{w_n}{w_{n-1}}. \tag{20}$$

If (19) is divided by (20) term by term, we have the following inequalities if w_i'/w_i is written as η_i, and w_0' ($= w_0$) and w_n' ($= w_n$) are introduced:

$$\frac{w_1'/w_1}{w_0'/w_0} > \frac{w_2'/w_2}{w_1'/w_1} > \cdots > \frac{w_n'/w_n}{w_{n-1}'/w_{n-1}}$$

$$\frac{\eta_1}{\eta_0} > \frac{\eta_2}{\eta_1} > \cdots > \frac{\eta_n}{\eta_{n-1}},$$

$$\eta_0 = \eta_n = 1. \tag{21}$$

Taking the logarithm of each term, we have

$$\ln \eta_1 - \ln \eta_0 > \ln \eta_2 - \ln \eta_1 > \cdots > \ln \eta_n - \ln \eta_{n-1},$$

$$\ln \eta_0 = \ln \eta_n = 0. \tag{22}$$

A graph of η_i is shown in Fig. 9. If there is a line like line A in Fig. 9 and $\ln \eta_i$ gets negative, then to the right of line A there should be another line like line B which pulls up the value at least to zero from a negative value, since $\ln \eta_n$ should be zero. The existence of line B to the right of line A is contradictory to (22) because the slope of line B is larger than line A.

This means that the curve of $\ln \eta_i$ should be convex with both ends being fixed at 0. This in turn demonstrates that all $\ln (w_i'/w_i)$'s ($i = 1, 2, \cdots, n-1$) are positive. In conclusion, it has been shown that $w_1' > w_1$, $w_2' > w_2$, $\cdots$, and $w_{n-1}' > w_{n-1}$. An important message is that if there are spurious capacitances, the optimized transistor size in a buffer is always larger than that without the spurious capacitances.

VII. Conclusions

A simple yet realistic buffer optimization procedure is proposed. The theory covers BiCMOS and CMOS mixed cases in a unified framework and is applicable to a buffer including various gate types. Optimum tapering factor is found to be $e + B/1.5A$ for cascaded inverters. This reduces to e when $B = 0$ and in this sense the proposed theory is an extension of the simple e-tapering rule. Good agreements are seen between theoretical calculations and SPICE simulations.

The effect of spurious capacitances on the optimization is also investigated. It is shown that the spurious capacitances increase the size of all transistors in an optimized buffer.

Appendix
Obtaining A's and B's

Think about a system as shown in Fig. 10. The total delay is expressed as

$$\text{Total delay} = \{(P_0 t_{IN,0} + Q_0(C_{J,0} + C_{O,0})\} + \{(P_1 t_{IN,1} + Q_1(C_{J,1} + C_{O,1})\} + \cdots. \tag{A1}$$

In the expression, subscript J and O stand for junction capacitance contribution and output load capacitance contribution, respectively. P's and Q's are coefficients which do not vary even if the load capacitance condition varies. t_{IN} is input signal transition time. P_i is independent of the gate size but is a function of the gate type (see [5] for more detail). $C_{J,i}$ is an increasing function of the size of the gate i, and $C_{O,i}$ is an increasing function of the size of the gate $i + 1$. Q_i is inversely proportional to the size of the gate i. The important point is that P's and Q's are not a function of fan-out f.

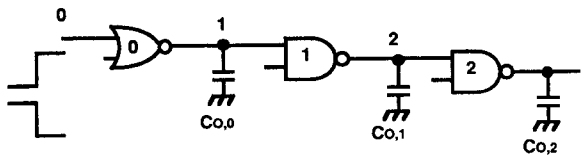


Fig. 10. Notation for logic gate chain.

The following delay is assigned to gate 0:

$$\begin{aligned}\text{Delay assigned to gate 0} &= Q_0(C_{J,0} + C_{O,0}) + P_1 t_{IN,1} \\ &= Q_0(C_{J,0} + C_{O,0}) + P_1 R_1 (C_{J,0} + C_{O,0}) \\ &= (Q_0 + P_1 R_1) C_{J,0} + (Q_0 + P_1 R_1) C_{O,0} \\ &= B_0 + A_0 f_0. \end{aligned} \tag{A2}$$

On the other hand,

$$\begin{aligned}\text{Delay } (0 \rightarrow 2) &= Q_0(C_{J,0} + C_{O,0}) \\ &\quad + (P_1 t_{IN,1} + Q_1(C_{J,1} + C_{O,1})) \\ &= ((Q_0 + P_1 R_1) C_{J,0} + Q_1(C_{J,1} + C_{O,1})) \\ &\quad + (Q_0 + P_1 R_1) C_{O,0} \\ &= p_0 + A_0 f_0 \end{aligned} \tag{A3}$$

$$\begin{aligned}\text{Delay } (0 \rightarrow 1) &= Q_0(C_{J,0} + C_{O,0}) = Q_0 C_{J,0} + Q_0 C_{O,0} \\ &= q_0 + r_0 f_0. \end{aligned} \tag{A4}$$

It is seen from these relationships that A_0 can be obtained from a slope of the delay $(0 \rightarrow 2)$ and B_0 can be obtained as

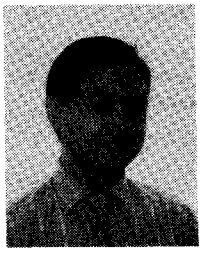
$$B_0 = (Q_0 + P_1 R_1) C_{J,0} = Q_0 C_{J,0} \frac{Q_0 + P_1 R_1}{Q_0} = q_0 \frac{A_0}{r_0}. \tag{A5}$$

Acknowledgment

The encouragement of H. Nakatsuka, K. Kanzaki, and Dr. S. Kohyama throughout the work is appreciated. Discussions with H. Hara, T. Nagamatsu, K. Seta, Mr. S. Kobayashi, and T. Kuroda were fruitful and are acknowledged here.

References

[1] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980.
[2] T. Sakurai *et al.*, "A 1 Mb virtually SRAM," in *ISSCC Dig. Tech. Papers*, Feb. 1986, pp. 252-253.
[3] J. Fishburn and A. Dunlop, "TILOS: A polynomial programming approach to transistor sizing," in *Proc. ICCAD*, Nov. 1985, pp. 326-328.
[4] N. Hedenstierna and K. O. Jeppson, "CMOS circuit speed and buffer optimization," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, no. 2, pp. 270-280, Mar. 1987.
[5] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its application to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584-594, Apr. 1990.
[6] T. Sakurai and A. R. Newton, "Delay analysis of series-connected MOSFET circuits," *IEEE J. Solid-State Circuits*, vol. 26, no. 2, pp. 122-131, Feb. 1991.
[7] H. Hara *et al.*, "0.5μm 2M-transistor BiPNMOS channelless gate array," in *ISSCC Dig. Tech. Papers*, Feb. 1991, pp. 150-151.
[8] T. Nagamatsu *et al.*, "A 1.9ns BiCMOS CAM macro with double match line architecture," in *Proc. IEEE Custom Integrated Circ. Conf.*, May 1991, pp. 14.3.1-14.3.4.



Takayasu Sakurai (S'77-M'78) was born in Tokyo, Japan, on January 10, 1954. He received the B.S., M.S., and Ph.D. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1976, 1978, and 1981, respectively. His Ph.D. work is on electronic structures of a Si-SiO_2 interface.

In 1981 he joined Semiconductor Device Engineering Laboratory, Toshiba Corporation, Kawasaki, Japan, where he was engaged in the research and development of CMOS dynamic RAM, 64- and 256-kb SRAM, 1-Mb virtual SRAM, cache memories, and BiCMOS ASIC's. He also worked on the modeling of interconnect capacitance and delay, soft-error free memory cells, new memory architectures, hot-carrier resistant circuits, arbiter optimization, and gate-level delay modeling. From 1988 through 1990 he was a Visiting Scholar at the University of California, Berkeley, doing research in the field of computer-aided design of VLSI's. He is currently back at Toshiba managing memory/logic VLSI development. His present interests include application-specific memories, BiCMOS VLSI's, VLSI microprocessors, FPGA's, and data compression/decompression VLSI.

Dr. Sakurai is a visiting lecturer at the University of Tokyo and a member of the Institute of Electronics, Information and Communication Engineers of Japan and the Japan Society of Applied Physics.