

An Iterative Gate Sizing Approach with Accurate Delay Evaluation

Guangqiu Chen

Hidetoshi Onodera

Keikichi Tamaru

Department of Electronics and Communication
Kyoto University

Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-01, Japan

Email: chen@tamaru.kuee.kyoto-u.ac.jp

Abstract

This paper introduces a new gate sizing approach with accurate delay evaluation. The approach solves gate sizing problems by iterating local sizing results from linear programming within small variable ranges of gate sizes. In each iterative step, variable ranges of gate sizes are updated according to the result from a previous step. Solutions with accurate delay evaluation which consider input signal slopes and separately evaluate rising and falling delays are obtained after several iterative steps. A speedup technique is used to pick out gates actually involved in each local sizing step so as to reduce CPU time. Experiments on sample circuits show that our approach can provide solutions with smaller circuit area than conventional approaches for the same circuit delay or provide solutions under tight delay constraints where conventional approaches can not reach. Moreover, our approach is faster than the conventional approaches for most circuits, especially under loose delay constraints.

1 Introduction

Gate sizing is a timing optimization process in high performance VLSI circuit design. In this design process, the size of each gate in a combinational circuit is properly tuned so that circuit area and/or overall power dissipation are minimized under specified timing constraints.

Gate sizing or the similar problem, transistor sizing, is an active research topic in recent years. Many approaches have been proposed [1-8]. Among them, a frequently used mathematical optimization technique for gate sizing is linear programming. Although the approaches may differ in subsidiary aspects, the way to formulate a gate sizing problem into linear program is similar, which is first proposed by Berkelaar and Jess in [1]. In the following text, we call such kind of gate sizing approaches as conventional approaches.

In the conventional approaches [1-5], gate delay is evaluated by a simple gate level delay model and approximated by a piecewise linear function under the assumption that it is a convex function with respect to its gate size and output capacitance. However, since real gate delay is not strictly convex, there are inevitable errors in delay evaluation in the conventional approaches. These errors will become larger when the variable ranges of gate sizes are larger. They can not be effectively reduced even if we use more detailed delay models or increase the number of piecewise linear regions because of the nonconvex nature of delay functions.

This paper proposes a new gate sizing approach with accurate delay evaluation. Our approach solves gate sizing problems by iterating local sizing steps. In each local sizing step, linear programming is used to solve problems locally within small variable ranges of gate sizes where gate delays can be approximated by linear functions. Iteration goes on by changing and decreasing the variable ranges of gate sizes according to the result from a previous step. Solution can be obtained after several steps of iteration when the results from local sizing steps can no longer be improved and the variable ranges of gate sizes are small enough.

Unlike conventional approaches, our approach can use any delay models. Moreover, the influence of input signal slope on delay is taken into account in the process of iteration, Rising and falling delays are evaluated separately. Although gate delays are approximated by linear functions in each local sizing step, delay evaluation is accurate enough because the nonlinearity of gate delays are small within small variable ranges of gate sizes.

Since the timing information of a circuit does not change drastically within small variable ranges of gate sizes, we can pick out gates actually involved in each local sizing step by evaluating the timing requirement for each gate beforehand. In this way, the number of variables in the corresponding linear programs are substantially reduced so that our approach becomes more efficient.

We have tested our algorithm on sample circuits including ISCAS85 benchmarks. Experiments show that our ap-

proach can provide results with smaller circuit area than conventional approaches under the same circuit delay or provide results under tight delay constraints where conventional approaches can not reach. Moreover, our approach is faster than the conventional approaches for most gate sizing problems, especially under loose delay constraints.

The rest of the paper is organized as follow. In Section 2 we will discuss the issue of delay evaluation in conventional gate sizing approaches. In Section 3 we will describe the proposed approach in detail. Section 4 will present experimental results and Section 5 will give conclusion and future work.

2 Delay evaluation in conventional approaches

The conventional approaches [1-5] evaluate delay by a simple gate level delay model which uses the worst case delay of a gate with respect to rising and falling delays and neglects the influence of input signal slope on delay. It can be expressed as

$$d_i = \tau_i + c_i \frac{C_L^i}{x_i} \quad (1)$$

where x_i and C_L^i are gate size and load capacitance of gate i respectively, τ_i and c_i are constants. Since C_L^i is determined by the sizes of its fanout gates and wiring capacitances, the delay of a gate can be expressed as a function of gate sizes in the circuit if wiring capacitances are assumed to be constants. Eq.(1) is further approximated by a piecewise linear function in the form of

$$d_i = \begin{cases} p_{01}x_i + \sum_{j \in \text{fanout } i} p_{j1}x_j + p_{c1} & X \in R_1 \\ p_{02}x_i + \sum_{j \in \text{fanout } i} p_{j2}x_j + p_{c2} & X \in R_2 \\ \vdots & \vdots \\ p_{0l}x_i + \sum_{j \in \text{fanout } i} p_{jl}x_j + p_{cl} & X \in R_l \end{cases} \quad (2)$$

where x_i is the size of gate i , $x_j (j \in \text{fanout } i)$ are the sizes of fanout gates of gate i , and p 's are constants. In this formula, $X = \{x_i | i = 1 \dots n\}$ represents the gate size vector. The delay function is divided into l piecewise linear regions R_1, R_2, \dots, R_l .

If the delay of gate i is assumed to be a convex function of gate sizes, Eq.(2) is equivalent to

$$d_i = \max_{k=1 \dots l} (p_{0k}x_i + \sum_{j \in \text{fanout } i} p_{jk}x_j + p_{ck}). \quad (3)$$

Finally, Eq.(3) is transformed into a set of linear constraints

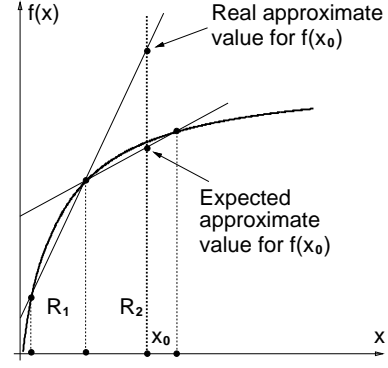


Fig.1 Errors in linear approximation due to nonconvexity

$$d_i \geq p_{0k}x_i + \sum_{j \in \text{fanout } i} p_{jk}x_j + p_{ck} \quad (k = 1, \dots, l) \quad (4)$$

in the linear program.

Since gate delay is not a strict convex function of gate sizes, there are errors when Eq.(2) is approximated by Eq.(3). The reason of this kind of error can be explained by a conceptual example in Fig.1. Because of the nonconvexity of function $f(x)$, the maximal value of all linearizing functions in some region may not be the one which approximates the original function for this region.

Moreover, the simple delay model of Eq.(1) is not accurate enough. In practice, gate delay is also influenced by its input signal slope. Rising and falling delays may be quite different for some gates in a circuit. It's claimed that the conventional approaches can also use more accurate delay models as long as they are convex. However, since real gate delay is not strictly convex, accurate delay models will introduce more nonconvex factors into delay functions so that the delay evaluation may conversely deteriorate. Improving the precision of delay model will not necessarily improve the precision of delay evaluation in the conventional approaches.

As a result, the conventional approaches [1-5] are usually applied to gate sizing problems under loose delay constraints where the precision of delay evaluation is not so important and with small variable ranges of gate sizes where errors in delay evaluation resulting from nonconvexity is not large.

3 Proposed gate sizing approach

Delay evaluation in gate sizing process is important because the size of a gate is determined by the timing requirement on this gate. In this paper, we propose a new gate sizing approach based on iteration of local sizing steps. In

each local sizing step, a gate sizing problem is formulated into a linear program within small variable ranges of gate sizes. Iteration goes on by modifying and decreasing the variable ranges of gate sizes. Results from a previous local sizing step are used to determine the variable ranges of gate sizes for the next step. After several steps of iteration when results can no longer be improved and the variable ranges of gate sizes become small enough, solution is obtained.

Moreover, a speed-up technique is used in our approach to reduce CPU time. The speed-up technique picks out gates which may be sized in each local sizing step and formulates these gate size variables only into linear programs so that the size of linear programs can be substantially reduced.

3.1 Formulation for local sizing step

In our approach, the linear program for each local sizing step is formulated as

$$\begin{aligned}
 & \text{minimize : } \sum_{i=1}^n K_i x_i + \alpha M \\
 & \text{subject to : } \\
 & \left\{ \begin{array}{l} d_i^r \geq p_0^r x_i + \sum_{j \in \text{fanout } i} p_j^r x_j + p_c^r \\ d_i^f \geq p_0^f x_i + \sum_{j \in \text{fanout } i} p_j^f x_j + p_c^f \\ t_k^{f/r} + d_i^r \leq t_i^r \\ t_k^{r/f} + d_i^f \leq t_i^f \\ t_m^r \leq T_{\text{spec}} + M \\ t_m^f \leq T_{\text{spec}} + M \\ X_{\text{lower}}^i \leq x_i \leq X_{\text{upper}}^i \end{array} \right. \quad (5)
 \end{aligned}$$

for every gate i ($i = 1, 2, \dots, n$) in the circuit. In this formula, x_i , d_i^r , d_i^f , t_i^r and t_i^f are gate size, delay and signal schedule time at the output of gate i . They are variables of the linear program. Subscript j and k denote fanout and fanin gates of gate i . Subscript m denotes gates at primary outputs. Variable M is a relax factor for circuit delay constraints. Parameter α is a large constant to keep M as small as possible. Coefficient K_i is a weighting factor of circuit area or power dissipation on the size of gate i . Superscripts r and f refer to rising and falling signals respectively. Parameters $t_k^{f/r}$ and $t_k^{r/f}$ represent different variables for different logic gates. Their meanings are

$$t_k^{f/r(r/f)} = \begin{cases} t_k^{r(f)} & k \in \text{negative logic} \\ t_k^{f(r)} & k \in \text{positive logic} \\ t_k^r \text{ and } t_k^f & k \in \text{exclusive logic} \end{cases} \quad (6)$$

This formula has several points different from the conventional ones [1-5]. First, gate delay is not approximated by a piecewise linear function but simply a linear function. Thus, there is no requirement that gate delay be a convex function. Any delay model can be used. In our algorithm, we use the analytical delay model introduced in Ref.[9].

Second, the variable range of gate size for gate i $[X_{\text{lower}}^i, X_{\text{upper}}^i]$ is only a small interval within its feasible range of gate size $[X_{\text{min}}^i, X_{\text{max}}^i]$. In this small variable range, the nonlinearity of gate delay is small. Therefore, it's reasonable to approximate it by a linear function of gate sizes. Although local sizing steps are defined within small variable ranges of gate sizes, our approach searches for the global solution in the feasible range of gate sizes by iteration. Details of the search strategy will be discussed later.

Third, rising(r) and falling(f) delay of gates are separately formulated so that delay evaluation is more accurate. In each local sizing step, input signal slopes of gates do not change very much due to the small variable range of each gate. Therefore, we can reasonably estimate typical input signal slopes for all gates at the beginning of each local sizing step. Since we update the estimates at every iteration, the influence of input signal slope on delay is considered in the iterative gate sizing process.

As a result, delay evaluation in our approach can be quite accurate as long as the variable ranges of gate sizes in local sizing steps are small enough and the delay model itself is accurate enough.

3.2 Iterative searching strategy

Based on Eq.(5), our approach performs an iterative searching process to find out the solution within whole feasible ranges of gate sizes. Fig.2 shows the algorithm of our approach. Here, F is the feasible solution space composed of all feasible ranges of gate sizes and V is the solution space composed of all variable ranges of gate sizes in one iterative step.

In the algorithm, the input signal slopes of gates T_{in} and the longest path delay in the circuit T_{circuit} are calculated at the beginning of each local sizing step by timing analysis. After each local sizing step ("LP_Solve"), function "Var_space" modifies X_{lower} and X_{upper} in Eq.(5) for every gate and use them in the next iterative step. The process ends when there is no obvious improvement within a certain number of iterative steps or the number of iterative steps is larger than a specified limit. During iteration, the best result is recorded and used as the solution of this gate sizing problem.

To calculate variable ranges of gate sizes for local sizing steps in the iteration, we define two parameters

$$\begin{aligned} X_{mid} &= (X_{upper} + X_{lower})/2 \\ \Delta X &= (X_{upper} - X_{lower}). \end{aligned} \quad (7)$$

for the variable range of each gate size $[X_{lower}, X_{upper}]$. Here X_{mid} and ΔX are called the middle point and span of the variable range. These two parameters for each gate can be determined by the result of previous local sizing step. Usually, parameter X_{mid} for the variable range of a gate is chosen to be as close as possible, preferably equal, to the resulting size of this gate from the previous local sizing step. Parameter ΔX is specified at the beginning of the algorithm and decreases by a specific rate for each iterative step. According to X_{mid} and ΔX , the variable range, i.e. X_{lower} and X_{upper} can be determined by Eq.(7).

3.3 Speed-up technique

In a gate sizing process, only gates related to critical paths may be sized. Other gates will always take their minimal possible sizes. The speedup heuristics used in our approach picks out gates actually involved in each local sizing step and formulates them into linear programs in order to reduce the size of linear programs.

Because of small variable ranges of gate sizes, the timing information of a circuit does not change drastically in a local sizing step. Therefore, it's possible for us to predict timing information of the resulting circuit roughly before gates are sized in each local sizing step. According to this timing information, we can pick out active gates, i.e. gates which may be sized in the local sizing step.

Fig.3 shows our heuristics to pick out active gates. In this heuristics, we first calculate the timing slack of each gate when all gates in the circuit are at their minimum sizes. Here, the slack of a gate is defined as the difference of required arrival time and actual arrival time at the output of this gate.

There are two procedures to pick out active gates. In the first procedure, we just pick out gates whose slacks are negative. In the second iterative procedure, we calculate the increase in delay for candidate gates, i.e. gates whose fanout gates contain at least one active gate, by enlarging the sizes of all current active gates to their maximal values. If the increase in delay of a candidate gate is larger than its slack value, it is picked out as a new active gate. When all candidate gates are so evaluated, the procedure enters the next iterative step which considers new active gates. The procedure stops when no new active gates can be found.

4 Experimental results

Our approach is implemented and tested on a Sun SPARCstation 20 with the C language. Sample circuits in-

```

Gate_Sizing()
{
   $V_0 \subset F$ ;
   $i=0$ ;
  while (improved &&  $i \leq i\_limit$ ) {
     $T_{in}, T_{circuit} \leftarrow \text{Simulate}(X_{min} \in V_i)$ ;
    if ( $T_{circuit} \geq T_{spec}$ )
       $X \leftarrow \text{LP\_Solve}(V_i)$ ;
    else  $X \leftarrow X_{min}$ ;
    if (improved)  $X_{opt} \leftarrow X$ ;
     $V_{i+1} \leftarrow \text{Var\_Space}(V_i, X_{opt}, F)$ ;
     $i++$ ;
  }
}

```

Fig.2 The algorithm of gate sizing process

```

Pick_Out_Active_Gate()
{
   $Delay \leftarrow \text{Delay\_Evaluation}(X_{min})$ ;
   $T_{required}, T_{actual} \leftarrow \text{Time\_Evaluation}(Delay)$ ;
   $Slack = T_{required} - T_{actual}$ ;
  /* Procedure 1 */
  for (each  $gate_i$ ) {
    if ( $slack_i < 0$ ) /*  $slack_i \in Slack$  */
       $Active\_Gate\_Set \leftarrow gate_i$ ;
  }
  /* Procedure 2 */
  do {
    for (each  $gate_i$ ) {
      if ( $\text{Fanout}(gate_i) \cap Active\_Gate\_Set \neq \emptyset$ )
         $Candidate\_Set \leftarrow gate_i$ ;
    }
    for (each  $gate_i \in Candidate\_Set$ ) {
       $C_L^i = 0$ ;
      for (each  $gate_j \in \text{Fanout}(gate_i)$ ) {
        if ( $gate_j \in Active\_Set$ )
           $C_L^i += \text{Capacitance}(x_j \in X_{max})$ ;
        else
           $C_L^i += \text{Capacitance}(x_j \in X_{min})$ ;
      }
       $\Delta delay_i \leftarrow \text{Delay\_Increment}(C_L^i - C_{min}^i)$ ;
      if ( $slack_i < \Delta delay_i$ ) /*  $slack_i \in Slack$  */
         $Active\_Gate\_Set \leftarrow gate_i$ ;
    }
  } while ( $Active\_Gate\_Set$  changed)
}

```

Fig.3 A process to pick out active gates in gate sizing

cluding most ISCAS85 benchmarks are used to test the feasibility of our algorithm. The gate delays in a circuit are evaluated by the approach introduced in Ref.[9]. A typical $0.8\mu\text{m}$ technology is used to set the parameters in this delay model. The size of a gate is measured by the size of a unit transistor (U). We assume that the feasible range of gate size for each gate is between 1U and 10U.

An algorithm based on the conventional approaches [1-5] is also implemented to compare its results with the proposed approach. In the conventional algorithm, we use the simple delay model of Eq.(1) and approximate gate delay by 10×1 piecewise linear functions, i.e. the delay function is divided into 10 regions with respect to its gate size and 1 region with respect to the size of its fanout gates. After a gate sizing problem is solved by the conventional approach, the resulting circuit is also evaluated by the delay modeling approach of Ref.[9] so that delay values from two approaches are comparable.

Fig.4 is complete resulting curves of delay vs. area for circuit c432. When delay constraints are not tight, both conventional and proposed approaches provide almost the same results. This is reasonable because under loose delay constraints, only a small percent of gates actually takes part in the gate sizing process. The increase in size of these gates has less influence on the total circuit area. However, when delay constraints become tight, the proposed approach provides better results, i.e. results with smaller area under the same circuit delay, than the conventional approach. For example, at the solution where resulting circuit delay is 32ns, the circuit area from the proposed approach is 956U while that from the conventional approach is 1444U. The area from the proposed approach is only 66% of that from the conventional approach.

To demonstrate the advantage of our iterative approach (proposed) over piecewise linear approach (conventional), we ran another experiment where we use the same delay evaluation technique as the proposed approach but solve the gate sizing problem by piecewise linear delay formulation used in the conventional approach. The delay vs. area curve of this piecewise linear (PWL) approach are also shown in Fig.4. As we can see in Fig.4, the results of this approach are even worse than the conventional approach which uses a very simple gate level delay model. This experiment shows that further refinement in delay model for the conventional approach is meaningless because of the nonconvex nature in delay functions.

Table 1 has listed the sample results from both conventional and proposed approaches for large ISCAS85 benchmark circuits. In Table 1, " T_{spec} " represents specified delay limits for gate sizing problems, "Delay" shows the resulting delay values evaluated by the delay modeling approach in Ref.[9] after gates are sized. Because of inaccu-

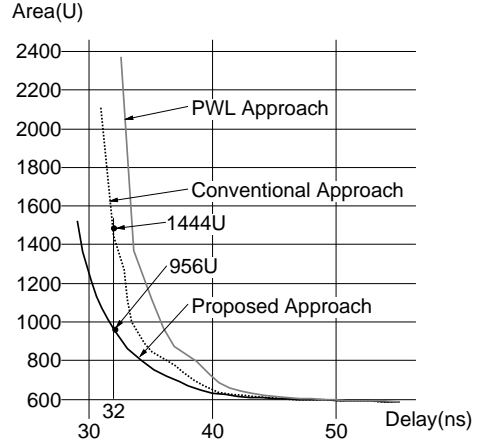


Fig.4 Results for circuit c432

racy in delay evaluation in the conventional approach, actual circuit delay after gate sizing is different from the specified delay constraints. For example, the circuit with delay 52.9ns should be obtained at the specified delay of 60ns for c1908. On the other hand, actual delay and specified delay are close for the resulting circuits from the proposed approach. To make results of two approaches comparable, the specified delay limits (T_{spec}) of the proposed approach are adjusted to the resulting delay values of the conventional approach. The sign “—” in Table 1 means that the approach can not provide results in this case. Column R_{active} shows the average percent of active gates in all gates of a circuit which are picked out by our speedup heuristics. Only this part of gates are actually formulated into linear programs.

From Table 1, we know that the conventional approach of our implementation can only provide results under relatively loose delay constraints. Under tight delay constraints, the algorithm failed because of numerical instability in the linear program. On the other hand, the proposed approach can offer results under a wider range of delay constraints.

As a result of these experiments, we can conclude that, under loose delay constraints, the proposed approach is much faster than the conventional approach for most circuits due to the speedup heuristics incorporated in the algorithm. Under tight delay constraints, the proposed approach can provide results with smaller circuit areas under the same circuit delay than the conventional approach in comparable CPU time and provide results with tight delay limits that the conventional approach can not reach.

5 Conclusion and future work

We have proposed a new LP based gate sizing approach with accurate delay evaluation. The approach can provide results with smaller circuit area under the same circuit de-

Table 1 Results for ISCAS85 Benchmarks

ckt.	#Gate	Conventional Approach				Proposed Approach				
		T_{spec}	$Delay$	$Area$	$CPU's$	T_{spec}	$Delay$	$Area$	$CPU's$	R_{active}
c1908	880	60ns	52.9ns	2747U	116s	53ns	52.9ns	2746U	35s	15%
		55ns	48.0ns	2817U	137s	48ns	48.1ns	2794U	153s	24%
		50ns	—	—	—	40ns	40.0ns	3149U	1003s	46%
c2670	1193	60ns	52.8ns	4331U	356s	53ns	52.9ns	4323U	80s	11%
		55ns	48.4ns	4401U	385s	48ns	48.0ns	4376U	126s	15%
		50ns	—	—	—	35ns	35.2ns	5314U	1476s	30%
c3540	1669	85ns	67.7ns	5226U	451s	67ns	67.5ns	5204U	288s	21%
		80ns	64.3ns	5361U	627s	64ns	64.1ns	5246U	685s	27%
		75ns	—	—	—	55ns	55.1ns	5606U	2973s	38%
c5315	2307	80ns	62.9ns	7480U	864s	63ns	63.1ns	7479U	54s	6%
		75ns	59.8ns	7560U	1027s	60ns	60.0ns	7522U	121s	10%
		70ns	—	—	—	40ns	40.1ns	10635U	1161s	22%
c6228	2406	250ns	168.2ns	7412U	1423s	168ns	167.7ns	7404U	2414s	34%
		240ns	159.0ns	7513U	1880s	159ns	159.0ns	7482U	6198s	44%
		230ns	—	—	—	150ns	149.9ns	7659U	21473s	51%
c7552	3512	60ns	50.9ns	11211U	2213s	51ns	51.1ns	11201U	238s	10%
		55ns	46.7ns	11458U	2917s	47ns	47.2ns	11320U	511s	11%
		50ns	—	—	—	40ns	40.1ns	11940U	6328s	25%

lay than conventional approaches for gate sizing problems under a wide range of delay constraints within wide variable ranges of gate sizes in reasonable CPU time. Our future work will focus on the application of our approach to practical power optimization problems and refine the delay evaluation by considering path sensitization.

References

- [1] M. R. C. M. Berkelaar and J. A. G. Jess, "Gate sizing in MOS digital circuits with linear programming," *Proc. EDAC'90*, pp. 217-221.
- [2] M. R. C. M. Berkelaar, P. H. W. Buurman and J. A. G. Jess, "Computing the entire active area/power consumption versus delay trade-off curve for gate sizing with a piecewise linear simulator," *Proc. ICCAD'94*, Nov. 1994, pp. 474-480.
- [3] G. Chen, H. Onodera and K. Tamaru "Experiments with power optimization in gate sizing," *IEICE Trans., Fundamentals*, Vol. E77.A, pp. 1913-1916, No. 11, Nov. 1994.
- [4] W. Chuang and I. N. Hajj, "A unified algorithm for gate sizing and clock skew optimization to minimize sequential circuit area," *Proc. ICCAD'93*, Nov. 1993, pp. 220-223.
- [5] Y. Tamiya, Y. Matsunaga and M. Fujita, "LP based cell selection with constraints of timing, area and power consumption," *Proc. ICCAD'94*, Nov. 1994, pp. 378-381.
- [6] J. P. Fishburn and A. E. Dunlop, "TILOS: a posynomial programming approach to transistor sizing," *Proc. ICCAD'85*, pp. 326-328.
- [7] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya and S. M. Kang, "An exact solution to the transistor sizing problem for CMOS circuits using convex optimization," *IEEE Trans. Computer-Aided Design*, Vol.12, No.11, Nov. 1993.
- [8] D. Maple, "Transistor size optimization in the tailor layout system," *Proc. 26th ACM/IEEE Design Automation Conf.*, June 1989, pp. 43-48.
- [9] T. Sakurai and A. R. Newton, "A simple MOSFET model for circuit analysis," *IEEE Trans. Electron Devices*, Vol.38, No. 4, pp. 887-893, Apr. 1991.
- [10] T. Sakurai and A. R. Newton, "Delay analysis of series-connected MOSFET circuits" *IEEE J. Solid-State Circuits*, VOL.26, No.2, Feb. 1991.