

# LOW POWER CLOCK DISTRIBUTION USING MULTIPLE VOLTAGES AND REDUCED SWINGS<sup>1</sup>

Jatuchai Pangjun and Sachin S. Sapatnekar  
Department of Electrical and Computer Engineering  
University of Minnesota  
Minneapolis, MN 55455

***Abstract:** Clock networks account for a significant fraction of the power dissipation of a chip and are critical to performance. This paper presents theory and algorithms for building a low power clock tree by distributing the clock signal at a lower voltage, and translating it to a higher voltage at the utilization points. Two low power schemes are used: reduced swing and multiple supply voltages. We analyze the issue of tree construction and present conclusions relevant to various technology generations according to the NTRS. Our experimental results show that power savings of an average of 45% are possible for a 0.25 $\mu$ m technology using multiple supply voltages, and about 32% using a single external supply voltage.*

## 1. INTRODUCTION

The clock network constitutes one of the most important parts of a synchronous VLSI chip as it can significantly influence the speed, area, and power dissipation of the system. Recent research on clock network construction [2,3,5,6,10,11,13,14,15] has developed procedures for building a zero or near-zero skew clock network with sharp clock edge rates at the clock utilization points.

However, one major drawback associated with clock networks is their power dissipation. Studies have shown that the clock network can dissipate 20-50% of the total power on a chip. In the context of the growing importance of low power designs for portable electronics, it is necessary to develop strategies to significantly reduce the power dissipation of the clock network, since this will lead to a major reduction in the overall power dissipation of the chip. The work in this paper is based on the observation that by using a lower V<sub>dd</sub> to distribute the signal over the chip, the clock network can be made to dissipate less power. However, for reasons related to performance requirements, the rest of the circuitry on the chip may use a higher V<sub>dd</sub>, and this implies that the clock levels would have to be converted to this higher V<sub>dd</sub> value at the utilization points.

The problem of clock tree synthesis for zero skew has been widely researched. Early techniques [10,11] relying on equalizing the path length to each sink were found to be inadequate for RC interconnects, and a method that explicitly equalizes the delay to each sink of the tree was proposed by Tsay [14]. This method performs a recursive bottom-up combination of two zero-skew subtrees by finding a tapping point to ensure zero skew in the larger subtree thus formed. While Tsay's algorithm suggested a framework to build zero-skew subtrees, it did not try to minimize the total wire length. The Deferred-Merged Embedding (DME) method [2,3,5] optimally embeds a given clock tree topology in the Manhattan plane with zero skew and attempts to minimize the total wire length. In addition to zero skew, a second requirement on a clock tree is that the slew rate for the clock edge must be sharp. This requires the insertion of

---

<sup>1</sup> This research was supported in part by the SRC under contract 98-DJ-609, by the NSF under award CCR-9800992, and by a scholarship from the Royal Thai Air Force.

buffers within the clock network to isolate the downstream capacitance, thus reducing the transition times. Various buffering algorithms have been proposed that can be used either directly or adapted for clock routing [7,13,17].

The high power dissipation in the clock net is primarily due to the large amount of capacitance driven by the clock net, in conjunction with the fact that the clock net switches on every transition. The total power dissipation consists of two components

- (a) the static power dissipation, which is due to a leakage current of transistors during steady state.
- (b) the dynamic power dissipation, which has two components: short circuit and charge/discharge of capacitance power dissipation. The short circuit power dissipation is a function of the slew rate of the input voltage; the sharper the clock edge, the lower the short circuit power dissipation.

In this work, we consider the static power dissipation to be negligible. We control the short-circuit power dissipation by enforcing a constraint that the clock edge should never have a transition (rise/fall) time that is larger than a given specification throughout the clock tree. By enforcing this sharp clock edge rate requirement, the short circuit power is bounded and can be neglected in comparison with the charge/discharge power. Note that the sharp clock edge is also an inherent requirement in the problem of clock network construction, and this approach effectively achieves two objectives for the price of one.

Therefore, the major component of the total power dissipation is the power that is used to charge and discharge the load capacitance in the circuits. Equation (1) below lists a well-known expression for the charge/discharge power dissipation.

$$P = fC_L V_{dd} V_s \quad (1)$$

where  $f$  is the clock frequency,  $C_L$  is the load capacitance,  $V_{dd}$  the supply voltage, and  $V_s$  the output swing of the buffer. For the case where the output of the buffer swings from 0 to  $V_{dd}$ ,  $V_s = V_{dd}$  and the formula reduces to  $P = f C_L V_{dd}^2$ .

Since  $f$  is a fundamental parameter for the circuit, it cannot be changed and its effects can only be reduced by techniques such as clock gating, which are not addressed in this work. Therefore, the power dissipation of the clock network can only be reduced by

- (a) reducing the total load capacitance, which is consistent with attempting to achieve the minimal wire length and the minimal buffer power dissipation; techniques for minimal wire length have been extensively addressed in the past, for example in [2,3,5,15].
- (b) reducing  $V_{dd}$ , which creates a quadratic reduction if  $V_s$  is also simultaneously reduced by the same factor (for example, when  $V_{dd} = k V_s$  for some value of  $k$ )
- (c) reducing  $V_s$  without reducing  $V_{dd}$ , which corresponds to a linear reduction in the power dissipation.

Since clock network construction techniques for minimizing the wire length lie in a well-researched area, our paper will use existing results on that subject to build clock trees and will, instead, focus on buffer insertion issues and low voltage clock circuits.

The objective of this work is to present new theory and results for building low power clock trees using a smaller voltage to distribute the signal over the chip, and then converting this low voltage clock signal back to a higher voltage at the utilization points. The organization of this paper is as follows. Section 2 states the problem and discusses some possible buffer schemes. Next, in Section 3, we derive a set of theoretical results and make empirical observa-

tions based on technology trends as predicted by the NTRS [12]. This is followed by an outline of the clock network construction algorithm in Section 4. Finally, we present our experimental results in Section 5 and 6, followed by a conclusion in Section 7.

## 2. STATEMENT OF THE PROBLEM

### 2.1 Structure of the Clock Tree

A clock scheme with multiple supply voltages was proposed in [9] and is illustrated in Figure 1 below.

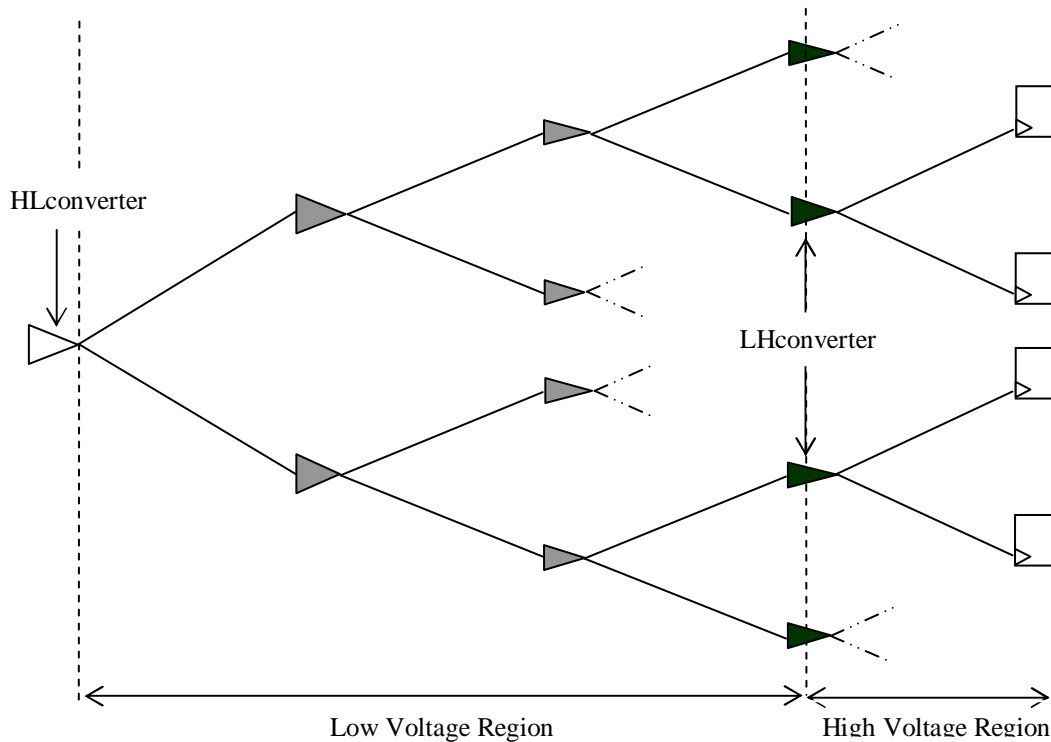


Figure 1: Low Power Clock Scheme

In Figure 1, HLconverter (High-to-Low converter) is a buffer that converts the incoming clock signal to the chip from a high voltage swing to a lower voltage swing. Alternatively, the input clock could be generated so that it has a low voltage swing. The clock signal is then transmitted on the chip as a low voltage signal, thereby ensuring a low power clock distribution network. At the points of utilization at the sink flip-flops, it is converted using the LHconverter (Low-to-High converter) block to the higher voltage swing, which is the voltage at which it is used by the logic network. Intermediate buffers are used in the clock tree (not necessarily at every branching point) to regenerate the signal and maintain a sharp slew rate as the signal passes through the network. The work in [9] implicitly assumes that in order to save the maximum power, the low voltage region must be maximized, thus minimizing the high voltage region. The work proposes, without a specific justification, that an HLconverter is inserted at the

root of the clock tree, and LHconverters are inserted at the clock sinks, thereby placing the entire clock tree in the low voltage region.

However, the flaw in this reasoning is that while such an approach minimizes the interconnect power dissipation, the cost of the LHconverters is not taken into account. Since an LHconverter consists of a number of transistors, placing an LH converter at every sink could lead to the addition of an excessive number of transistors, which could have an adverse impact not only on the power, but also on the layout and the area of the chip.

In this work, we propose to investigate the tradeoffs that are required to be made in such an approach. We will present criteria required to determine both a minimum power and a minimum area solution, and show some tradeoffs between the two, drawing conclusions based on realistic technology parameters.

The delay model used here is the well-known Elmore delay model, as in other work on the subject [2,3,5,6,14,15]. The slew transition time at a node is taken to be twice the Elmore delay from the previous buffer to that node.

## 2.2 Level Converter Circuits

### 2.2.1 A Level Converter Using Multiple Supply Voltages

The structure of the HLconverter is relatively straightforward. To convert the clock swing from a higher voltage range of gnd to  $V_{ddH}$  to a lower voltage range of gnd to  $V_{ddL}$ , a conventional buffer driven by a supply voltage of  $V_{ddL}$  will be adequate. The structure of an LHconverter is more involved, and the design we use is the same as that utilized in earlier work (for example, [9,16]) and is illustrated in Figure 2. We note that the use of feedback in the part driven by  $V_{ddH}$  serves to speed up the transition and therefore ensures that the transient current is not significant; our simulations show that it is, in fact, less than the transient current for a single inverter driven by  $V_{ddH}$ .

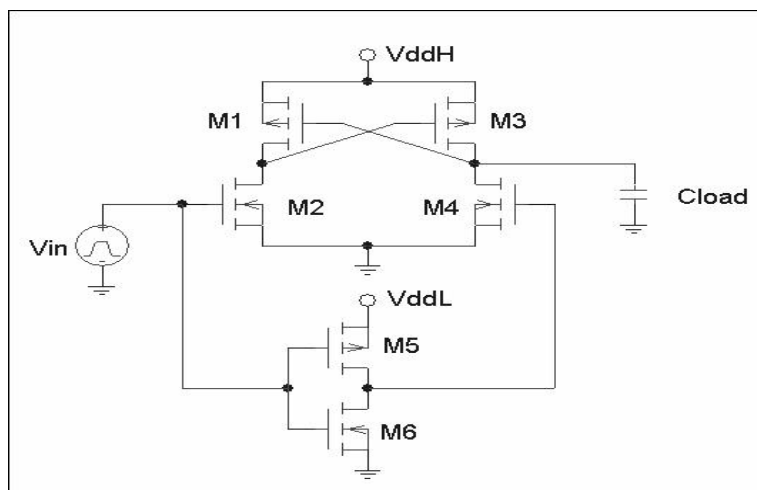


Figure 2: An LHconverter circuit

### 2.2.2 A Level Converter Using a Reduced Clock Swing

Equation (1) suggests that another variable that can be adjusted to reduce the power dissipation is the output clock swing. Zhang and Rabaey [19] show that reducing the voltage swing of the signal on a wire is a good approach for achieving improved energy efficiency. The work in [19], however, presented only the signal drivers and receivers, and did not discuss intermediate drivers that could be used for regenerating the signal as it is propagated at a low voltage. For the clock tree problem, the use of a driver to drive a long interconnect wire without repeater drivers would result in unacceptable delay and transition times. In this paper, we present a reduced-swing clock scheme with drivers, intermediate drivers and receivers, as illustrated in Figure 3. The reduced swing driver as shown in Figure 3 is presented in [8] and its output swings from  $V_{tn}$  to  $V_{dd} - |V_{tp}|$ . Therefore, an intermediate reduced swing buffer cannot be an inverter since the magnitude of its input signal would result in a significant short circuit current. We present a reduced swing buffer consisting of 4 transistors. M4 and M5 act as an inverter, and M3 and M6 effectively alter the supply and ground voltages to  $V_{dd} - |V_{tp}|$  and  $V_{tn}$ , respectively, thereby ensuring a zero steady-state short circuit current and keeping the output swing the same as the input swing.

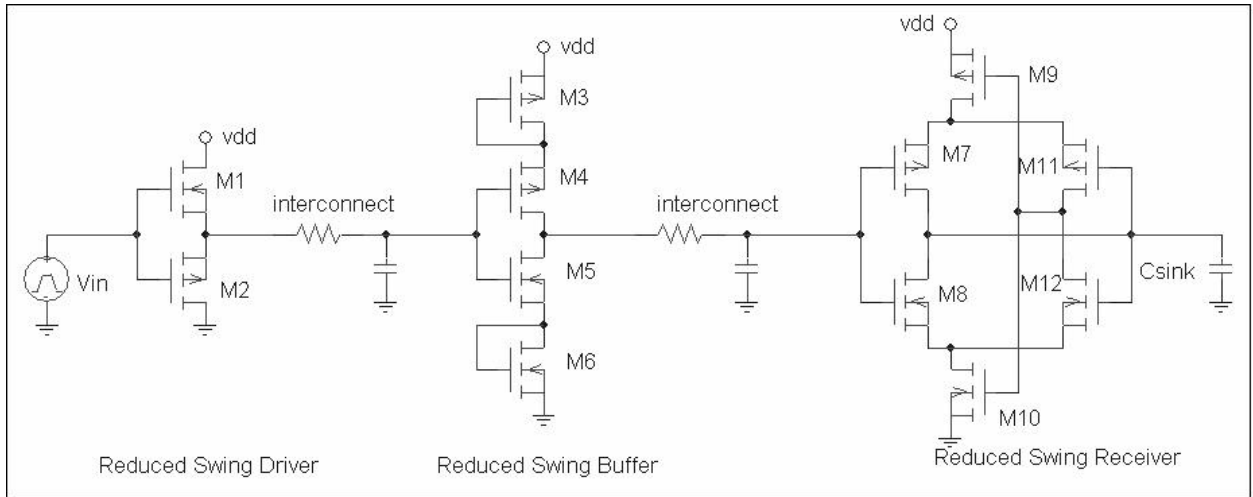


Figure 3: Reduced Swing Clock Scheme

Finally, the reduced swing receiver is a modification version of the fully complementary Self-Biased CMOS differential amplifier presented by Bazes [1]. The modification is performed by feeding back the output signal, which is the inversion of one of the inputs, to the other differential input node. Therefore, the clock signal only swings from  $V_{tn}$  to  $V_{dd} - |V_{tp}|$ .

A few comments about the circuits above are in order:

1. Whenever an active resistor is used to maintain a restricted output swing, for example, in the reduced swing buffer, the transistors will have to be sized appropriately to reduce the effect of two transistors in series.
2. The presence of transistors M3 [M6] in the reduced swing buffer serves the purpose of limiting the output swing by turning off a path to  $V_{dd}$  [ground] when the output voltage reaches

$V_{dd} - |V_{tp}| [V_{tn}]$ . As a result, if the voltage on the wire connected to this rises above  $V_{dd} - |V_{tp}|$  due to coupling with another wire, it has no discharge path to reduce the voltage back to  $V_{dd} - |V_{tp}|$ . This could cause the fall delay to increase from the nominal value and therefore bring about unexpected skews. Consequently, it is important to exercise extra care during place and route to minimize noise coupling.

3. If the clock is stopped for an extensive period of time as a power-saving measure, leakage current could cause the outputs of the reduced swing buffer to rise towards  $V_{dd}$  or fall towards ground, depending on the polarity of the output state. This could cause unexpected skews in the first clock cycle after the clock is reactivated. This may be overcome by using a design discipline that starts the clock one or more cycles prior to when it is required.
4. The delays of these circuits are susceptible to power supply noise (as in the case of other clocking structures used today [21]), and noise effects similar to those described in item 3 above. We will present results on the effect of noise on the delays of our circuits, in comparison with noise effects on CMOS inverters, which are used in traditional circuits, in our Section 6.

### 3. BUFFER INSERTION

Traditionally, buffer insertion has been performed in a post-routing phase. In this paper we will modify the algorithm in [5] to insert buffers in the bottom up phase of the DME algorithm. After each pair of subtrees is merged, we consider the possibility of inserting a buffer at the root of the two child subtrees. The criterion for inserting a buffer is that the slew rate at each buffer input and each sink node is faster than a given specification. This not only limits the effect of input slope on the delay, but also controls the short-circuit power and also provides a sharp clock edge at the utilization points at the sink of the clock tree.

As shown in Figure 1, we will assume that the lowest buffer stage in the clock tree is an LHconverter stage, and our first task is to find the locations of these type of buffers. Let us consider the total power dissipated by the clock tree to consist of two components:

- (i)  $P_b$ , the power dissipated by buffers in the tree, and
- (ii)  $P_w$ , the power dissipated by the wires in the clock tree.

The location of the LHconverters has a major impact on the power dissipation of the clock tree for two reasons:

- (a) Since the LHconverters are at the lowest stage of the clock tree, they are more numerous than any other type of buffer. Moreover, they contain more transistors per buffer than any other buffer used in the clock tree (see Figure 3). Therefore, moving the LHconverters downstream towards the sinks results in an increased value of  $P_b$ .
- (b) The wires downstream of the LHconverters are driven at the high swing, and therefore consume a larger amount of power per unit wire length. Consequently, it is advantageous to move the LHconverters as far downstream as possible to reduce the value of  $P_w$ .

From these points, it is clear that the positioning of the LHconverters plays an important role in the total power dissipation of the tree. In light of point (a) above, we focus on minimizing  $P_b$  by minimizing the total power dissipation of the LHconverters and assume that the power dissipation of buffers at other levels will change by only a small amount on moving the LHconverters.

This is supported by the fact that other buffers are further up the clock tree and are much less numerous than the LHconverters, and for practical purposes, can be assumed to consume a constant amount of power over all locations of the LHconverters. In practice, we found this assumption to be valid.

### 3.1 Theoretical Results on Buffer Positioning

We now present results on criteria used to determine the positions of the LHconverters, using a common area measure that estimates it as the sum of buffer widths.

**Theorem 1:** *For a minimum buffer area solution, LHconverters must be inserted at the clock sinks, appropriately sized to meet the clock slew rate (transition time) constraints.*

**Proof:** Let  $T_1$  and  $T_2$  be two subtrees that are zero-skew merged to form a larger subtree, as shown in Figure 4. Let us consider the possibility of inserting an LHconverter in this subtree, and we consider two options for this purpose:

- (a) An LHconverter of size  $w_1$  drives the subtree that is formed by merging  $T_1$  and  $T_2$
- (b)  $T_1$  and  $T_2$  are driven by LHconverters of size  $w_2$  and  $w_3$ , respectively

In each case, the LHconverter sizes are chosen so that edge rate requirements for each subtree cannot be met by using the same LHconverter size further up the tree (towards the root) across a merging point. This is an essential condition since its violation would imply that another optimal area solution exists with smaller, corresponding to moving the LHconverter up the tree across a merging point to use one minimum-sized buffer instead of two.

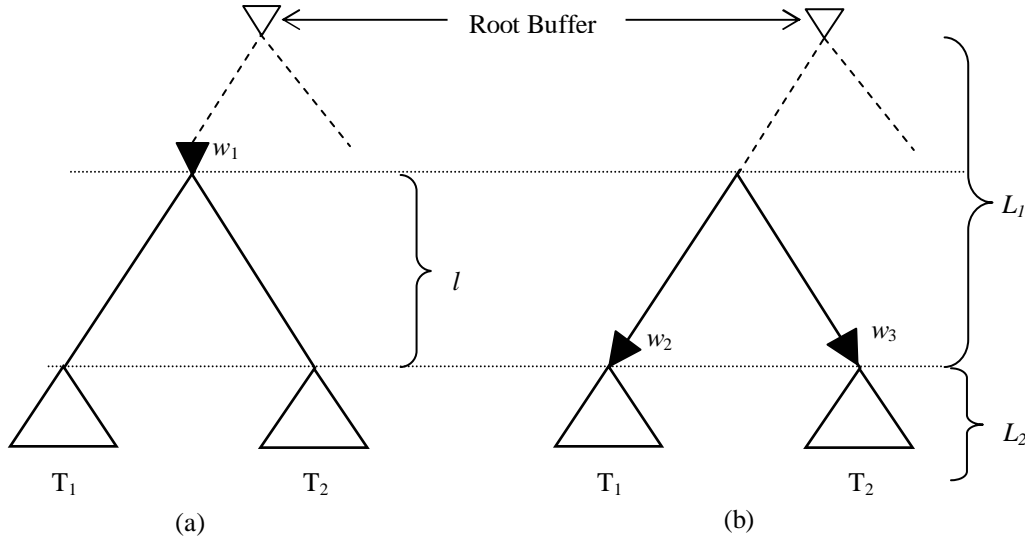


Figure 4: Buffer Placement

Assume that both options correspond to the same total wire length for the clock tree; this is an approximation, but the total wire length is observed not to change significantly when an LHconverter at a low level of the clock tree is moved upwards. In order for option (b) to be an area-optimal solution,  $w_1$  must be greater than  $w_2 + w_3$ . We will overload the notation by using the name " $w_j$ " to refer to the LHconverter of size  $w_j$ .

We further denote the total wire length in the low voltage and high voltage regions as  $L_1$  and  $L_2$ , respectively, and  $l$  as the wire length corresponding to the two wires between the buffer  $w_1$  and the buffers  $w_2$  and  $w_3$ , as illustrated in Figure 4.

We will now proceed to show that while option (a) is better in term of the number of buffers used, option (b) is better in terms of the total buffer area. Starting from option (b), if the buffers  $w_2$  and  $w_3$  are to slide up the tree towards the root node, we will show that it will be essential to increase their size. Therefore, as long as the buffer lies on a given wire segment, its smallest area must correspond to a position at the lower end of the tree, away from the root.

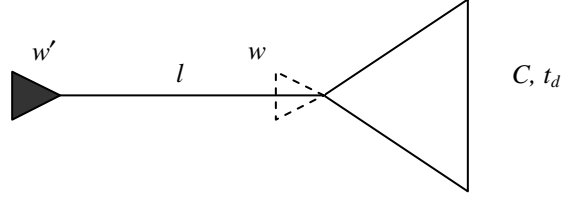


Figure 5: Buffer Sliding

Figure 5 shows a buffer driving a subtree, which is characterized by the downstream capacitance,  $C$ , and the downstream delay,  $t_d$ . If we want to slide the buffer up towards the parent node, it must be sized up from  $w$  to  $w'$  in order to satisfy the transition time constraint. The relationship between  $w$  and  $w'$  can then be expressed as:

$$\frac{t_{rise}}{2} = \frac{k}{w'}(C + lC_o) + r_0 l \left( C + \frac{lC_o}{2} \right) + t_d = \frac{k}{w}C + t_d \quad (2)$$

where  $C$  and  $t_d$  are, respectively, the delay and capacitance downstream of the location of buffer  $w$ ,  $l$  is the length of the segment along which  $w$  is moved up to  $w'$ , and  $k$ ,  $r_0$  and  $C_o$  are, respectively, the unit buffer resistance, the unit wire resistance and the unit wire capacitance. This leads to the relation

$$\frac{1}{w'} \left( 1 + \frac{lC_o}{C} \right) + \frac{r_0 l}{k} \left( 1 + \frac{lC_o}{2C} \right) = \frac{1}{w} \quad (3)$$

Since the multiplicative factor for  $1/w'$  is larger than 1 and the resulting quantity is added to a positive number, this implies that  $1/w > 1/w'$ , i.e.,  $w' > w$ .

Therefore, when both buffers in Figure 4(b) are made to slide up until they are just downstream of the merging point, as shown in Figure 6(b) the size  $w_2' > w_2$  and  $w_3' > w_3$ . We will now study the result of moving the two buffers  $w_2'$  and  $w_3'$  across the merging point to  $w_1$  while maintaining the adherence to the delay specification, as shown in Figure 6(a).

Note that  $l_1$  and  $l_2$  are the same for both Figure 6(a) and 6(b). To satisfy the transition time requirement at the leaf nodes, the following relationship must hold.

$$\frac{k}{w_1} \left( (l_1 + l_2)C_o + C_1 + C_2 \right) + r_0 l_1 \left( \frac{C_o l_1}{2} + C_1 \right) + t_{d_1} = \frac{k}{w_2'} \left( l_1 C_o + C_1 \right) + r_0 l_1 \left( \frac{C_o l_1}{2} + C_1 \right) + t_{d_1} \quad (4)$$



Simplifying this expression, we find that the relationship between  $w_2'$  and  $w_1$  is<sup>2</sup>

$$w_2' = \frac{l_1 C_o + C_1}{((l_1 + l_2) C_o + C_1 + C_2)} w_1 = \frac{C_{left}}{C_{total}} w_1 \quad (5)$$

where  $C_{left}$  is the total down stream capacitance in the left subtree and  $C_{total}$  is the sum of the down stream capacitance in both subtrees. Similarly, defining  $C_{right}$  as the total capacitance in the right subtree, the expression for  $w_3'$  can be derived as

$$w_3' = \frac{l_2 C_o + C_2}{((l_1 + l_2) C_o + C_1 + C_2)} w_1 = \frac{C_{right}}{C_{total}} w_1 \quad (6)$$

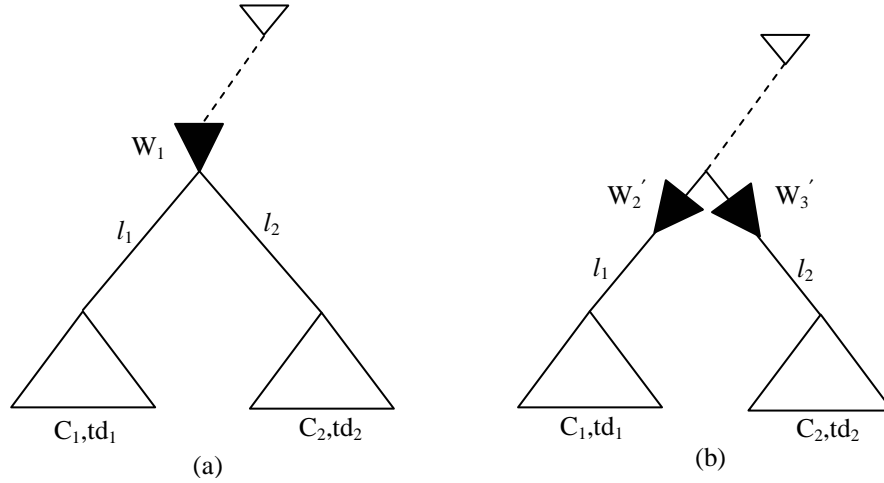


Figure 6: Buffer Merging

Therefore, adding (5) and (6), we obtain the result

$$w_1 = w_2' + w_3' \quad (7)$$

Since  $w_2'$  and  $w_3'$  are greater than  $w_2$  and  $w_3$  respectively, and since  $w_1 = w_2' + w_3'$ , the scenario in Figure 4 (a) is worse than that in Figure 4(b) in term of buffer area. From this, we may conclude that positioning an LHconverter lower in the tree will lead to a smaller area cost as long as the LHconverter is always proportionally sized so as to just meet the transition time constraints at the sinks.

However, once the LHconverter transistors are down to their minimum size, it will be impossible to further reduce its area by pushing it further down the tree. In fact, further moves across a branching point would require one minimum-sized LHconverter to be replicated as two minimum-sized LHconverters, and would therefore cause an increase in area. However, in practice, the use of minimum-sized buffers anywhere in a clock tree will not satisfy the transition time constraints at the sinks. Therefore, it will be essential to add LHconverters at the sinks, appropriately sized to meet the desired slew rates. This concludes the proof of the theorem.

<sup>2</sup> A result similar to Equations (5) and (6), referred to as buffer splitting, was independently proved by Zhou *et al.* in [18].

Next, we consider the two scenarios shown in Figures 4(a) and 4(b) in terms of the power dissipation. As before, we will assume that each of  $w_1$ ,  $w_2$  and  $w_3$  exactly meets the transition time constraints at the sinks. In order to prove this, we introduce a power dissipation model for an LHconverter in the following form:

$$P = k_1 w + k_2 \quad (8)$$

where the size of the buffer is characterized by the buffer size  $w$ , and  $k_1$  and  $k_2$  are fixed constants<sup>3</sup>. For a multistage circuit such as a LHconverter, we have a multistage circuit (see, for example, Figure 2) that must be sized. We assume that the output stage is sized in such a way that the ratio of the NMOS transistor to the PMOS transistor is constant; all internal stages are sized so that their width increases are proportional to  $w$ . In ratioed buffers, the linear dependence of power on the buffer size  $w$  is a reasonable approximation since the size of the output stage changes by a much larger amount by the size of any input stage. Alternatively, if only the output stage of the buffer is sized and the input stages are left untouched, the relation is exact.

Therefore, for the scenario of Figure 4, in order for option (a) to be better than option (b), the following condition must hold:

$$k_1 w_1 + k_2 + k_3(L_1 - l) + k_4(L_2 + l) < k_1 w_2 + k_2 + k_1 w_3 + k_2 + k_3 L_1 + k_4 L_2 \quad (9)$$

where  $k_3 = fV_{ddL}^2C_o$  and  $k_4 = fV_{ddH}^2C_o$ .

**Theorem 2:** Let  $P_1$  be the power in the clock tree corresponding a specific positioning of the LHconverters, sized to meet the transition time constraints at the sinks. Let  $P_2$  be the power corresponding to LHconverters being inserted at any higher location in the tree, appropriately sized so that the transition time constraints are satisfied. The power dissipation  $P_1 < P_2$  provided the following condition holds:

$$k_2 > k_1(w_1 - (w_2 + w_3)) + l(k_4 - k_3) \quad (10)$$

**Proof:** Follows immediately from a simplification of (9).

The inequality in (10) states that  $k_2$  must be greater than the sum of the power dissipation due to the size increase of the buffers, which is the first term on the right hand side, and the power savings due to the low power region of wire segment  $l$ , which is the second term.

In order for (10) to be true,  $k_2 > k_1(w_1 - (w_2 + w_3))$  and  $k_2 > l(k_4 - k_3)$ , since both terms of the right hand side are positive (from the relationship between  $w_1$ ,  $w_2$  and  $w_3$  shown in the proof of Theorem 1, and from the definition of  $k_3$  and  $k_4$ ). The latter condition can be expressed as

$$l < k_2 / (k_4 - k_3)$$

Consider the level converter in Figure 2, and consider  $k_2$  to correspond to the power dissipation of all transistors but the two that constitute the output stage, assuming them (temporarily) to be at minimum size. Using the NTRS roadmap [12] and the device parameters in [4], we calculate the maximum allowable value of  $l$  for various technologies in Table 1 in order for (10) to be true. The table also shows the average value of the minimum pairwise distance between sinks (calculated using an approach similar to Edahiro [5]) and the minimal distance of the merging nodes up to the first buffer level of  $r_1$ - $r_5$  benchmarks of [14].

---

<sup>3</sup> We point out briefly here that if a similar model for area is available, the machinery developed here may be extended for that area model.

Table 1: Required and Average Wirelength

Technology(nm)	250	180	150	130	100
$V_{ddH}/V_{ddL}$ (Volts)	2.5/1.8	1.8/1.5	1.5/1.2	1.5/1.2	1.2/0.9
Maximum allowable $l$ ( $\mu\text{m}$ )	26.5	9.7	2.8	7.5	4.9
Benchmarks	$r_1$	$r_2$	$r_3$	$r_4$	$r_5$
Average $l$ between sinks( $\mu\text{m}$ )	293	274	248	205	188
Average $l$ up to first buffer( $\mu\text{m}$ )	463	357	366	315	293

Noting that the value of  $l$  increases linearly with  $k_2$ , we observe that even if we permit  $k_2$  to correspond to realistic non-minimum sizes of the transistors in the non-output stages of the LHconverter, it will be impossible to approach the average  $l$  values in the benchmarks. Moreover, criterion (10) actually requires  $k_2$  to be even larger than that corresponding to the values of  $l$  in the table in order to overcome the contribution of the first term on the right hand side,  $k_1(w_1 - (w_2 + w_3))$ . Therefore, for these benchmark circuits, the use of LHconverters at points other than the leaf nodes, appropriately sized to meet transition time constraints, is likely to almost always result in a higher power dissipation than the use of LHconverters at the leaf nodes. The only situation where moving an LHconverter upstream towards the root will be advantageous will be in the case where two sinks are very close to each other. The algorithm we will propose in Section 5 will allow for that possibility<sup>4</sup>.

The figures in Tables 1 correspond to the multiple  $V_{dd}$  case. It was observed that the numbers corresponding to the use of a single  $V_{dd}$  with a reduced swing voltage were also similar.

## 4. PROPOSED ALGORITHM

### 4.1 Outline of the Algorithm

The clock tree construction procedure is similar to previous work in many ways, and is outlined here. The primary difference is in the use of an HLconverter at the root of the tree and LHconverters at various points in the clock tree.

An outline of the algorithm is illustrated in Figure 7. The algorithm maintains two sets, **A** and **B**. **A** is a set of non-buffered nodes and is initialized to a set of sinks, while **B** is a set of buffered merging segments and is initialized to an empty set. The merging segments correspond to the roots of subtrees, and a buffered merging segment is one that has buffers placed at the root of its two child nodes.

<sup>4</sup> We find, in fact, in our results that the LHconverters are almost always inserted at the leaf nodes for  $r_1$ - $r_5$ .

**Algorithm** Bottom-up Buffer Insertion**Input:** set of sinks  $S$ , technology parameters**Output:** Tree of buffered merging segments

```

BEGIN
   $\mathbf{A} = S$  /*  $\mathbf{A}$  is a set of non-buffered segments */
   $\mathbf{B} = \Phi$  /*  $\mathbf{B}$  is a set of buffered segments */
  while (  $|\mathbf{A}| > 1$  or  $|\mathbf{B}| > 0$  )
    if (  $|\mathbf{B}| > 0$  and (  $|\mathbf{A}| = 0$  )
       $\mathbf{A} = \mathbf{B}$ ;  $\mathbf{B} = \Phi$  /*if  $\mathbf{A}$  is empty and  $\mathbf{B}$  is non-empty, swap them*/
       $\mathbf{G}(E,V) = DT(\mathbf{A})$ ; /* Build Delaunay Triangulation on  $\mathbf{A}$  */
       $\mathbf{I} = \text{Find\_independent\_edges}(\mathbf{G})$ ;
        /*Find an independent edge set*/
      for (  $i=0; i < |\mathbf{I}|; i++$  )
         $a = \text{Zero\_merge}(b,c)$ ; /*for each edge in  $\mathbf{I}$ , Zero merge
                                   the two end segments*/
         $\mathbf{A} = \mathbf{A} - \{b,c\}$  /* delete merging segment  $b,c$  from  $\mathbf{A}$ */
        if (buffer insertion criterion satisfied)
          /*if the transition time exceeds the limit, i.e.,
               $R_d * a.C + a.td > t_{\text{rise}}/2$  for a buffer, or
              insertion criterion for an LHconverter */
           $b = \text{insert\_buffer}(b)$ ;
           $c = \text{insert\_buffer}(c)$ ;
           $a = \text{Zero\_merge}(b,c)$ ;
           $\mathbf{B} = \mathbf{B} \cup \{a\}$ 
        else
           $\mathbf{A} = \mathbf{A} \cup \{a\}$ 
    END

```

Figure 7: Algorithm Hierarchical Clustering-Based Buffer Insertion

A zero skew clock tree is constructed using the method described in [5]. It first builds a Delaunay Triangulation on  $\mathbf{A}$ , and then constructs a nearest neighbor graph. A set of independent edges is identified by selecting a subset of edges from the nearest neighbor graph. For each of these independent edges, a zero-skew merge is performed for the merging segments that are connected by the selected independent edge. The two merged segments will be deleted from  $\mathbf{A}$  and the new merging segment will be checked to see if it satisfies the transition time (slew rate) constraint. This may be verified by hypothetically inserting a buffer at the root of the new subtree and checking for transition time violations. If such a violation exists, then buffers are inserted at the root of the two child nodes and the two child nodes are zero-skew merged again. Moreover, if buffers are inserted at the child nodes of a given node, then this node will be added to a buffered segment set  $\mathbf{B}$ ; if not, it is added to  $\mathbf{A}$ . The procedure continues until the set  $\mathbf{A}$  is empty, which occurs when all nodes at this level are buffered.

Once we have added the first level of clock buffers, we swap the sets  $\mathbf{A}$  and  $\mathbf{B}$ , and repeat the whole procedure a gain. The algorithm proceeds until there are only one node left in  $\mathbf{A}$

and  $\mathbf{B}$  is empty. At this point, the procedure returns a tree of segments. By construction, we may then state the following result.

***Property:** Algorithm Bottom-up Buffer Insertion ensures that for any path from the root to the sinks, there will be an equal number of buffers.*

## 4.2 Finding a Minimum Power Solution

The addition of the first level of LHconverters can be performed as a special case. The minimum area solution is found by applying the above procedure directly, using LHconverters at the first level. The minimum power solution is found by applying a dynamic programming procedure, based on the separable property of the clock tree power computation.

For each merging point, we store a tuple  $[\text{Sol}_{\text{buffered}}, \text{Sol}_{\text{unbuffered}}]$ . The parameters  $S_{\text{buffered}}$  and  $S_{\text{unbuffered}}$  correspond, respectively, to the best solution so far for the situation where an LHconverter has, or has not, been added to the downstream subtree.

Each solution is parameterized by the total power of the downstream subtree and the location of the merging segment for the root of the subtree. When two subtrees are merged, the two  $S_{\text{unbuffered}}$  solutions are combined to create an  $S_{\text{unbuffered}}$  solution at the current level. An  $S_{\text{buffered}}$  solution may be created either by combining the two  $S_{\text{buffered}}$  solutions from the subtrees, or by combining the two  $S_{\text{unbuffered}}$  solutions and placing an LHconverter at the merging point, sized in order to meet the transition time requirements at the leaf nodes. This continues up the tree until the required buffer size is larger than the maximum size, and at this point the optimal solution is chosen. For the benchmarks r1-r5, it was almost always found that the optimal position of the set of buffers was at the leaf nodes; more detailed results are provided in Section 5.

## 5. EXPERIMENTAL RESULTS

Our algorithm was tested on the five benchmarks from [14]. The parameters that we used are based on a 250 nm technology as specified in [4,12], and are listed in Table 2.  $R_0$  is the unit driver resistance; all other parameters are as described earlier.

Table 2: 250 nm Technology Parameters

Parameters	Values
$C_o$	53.1 aF
$R_o$	0.293 $\Omega/\mu\text{m}$
$R_d$	17.1 K $\Omega$
$C_b$	170 aF
$f$	500 MHz
$V_{\text{ddH}}-V_{\text{ddL}}$	2.5-1.8V

A comparison of the results of our algorithm with algorithm CL from [5] showed that the wire length used in the two is similar. Table 3 shows the comparison of power dissipation between algorithm CL, augmented with a buffer insertion algorithm, and our algorithm. The total power values shown in the table are the sum of the wire power  $P_w$  and the buffer power  $P_b$  in the clock network. We use different values for the resistance of a LLconverter and an LHcon-

verter, corresponding to the fact that the resistance depends on the  $V_{dd}$  value used. According to the delay metrics used, the clock skew is zero by construction. Due to the unavailability of public-domain transistor models for a 0.25 $\mu$ m technology, we were unable to verify the skews using SPICE.

The figures in Table 3 are based on a 500MHz clock with the transition times accounting for no more than 10% of the clock period. The percentage power savings relative to CL are shown in parentheses. The results of our power minimization algorithm using  $V_{ddH}=2.5V$  and  $V_{ddL}=1.8V$  are shown next under “Dual  $V_{dd}$ .” Finally, we show the results of applying our power minimization algorithm under  $V_{ddH}=2.5V$  only, but using a lower swing voltage,  $V_s$  that varies from  $V_{tn}$  to  $V_{ddH} - |V_{tp}|$ ; we assume  $V_{tn} = |V_{tp}| = 0.2 V_{dd}$ . These results are presented under the column marked “Low Swing.” Although our clock tree construction does not force LHconverters to be placed at the sink nodes, and places them at a power-optimal position (see Section 5), the optimal position for most of the LHconverters is at the sink nodes, as predicted by our discussion of Table 1. The few cases where the buffers are moved higher up correspond to situations where the sinks are close to each other, and the savings afforded by reducing the number of buffers offsets the overhead of driving the wires at a higher  $V_{dd}$  value. In particular, for the benchmarks r1 through r5, the number of buffers that are moved one level up from the sinks are 2, 10, 14, 28 and 56, respectively; no buffers are moved more than one level up from the sinks for any of these benchmarks.

Table 3: Power Dissipation of the Clock Trees

Benchmark	CL (mW)	Dual $V_{dd}$ (mW)	Low Swing (mW)
r1	26.93	13.34(50.5%)	16.31(39.4%)
r2	49.63	27.67(44.2%)	33.43(32.6%)
r3	62.19	35.11(43.5%)	43.80(29.6%)
r4	130.57	71.77(45.0%)	90.44(30.7%)
r5	183.02	105.67(42.3%)	134.62(26.4%)

From Table 3, the power saved when using multiple supply voltages and reduced swing buffers are an average of 45% and 31%, respectively. An upper bound on the possible power savings is determined by

$$\Delta P_{max} = 1 - (V_{ddL}^2/V_{ddH}^2) = 52\%$$

for the “Dual  $V_{dd}$ ” case. For the “Low Swing” case, the value of  $\Delta P_{max}$  can be calculated to be 40%. We observe that the use of dual  $V_{dd}$  provides values that are closer to these ideal numbers. This is because a reduced swing repeater consists of 4 transistors and its resistance,  $R_d$ , is twice that of an inverter of the same size. Therefore, it requires more repeater drivers to be inserted into the clock tree or a greater degree of sizing to maintain the specifications of the clock tree, causing an additional overhead of power dissipation. Even so, it is observed that in all cases, significant savings are achieved by each method.

## 6. EFFECTS OF SUPPLY NETWORK NOISE ON THE SINGLE $V_{DD}$ SCHEME

In a low voltage environment, the effects of noise in the supply network could be significant. Even for conventional static CMOS buffers, it is well known that variations on the supply volt-

age lines due to transient currents through voltage drop and ground bounce effects can lead to delay variations [21]. A similar effect is seen for the circuits used in this paper, and we quantify these effects here. Our experimental results have shown that the structures in Section 2.2.1 have a similar sensitivity to supply voltage variations as the conventional approach to buffering that uses static CMOS inverters and one supply voltage level; therefore, we do not specifically address those structures in this section.

However, for the circuit structures shown in Section 2.2.2 that operate under a single  $V_{dd}$  but under a reduced voltage swing, supply voltage noise can be a significant issue. The specific effects that influence the behavior of these circuits are similar to those listed in items 3. and 4. at the end of Section 2.2.2, and we will particularly focus on the latter here. In particular, we show that the gain from using a single supply is somewhat offset by the fact that the supply network must be designed more carefully to limit voltage drop effects.

A comparison, through SPICE simulations, of the normalized delay variations for a static CMOS inverter and for the reduced swing CMOS buffer of Figure 3 is shown in Table 4.

Table 4: Delay variations under supply voltage variations

Supply voltage variation	Static CMOS Inverter	Reduced swing Buffer	Reduced swing receiver
+10%	21%	36%	30%
+5%	12%	21%	18%
-5%	14%	31%	25%
-10%	32%	80%	63%

From the results in this table, we see that the reduced swing circuitry is significantly more sensitive to supply voltage variations, and that careful supply network design is of paramount importance. In particular, at these low voltage swing levels, reduced supply voltages of 10% can significantly affect the delay and therefore possibly degrade the skew. Therefore, the use of such a scheme practically implies that supply voltage variations should be held to within plus or minus 5%.

A second experiment tested the effect of the jitter at the output of the reduced swing receiver due to power supply level variations on the jitter at the output of the reduced swing buffer. In one case, the supply voltage level for the reduced swing buffer was varied, but that for the receiver was kept pure; in the second case, both were varied and kept at identical levels. The results are shown in Table 5.

Table 5: Supply voltage noise transmission from the reduced swing buffer to the reduced swing receiver

Supply voltage variation	At buffer only	At buffer and at receiver
+10%	33%	25%
+5%	19%	14%
-5%	28%	19%
-10%	76%	46%

Several observations may be made from this table. Firstly, the delay variations at the buffer output are slightly diminished as they are passed through another stage if the supply voltage at the receiver is assumed to be clean. If the same supply voltage is used for both, significant cancellations are seen and the delay variations are reduced, so that they become more reasonable<sup>5</sup>. This also provides an indication that the supply network should be designed: it should be built so that the resistance between the supply voltage nodes for the buffer and the subsequent receiver is as small as possible, to ensure similar  $V_{dd}$  values at both modules. This may be achieved by selecting the supply net topology so that there is a direct, or nearly direct, wire connection between the supply node for consecutive buffers, or a buffer and its receiver.

## 7. CONCLUSION

We have presented an analysis of the problem of clock tree routing at different voltages for distribution and utilization. Our implementation guarantees that number of buffers along any path from root to sinks is equal, and uses a low voltage to distribute the clock signal before reconverting it to a high voltage at the utilization points. We have applied our algorithm to the low power clock schemes: one scheme uses reduced-swing buffers, while the other uses multiple supply voltages. The experimental results show that the low power clock schemes using our algorithm provide significant savings in the total power dissipation.

The ideal power reduction is given by  $[1 - (V_{ddL}^2/V_{ddH}^2)]$  using two voltages, and  $[1 - (V_{swing}/V_{dd})]$  using a single  $V_{dd}$  with a reduced swing,  $V_{swing}$ . As device technologies scale down, technology constraints such as the maximum electrical field sustainable by the thin oxide will dictate the maximum  $V_{dd}$  that can be used. The minimum  $V_{ddLOW}$  and the value of  $V_{swing}$  are both dependent of the threshold voltage - these do not scale quite as fast as  $V_{dd}$  in future technologies - and on noise considerations, which will limit the value of  $V_{ddLOW}$ . Therefore, it is expected that the marginal benefits of using two supply voltages will reduce with time using the circuits described in this paper. In the medium term, we believe that this approach has the potential to provide significant benefits. However, it may be possible to invoke techniques using subthreshold logic to maintain the gains of an approach that distributes the clock signal at a different value from its value at the utilization points.

In closing, it is appropriate to remark on the effects of such a procedure on today's design methodologies. The procedure for constructing zero-skew clock trees under Elmore delays is an extension of techniques that are used today, based largely on [14]. It is expected that better noise avoidance techniques will be required under low voltages, and techniques for buffer insertion such as [20] and noise-aware routing tools can be used to ensure this. The work in this paper presents two possibilities: either using two independent supply voltages, or using a single supply voltage and a level converter to lower the clock distribution voltage. We do not consider the issues related to routing multiple  $V_{dd}$ 's in this work, although several other researchers have addressed and are addressing the problem. Depending on the discipline used, further constraints may be introduced on the locations of clock buffers: for example, under the model in [9], where each row in the standard cell-based design uses the same  $V_{dd}$ , buffer locations cannot be arbitrary. We expect that the framework presented here can be extended in the presence of

---

<sup>5</sup> It must be pointed out, though, that a similar experiment with cascaded CMOS inverters also shows the delay variations to be reduced in similar proportions.



such constraints since typical zero-skew algorithms such as [14] can be extended to accommodate restrictions on buffer locations.

## References

- [1] M. Bazes. "Two Novel Fully Complementary Self-Biased CMOS Differential Amplifiers," *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 2, February 1991.
- [2] K. D. Boese and A. B. Kahng. "Zero-Skew Clock Routing Trees With Minimum Wire length," *Proceedings of the IEEE International Conference on ASIC*, pp. 1.1.1-1.1.5, 1992.
- [3] T. H. Chao, Y. C. Hsu, and J. M. Ho. "Zero Skew Clock Net Routing," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 518-523, 1992.
- [4] J. Cong. "Challenges and Opportunities for Design Innovations in Nanometer Technologies," *SRC Design Sciences Concept Paper*, Dec. 1997.
- [5] M. Edahiro. "A Clustering-Based Optimization Algorithm in Zero Skew Routing," *Proceedings of the ACM/IEEE Design Automations Conference*, pp. 612-616, June 1993.
- [6] M. Edahiro. "An Efficient Zero-Skew Routing Algorithm," *Proceedings of the ACM/IEEE Design Automations Conference*, pp. 375-380, June 1994.
- [7] L. P. P. van Ginneken. "Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 865-868, 1990.
- [8] H. I. Hanafi, R. H. Dennard, and C. L. Chen. "Design and Characterization of a CMOS Off-Chip Driver/Receiver with Reduced Power-Supply Disturbance," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 5, May 1992.
- [9] M. Igarashi, K. Usami, K. Nogami, F. Minami, Y. Kawasaki, T. Aoki, M. Takano, C. Misuno, T. Ishikawa, M. Kanazawa, S. Sonoda, M. Ichida, and N. Hatanaka. "A Low-Power Design Method Using Multiple Supply Voltages," *Proceedings of the ACM International Symposium on Low Power Design*, pp. 36-41, 1997.
- [10] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh. "Clock Routing for High Performance ICs," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 573-579, 1990.
- [11] A. B. Kahng, J. Cong, and G. Robins. "High-Performance Clock Routing Based on Recursive Geometric Matching," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 322-327, 1991.
- [12] Semiconductor Industry Association, "National Technology Roadmap for Semiconductors," 1997.
- [13] G. E. Tellez and M. Sarrafzadeh. "Clock Period Constrained Minimal Buffer Insertion in Clock Trees," *Proceedings of the ACM/IEEE International Conference on Computer-Aided Design*, pp. 219-223, 1994.

- [14] R. S. Tsay. "Exact Zero Skew Clock Routing Algorithm," *IEEE Transactions on Computer-Aided Design*, 12(2):242-249, February 1993.
- [15] C. W. Tsao. "VLSI Clock Net Routing," *Ph. D. Thesis*, University of California Los Angeles, 1996.
- [16] K. Usami and M. Horowitz. "Clustered Voltage Scaling Technique for Low-Power Design," *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 3-8, 1995.
- [17] A. Vittal and M. Marek-Sadowska. "Power Optimal Buffered Clock Tree Design," *Proceedings of the ACM/IEEE Design Automation Conference*, June 1995.
- [18] X. Zeng, D. Zhou and W. Li, "Buffer Insertion for Clock Delay and Skew Minimization," *Proceedings of the ACM International Symposium on Physical Design*, pp. 36-41, 1999.
- [19] H. Zhang and J. Rabaey. "Low-Swing Interconnect Interface Circuits," *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 161-166, 1998.
- [20] C. J. Alpert, A. Devgan and S. T. Quay, "Buffer insertion for Noise and Delay Optimization," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 362-367, 1998.
- [21] R. Saleh, S. Z. Hussain, S. Rochel and D. Overhauser, "Clock Skew Verification in the Presence of IR-Drop in the Power Distribution Network," *IEEE Transactions on Computer-Aided Design*, 19(6), pp. 635-644, June 2000.