

Clock Distribution Using Multiple Voltages *

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 the performance. This paper presents theory and algorithms for building a low power clock tree. Two low power schemes are used: a reduced swing scheme and one using multiple supply voltages. We analyze the issue of tree construction and present conclusions relevant to various technology generations according to the National Technology Roadmap of Semiconductors (NTRS). Our experimental results show that the power could be saved an average of 45% for a 0.25 μm technology using multiple supply voltages, and 31% using reduced swing buffers.

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 [1–9] 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.

In this work, we present new theory and results for building low power clock trees using multiple supply voltages or a reduced swing clock signal before converting the clock signal back to the original voltage at the utilization points. This clock scheme is presented in [10] and is shown in Figure 1. The work in [10] implicitly assumes that placing converters at sinks is the best configuration, without any justifications. The main contribution of this work is to prove empirically that the configuration in [10] indeed is close to the optimal configuration.

The organization of this paper is as follows. In Section 2, we review previous work in the area and place our work in this context. Section 3 states the problem and discusses some possible buffer schemes. Next, in Section 4, we derive a set of theoretical results and make empirical observations based on technology trends as predicted by the NTRS [11]. This is followed by an outline of

the clock network construction algorithm in Section 5. Finally, we present our experimental results in Section 6, followed by a conclusion.

2 PREVIOUS WORK

The area of clock tree synthesis for zero skew has been widely researched. After early techniques [5, 6] that rely on equalizing the path length to each sink were found to be inadequate for RC interconnects, a method that explicitly equalizes the delay to each sink of the tree were proposed by Tsay [8]. 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.

Tsay's algorithm only suggested an algorithm to zero-merge two subtrees, but did not try to minimize total wire length. The Deferred-Merged Embedding (DME) method, which was discovered independently by three groups [1–3], 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 ensuring zero skew, it is also imperative to ensure that a sharp slew rate for the clock edge. This requires the insertion of 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, 12, 13].

The large amount of capacitance that is driven by the clock net, in conjunction with the fact that the clock net switches on every transition, leads to a major fraction of the total chip power being dissipated in clock distribution. The total power dissipation consists of two components: static and dynamic power dissipation.

In this work, we consider the static power dissipation due to leakage current 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 , C_L , V_{dd} , V_s are respectively the clock frequency, the load capacitance, the supply voltage, and the output swing of the buffer. If the output of the buffer swings from gnd to V_{dd} , then $V_s = V_{dd}$ and the formula reduces to $P = fC_L V_{dd}^2$.

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

*This research was supported in part by the Semiconductor Research Corporation under contract 98-DJ-609, by the National Science Foundation under award CCR-9800992, and by a scholarship from the Royal Thai Air Force.

- (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 [1–3, 9].
- (b) reducing V_{dd} , which creates a quadratic reduction if V_s is also simultaneously reduced by the same factor (for example, when $V_{dd} = kV_s$ 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 net minimization is 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.

3 STATEMENT OF THE PROBLEM

3.1 Structure of the Clock Tree

A clock scheme with multiple supply voltages was proposed in [10] and is illustrated in Figure 1 below. In this figure, an HLconverter is a buffer that converts the incoming clock signal to the chip from high voltage swing to a lower voltage swing. The clock signal is then transmitted on the chip as a low voltage signal, thereby ensuring a low power clock distribution network. Intermediate buffers (LLconverters) 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 LLconverters, as shown in Figure 1, are used as repeaters. These repeaters can be as simple as inverters, or they may be more complex, depending on the clock scheme. Later we will show that we may not use an inverter for this type of repeater. At the points of use at the sink flip-flops, the clock signal is converted using the LHconverter block to the higher voltage swing, which is the voltage at which it is used by the logic network.

The work in [10] implicitly assumes that in order to save the maximum power, the low power region must be maximized, thus minimizing the high power 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 power region.

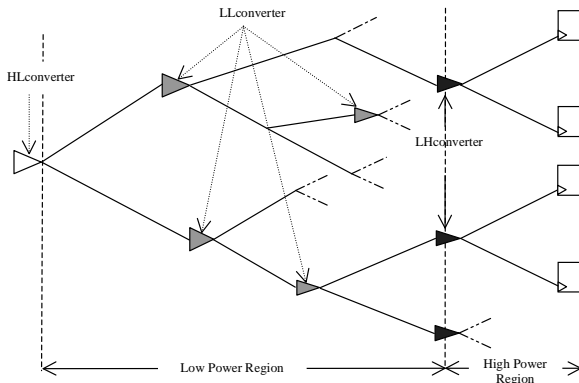


Figure 1: Low Power Clock Scheme

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 LHconverter 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, 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 [1–4, 8, 9]. The slew transition time at a node is taken to be twice the Elmore delay from the previous buffer to that node.

3.2 Level Converter Circuits

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. A conventional repeater can also be used as a LLconverter for this clock scheme. The structure of an LHconverter is more involved, and the design we use is the same as that utilized in earlier work (for example, [10, 14]) and is illustrated in Figure 2.

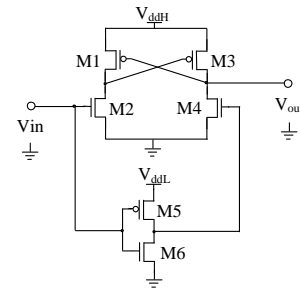


Figure 2: A LHconverter Circuit

From (1), another variable that we can reduce is the output clock swing, V_s . Zhang and Rabaey [15] show that reducing voltages swing of the signal on the wire is a good approach for achieving improved energy efficiency. The work in [15], 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 lower 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, illustrated in Figure 3. The reduced-swing driver as shown in Figure 3 is presented in [16] 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.

Finally, the reduced swing receiver is a modification version of the fully complementary self-biased CMOS differential amplifier presented by Bazes [17]. 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 $V_{dd} - |V_{tp}|$.

4 BUFFER INSERTION

From Figure 1, an HLconverter is greedily inserted at the root of the clock tree. Therefore, there are two type of buffers whose locations

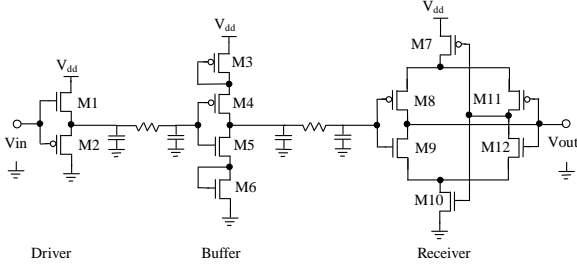


Figure 3: Reduced Swing Clock Scheme

are to be determined: LLconverters and LHconverters. An algorithm for LLconverter insertion can be similar to a general buffer insertion algorithm since their positions do not affect power dissipation significantly. Traditionally, buffer insertion has been performed in a post-routing phase [7, 12]. Vittal *et al.* [13] have shown that simultaneous topology design and buffer insertion can do significantly better in terms of area, rise times and power dissipation. In this paper, we will exploit algorithms in both [13] and [3]. The modification is done by performing buffer insertion 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 edge rate at each buffer input and each sink node are within a given specification. This not only limits the effect of input slope on the delay, but also controls the short-circuit power and 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 the LHconverter stage, and we want 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 valid.

4.1 Theoretical Results on Buffer Positioning

We now present results on criteria used to determine the positions of the LHconverters.

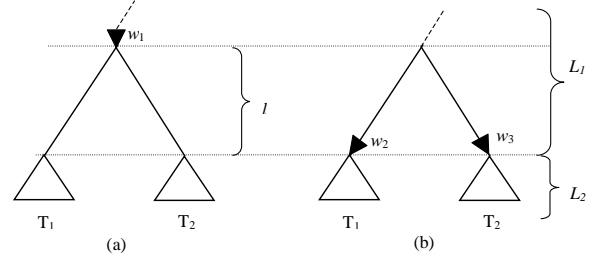


Figure 4: Buffer Placement

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: Total buffer area is the summation of all the area of buffers (HLconverter, LLconverters, and LHconverters). However, as stated earlier, we assume that total buffer area of an HLconverter and LLconverters is constant. Therefore, we only consider the buffer area of LHconverters. 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 the two illustrated 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_1 and w_2 , 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.

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.

Consider a buffer driving a subtree as shown by Figure 4(b) and Figure 5(b). 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:

$$\frac{t_{rise}}{2} = \frac{k}{w'}(C + lC_0) + r_0l(C + \frac{lC_0}{2}) + t_d = \frac{k}{w}C + t_d \quad (2)$$

where C and t_d are, respectively, the delay and capacitance downstream of the location of buffer, w and l is the length of the segment

along which w is moved up to w' , and k , r_0 and C_0 are, respectively, the unit buffer resistance, the unit wire resistance and the unit wire capacitance. This leads to the relation

$$\frac{1}{w'}(1 + \frac{lC_0}{C}) + \frac{r_0l}{k}(1 + \frac{lC_0}{2C}) = \frac{1}{w} \quad (3)$$

Since the multiplicative factor for $1/w'$ is larger than unity 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 5(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 5(a).

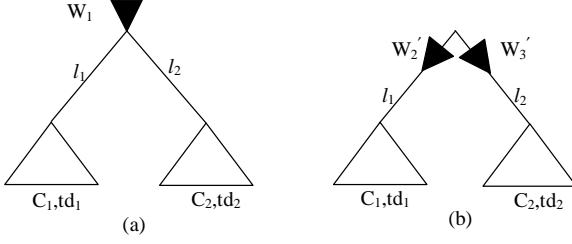


Figure 5: Buffer Merging

Note that l_1 and l_2 are the same for both Figure 5(a) and 5(b). To satisfy the transition time requirement at the leaf nodes, the following relationship must hold.

$$\frac{k}{w_1}((l_1 + l_2)C_0 + C_1 + C_2) + r_0l_1(\frac{C_0l_1}{2} + C_1) + t_{d1} = \frac{k}{w'_2}(l_1C_0 + C_1) + r_0l_1(\frac{l_1C_0}{2} + C_1) + t_{d1} \quad (4)$$

Simplifying this expression, we find that the relationship between w'_2 and w_1 is

$$w'_2 = \frac{l_1C_0 + C_1}{((l_1 + l_2)C_0 + C_1 + C_2)}w_1 = \frac{C_{left}}{C_{total}}w_1, \quad (5)$$

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

$$w'_3 = \frac{l_2C_0 + C_2}{((l_1 + l_2)C_0 + C_1 + C_2)}w_1 = \frac{C_{right}}{C_{total}}w_1 \quad (6)$$

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.

Next, we consider the two scenarios shown in Figures 4(a) and 4(b) in term 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:

$$P = k_1w + k_2, \quad (8)$$

where the size of the buffer is characterized by the buffer size w , and k_1 and k_2 are fixed constants. For an 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:

$$k_1w_1 + k_2 + k_3(L_1 - l) + k_4(L_2 + l) < k_1w_2 + k_2 + k_1w_3 + k_2 + k_3L_1 + k_4L_2 \quad (9)$$

where $k_3 = fV_{ddL}^2C_0$ and $k_4 = fV_{ddH}^2C_0$.

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 2, 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 [11] and the device parameters in [18], 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 pair-wise distance between sinks (calculated using an approach similar to EdaHiro [3]) and the minimal distance of the merging nodes up to the first buffer level of benchmarks r_1 through r_5 [8].

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

Table 1: Required and Average Wire Length

| | | | | | |
|-------------------------------|-------|-------|-------|-------|-------|
| Technology (nm) | 250 | 180 | 150 | 130 | 100 |
| Max allowable l (μm) | 26.5 | 9.7 | 2.8 | 7.5 | 4.9 |
| Benchmarks | r_1 | r_2 | r_3 | r_4 | r_5 |
| Avg. l sinks (μm) | 293 | 274 | 248 | 205 | 188 |
| Avg. l 1^{st} (μm) | 463 | 357 | 366 | 315 | 293 |

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. We find, in fact, in our results that the LHconverters are always inserted at the sink nodes for r_1 , and there is a very few cases where the LHconverters are not inserted at sinks for $r_2 - r_5$. 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 reduced swing buffers were also similar.

5 PROPOSED 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 6. The algorithm maintains two sets: \mathbf{A} and \mathbf{B} . \mathbf{A} is a set of non-buffered nodes and is initialized to a set of sinks, while \mathbf{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.

Algorithm: Bottom-up Buffer Insertion

Input: set of sinks \mathbf{S} , technology parameters

Output: Tree of buffered merging segments

BEGIN

```

 $\mathbf{A} = \mathbf{S}; \mathbf{B} = \Phi$ 
while ( $|\mathbf{A}| > 1$  or  $|\mathbf{B}| > 0$ )
  if ( $|\mathbf{B}| > 0$ ) and ( $|\mathbf{A}| = 0$ )
     $\mathbf{A} = \mathbf{B}; \mathbf{B} = \Phi$ 
   $\mathbf{G}(\mathbf{E}, \mathbf{V}) = \mathbf{DT}(\mathbf{A})$ 
   $\mathbf{I} = \text{Find\_Independent\_Edges}(\mathbf{G})$ 
  for ( $i = 0; i < |\mathbf{I}|; i++$ )
     $a = \text{Zero\_merge}(b, c);$ 
     $\mathbf{A} = \mathbf{A} - \{b, c\}$ 
    if (buffer insertion criterion satisfied)
       $b = \text{Insert\_buffer}(b);$ 
       $c = \text{Insert\_buffer}(c);$ 
       $a = \text{Zero\_merge}(b, c);$ 
       $\mathbf{B} = \mathbf{B} \cup \{a\}$ 
    else
       $\mathbf{A} = \mathbf{A} \cup \{a\}$ 

```

END

Figure 6: Algorithm Clustering-Based Buffer Insertion

A zero skew clock tree is constructed using the method described in [3]. First, it builds Delaunay Triangulation on \mathbf{A} , and

builds a nearest neighbor graph. A set of independent edges is then constructed by selecting edges from the nearest neighbor graph. For each of the 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 still satisfies the transition time constraint. We check if the node still satisfies the transition time by pretending insert a buffer at the root of the new subtree. Then, it checks for transition time violation. If the transition time exceed the limit, the buffers are inserted at the root of the two child nodes. Then, the two child nodes are zero merged again. If buffers are inserted at its child nodes, 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. This leads to the following result.

Theorem 3 *Algorithm Bottom-up Buffer Insertion ensures that for any path from root to sinks there will be an equal number of buffers.*

The addition of the first level of LHconverters can be performed as a special initialization 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 $[Sol_B, Sol_{UB}]$. The parameters Sol_B and Sol_{UB} 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 Sol_{UB} solutions are combined to create an Sol_{UB} solution at the current level. An Sol_B solution may be created either by combining the two Sol_B solutions from the subtrees, or by combining the two Sol_{UB} 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 $r_1 - r_5$, it was almost always found that the optimal position of the set of buffers was at the leaf nodes. There are very few cases (less than 1%) that the converters are moved up to the higher nodes.

6 EXPERIMENTAL RESULTS

Our algorithm was tested on the five benchmarks from [8], and the results are shown in Table 2. The parameters that we used are based on a 250 nm technology as specified in [11, 18]. The total power values shown on the table are the sum of P_b and P_w of the clock tree. The values of P_w and P_b are calculated using 1, 8, respectively. Power dissipated by the sinks is constant and is ignored. The buffer resistance, R_d , is calculated as follows. For the multiple supply voltages scheme, the R_{dL} of a LLconverter is computed by $R_{dH} * V_{dH} / V_{dL}$, where R_{dH} is a driver resistance under V_{dH} , and the R_d for LHconverter is the same as an inverter. For the reduced signal swing scheme, the R_d of a LLconverter and LHconverter are twice as much as that of an inverter of the same size. Since our algorithm is a modification version of algorithm DME, the skew of the clock tree is zero. The “wire snaking” technique is used, if necessary, for finding a tapping point. Due to unavailability of transistor models for 0.25 μm technology, we were unable to measure the actual clock skew.

Table 2: Power Dissipation of Clock Trees

| BM | CL (mW) | Dual V_{dd} (mW) | Low Swing (mW) |
|-------|------------|-----------------------|-------------------|
| r_1 | 26.93 | 13.34(50.5%) | 16.31(39.4%) |
| r_2 | 49.63 | 27.67(44.2%) | 33.43(32.6%) |
| r_3 | 62.19 | 35.11(43.5%) | 43.80(29.6%) |
| r_4 | 130.57 | 71.77(45.0%) | 90.44(30.7%) |
| r_5 | 183.02 | 105.67(42.3%) | 134.62(26.4%) |

A comparison of the results of our algorithm with algorithm CL from [3] showed that the wire length used in the two is similar. Table 2 shows the comparison of power dissipation between algorithm CL, augmented with a buffer insertion algorithm, and our algorithm are shown under the other two columns. The results are based on a 500 MHz clock rate 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_{in} to $V_{ddH} - |V_{tp}|$; we assume $V_{tn} = |V_{tp}| = 0.2V_{dd}$. These results are presented under the column marked “Low Swing.” Note that number of LHconverters used are similar to number of sinks. However, there are very few cases where LHconverters are moved to higher levels. For example, there are 3, 10, 14, 28, and 56 LHconverters inserted at one level from sinks for benchmarks r_1 to r_5 , respectively. No LHconverter is inserted at any higher level for all benchmarks.

From Table 2, the power is 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 $\Delta P_{max} = 1 - \left(\frac{V_{ddL}}{V_{ddH}}\right)^2 = 52\%$ for the “Dual V_{dd} ” case. For the “Low Swing” case, the value of ΔP_{max} can be calculated to be 40%. From Table 2 we observe that when using multiple supply voltages, the results are closer to the ideal values. This is because a reduced swing repeater consists of 4 transistors as shown in Figure 3 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 in order to maintain the specified clock rate. These added buffers increase the total power dissipation. However, it is observed that in all cases, significant savings are achieved by each method.

6.1 CONCLUSION

We have presented an analysis of the problem of clock tree routing at different voltages for distribution and utilization, for various technology generations according to the NTRS. 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 apply our algorithm to the low power clock schemes: one scheme uses reduced-swing buffer, while the other user multiple supply voltages. The experimental results show that the low power clock schemes using our algorithm provide significant savings in the total power dissipation.

References

[1] K. D. Boese and A. B. Kahng, “Zero-skew clock routing trees with minimum wire length,” in *Proceedings of the IEEE International Conference on ASIC*, pp. 1.1.1–1.1.5, 1992.

[2] T. H. Chao, Y. C. Hsu, and J. M. Ho, “Zero skew clock net routing,” in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 518–523, 1992.

[3] M. Edahiro, “A clustering-based optimization algorithm in zero-skew routing,” in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 612–616, June 1993.

[4] M. Edahiro, “An efficient zero-skew routing algorithm,” in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 375–380, June 1994.

[5] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, “Clock routing for high performance ICs,” in *Proceedings of the ACM/IEEE Design Automation Conference*, 1990.

[6] A. B. Kahng, J. Cong, and G. Robins, “High-performance clock routing based on recursive geometric matching,” in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 322–327, 1991.

[7] G. E. Tellez and M. Sarrafzadeh, “Clock period constrained minimal buffer insertion in clock trees,” in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 219–223, 1994.

[8] R. S. Tsay, “Exact zero skew clock net routing,” *IEEE Transactions on Computer-Aided Design*, vol. 12, pp. 242–249, Feb. 1993.

[9] C. W. Tsao, *VLSI Clock Net Routing*. PhD thesis, University of California Los Angeles, 1996.

[10] 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,” in *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 36–41, 1997.

[11] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*. 1997.

[12] L. P. P. van Ginneken, “Buffer placement in distributed RC-tree networks for minimal Elmore delay,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 865–868, 1990.

[13] A. Vittal and M. Marek-Sadowska, “Power optimal buffered clock tree design,” in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 497–502, June 1995.

[14] K. Usami and M. Horowitz, “Clustered voltage scaling technique for low-power design,” in *Proceedings of the ACM International Symposium on Low Power Design*, pp. 3–8, 1995.

[15] H. Zhang and J. Rabaey, “Low-swing interconnect interface circuits,” in *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 161–166, 1998.

[16] 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, pp. 783–790, May 1992.

[17] M. Bazes, “Two novel fully complementary self-biased CMOS differential amplifiers,” *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 165–168, Feb. 1991.

[18] J. Cong, “Challenges and opportunities for design innovations in nanometer technologies,” tech. rep., Semiconductor Research Corporation, Research Triangle Park, NC, 1997.