Delay Minimization for Zero-Skew Routing

Masato Edahiro [1]

Research Report CS-TR-415-93
March 1993

Abstract

Delay minimization methods are proposed for zero-skew routings. A delay-time estimation formula is derived, which can be used as an objective function to be minimized in zero-skew routing algorithms. Moreover, the optimum wire width is formulated. Experimental results show that our methods with a clustering-based algorithm achieve 50% reduction of the delay time on benchmark data with 3000 pins.

[1]Department of Computer Science, Princeton University and C&C Systems Research Laboratories, NEC Corporation

1 Introduction

With the increase of the clock rate in VLSI, the clock-net routing scheme plays more critical roles. In order to make the clock rate higher, at least two factors should be taken into account in clock-net routing. First, since the clock skew affects the clock period directly, exact zero skew is desired. Next, the delay time should be minimized in a clock net. Consider an example of single-phase clocking in Fig. 1. In CMOS design, the delay time is dominated by the rise/fall time [19]. Thus, longer delay time in clocking causes longer rise/fall time as shown in Fig. 1 (b). Obviously, longer rise/fall time makes switching time longer in synchronous elements, and therefore, the clock cycle time must be longer in order to assure the time period for processing in combinational logics between the synchronous elements. Consequently, the delay time in a clock net must be minimized in order to make the clock rate faster in reliable VLSI circuits.

Also, minimizing the total wire length in routings is useful to lower power dissipation by minimizing the total load capacitance. Even though shorter total wire length tends to cause shorter delay time, it is possible that the minimum total-wire-length routing is not always the minimum delay routing as we will show in this paper.

While the exact zero skew was theoretically accomplished using a bottom-up routing algorithm [20], the delay minimization is still a difficult problem. In the zero-skew routing algorithm, the delay time and the total wire length in resultant routings strongly depend on the processing order in terminals. Thus, in order to minimize the delay time or the total wire length, the optimum ordering algorithm needs to be devised. However, this optimization problem seems quite difficult, and no optimum polynomial-time algorithm has been proposed. There have been many heuristic algorithms proposed for the total-wire-length minimization in zero-skew routings [4, 5, 6, 7, 9, 10, 11, 15, 16]. Especially, a clustering-based algorithm in [10, 11] is the best known for the minimization. Since minimizing the total wire length tends to minimize the delay time, this algorithm turns out the best known for the delay minimization as well.

Another difficulty for the delay time minimization is to estimate the delay time accurately in zero-skew routings. It is obvious that we cannot minimize the delay time unless the delay time is accurately

(a) Short Delay (b) Long Delay

Figure 1: Single Phase Clocking.

estimated. Although the delay on wires has been thoroughly analyzed [2, 3, 8, 13, 17, 18, 21], it seems very difficult to estimate the total delay time for general RC networks with a driver. The only estimation is the sum of the switching delay of the driver and the propagation delay of the networks. The switching delay is calculated from the total load capacitance with process-dependent constants [19], while the propagation delay is accumulation of path delays [20]. Since these two delays should interact in the total delay, just a sum of two delays is inaccurate.

In this paper, first, with characteristics of zero-skew routings, we derive a simple but more accurate formula for the delay time estimation in zero-skew routings, which turns out an objective function to be minimized for the delay time optimization. We show that, if a routing has exact zero skew, each node in the routing should obey a simple circuit equation. By solving this circuit equation, we will obtain the delay time estimation for the entire routing, which includes the interaction between the switching delay and the propagation delay, though it is still a linear function of the total load capacitance and the propagation delay.

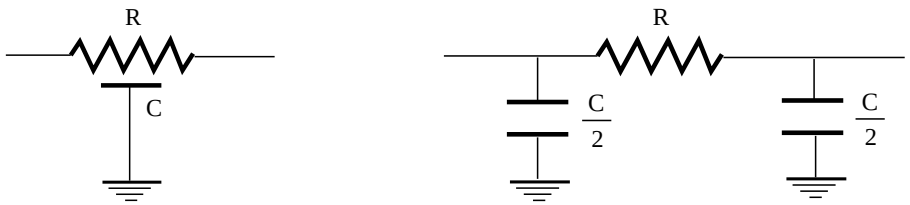
Next, we propose an approximation for the optimum wire width to minimize the delay time in zero-skew routings, assuming that the total delay time is a linear function of the total load capacitance and the propagation delay as we have estimated above. It is easy to see that large width of wires increases wire capacitance, while small width increases wire resistance. Thus, there should be an optimum wire width to minimize the delay time. In fact, we will show that our estimated delay time has the global minimum as a function of wire widths. Although there have been several researches on the wiresizing problem [3, 8], this is the first optimum wire width formulation for zero-skew routings. We show that our technique is greatly effective to reduce the delay time in experimental results.

Experimental results show that our delay minimization technique included in a clustering-based algorithm [10, 11] achieves 10%-50% reduction of the total delay time compared with the best-known algorithm [10, 11] on benchmark data [14, 20].

2 Zero-Skew Routing

2.1 Delay Model for Wires

In this paper, we use a distributed RC model approximated by the lumped π2-model for wires [3] (Fig. 2). We consider that wire delays are well-approximated by the distributed RC model because inductive effects are negligible for wires inside VLSI in current CMOS design [3]. In addition, π2-model is accurate enough to approximate the distributed RC model for such circuit analysis as shown in this paper [18].



(a) Distributed RC Model (b) $\pi 2$ Model

Figure 2: Delay Model for Wires.

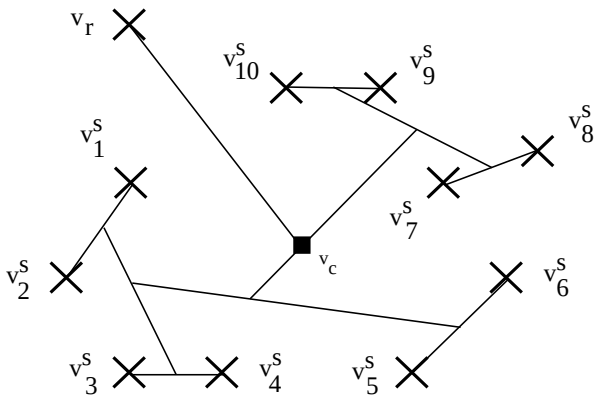


Figure 3: Clock Tree with a root v_r and leaves $S = \{v_1^s, v_2^s, \ldots, v_{10}^s\}$.

2.2 Zero-Skew Routing

Given a fan-out terminal v_r and a set of n fan-in terminals $S = \{v_1^s, v_2^s, \ldots, v_n^s\}$, a *clock tree* is defined by a tree rooted by v_r whose n leaves are S (Fig. 3). We call the fan-out terminal *root* and fan-in terminals *leaves*. A set of leaves in the subtree rooted by a node v is called *leaves connecting to* v and denoted by S_v. In this paper, clock trees are always binary, though nodes may degenerate and they do not look binary in some cases.

We assume that the *load capacitance* $C(v_i^s)$ is given for each leaf v_i^s, which is usually the gate capacitance of transistors connecting with the leaf. Also, the *load capacitance* $C(v)$ for an internal node v is defined by the total capacitance in S_v that includes wire capacitance as well as gate capacitance.

Then, a *zero-skew routing* for the given root and leaves is defined by a clock tree in which all the propagation delay time from the root to all leaves is equal. From this definition, it is clear that, for any node in the zero-skew routing, all the propagation delay from the node v to leaves in S_v should

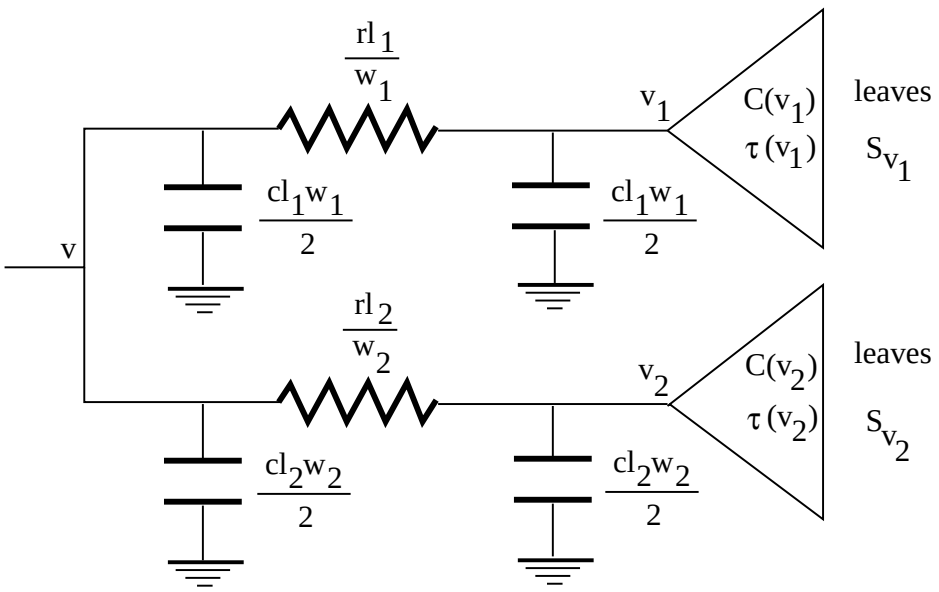


Figure 4: π2-Model for Zero-Skew Merge at v.

be equal. We call this delay the *propagation delay time* $\tau(v)$ *for* v. For leaves v_i^s, $\tau(v_i^s) = 0$[2].

2.3 Zero-Skew Merge

Now, we can derive equations for $\tau(v)$ and $C(v)$ to be satisfied at each internal node v on a zero-skew routing. Let v_1 and v_2 be children of v on the clock tree, and l_1 and l_2 be lengths of wires from v to v_1 and v_2, respectively. Also, let r and c be wire resistance and capacitance for an unit length wire. Then, by using π2-model for the wiring delay [3, 18], in a zero-skew routing, $\tau(v)$ and $C(v)$ should satisfy the following equations [20]:

$$\begin{aligned} \tau(v) &= rl_1\left(\frac{cl_1}{2} + C(v_1)\right) + \tau(v_1) = rl_2\left(\frac{cl_2}{2} + C(v_2)\right) + \tau(v_2), \\ C(v) &= C(v_1) + C(v_2) + c(l_1 + l_2). \end{aligned}$$

In this paper, we assume that wire widths are variable. For this model (Fig. 4), the above formulae are rewritten by

[2]This is easily generalized in hierarchical clock design where all $\tau(v_i^s)$ has an equal value. We discuss in remarks that the case in which $\tau(v_i^s)$'s have distinct values is impractical.

$$\tau(v) = \frac{rl_1}{w_1}\left(\frac{cl_1w_1}{2} + C(v_1)\right) + \tau(v_1) = \frac{rl_2}{w_2}\left(\frac{cl_2w_2}{2} + C(v_2)\right) + \tau(v_2), \quad (1)$$

$$C(v) = C(v_1) + C(v_2) + c(l_1w_1 + l_2w_2), \quad (2)$$

where w_1 and w_2 are widths of wires from v to v_1 and v_2. In this case, r and c are wire resistance and capacitance for an unit length and width wire.

It is important to note that, given two zero-skew routings rooted by v_1, v_2 and wire widths w_1, w_2, using the equations (1) and (2), the location of v can be calculated in constant time such that v is the parent of v_1 and v_2, the tree rooted by v is a zero-skew routing, and $l_1 + l_2$ is minimized [4, 6, 12]. This operation to determine the location of v is called the *zero-skew merge*.

2.4 Routing Algorithm

A zero-skew routing for given a root and leaves can be constructed in a bottom-up fashion. Let K be a node set initialized by the leaf set S. At each iteration of the algorithm, two nodes v_1 and v_2 are taken from K, the location of v is calculated using a zero-skew merge, and v is put in K. After $n-1$ iteration, a zero-skew routing is constructed. Details of this algorithm are described in [4, 6, 9, 16, 20].

Although this algorithm always guarantees zero skew regardless of the selection of v_1 and v_2 from K [4, 6, 16, 20], the delay time and the total wire length highly depend on the selection strategy.

In this paper, we use two types of selection strategy. One is called the *length-minimum selection [LM]*, which selects a pair of nodes in K so as not to increase wire lengths. This type of selection strategy is used in the best-known algorithm for the total-wire-length minimization such as [10, 11].

The other algorithm is called the *delay-minimum selection [DM]*, which tries to minimize the total delay time. As we will propose in the next section, the total delay time is estimated by a linear function of the load capacitance $C(v_r)$ and the propagation delay $\tau(v_r)$ at the root v_r. The algorithm [DM] selects v_1 and v_2 so as to minimize the function value for $C(v)$ and $\tau(v)$, where v is the node calculated by a zero-skew merge between v_1 and v_2. It is clear that this selection method tends to minimize the total delay.

3 Delay Time Estimation

Suppose that a pre-constructed zero-skew routing, in which all $C(\cdot)$'s and $\tau(\cdot)$'s have been calculated for the root v_r and internal nodes using the equations (1) and (2), is driven by a clock driver whose process parameters are known. *The (total) delay time for a leaf* v is defined by time difference between the input transition (50% level) of the clock driver and the 50% level at the leaf v. Note

that, in a zero-skew routing, the delay time for all leaves should be equal. Thus, we simply call it the *(total) delay time.*

In this section, we propose a delay-time estimation formula for zero-skew routings. The main result in this section is that the delay time t_d is estimated by

$$t_d \approx 1.85\frac{C(v_r)}{\beta V_{DD}} + 0.7\tau(v_r),$$

where β is the MOS transistor gain factor of the clock driver.

In order to estimate the delay time, we first construct a circuit equation for fall/rise time at leaves in a zero-skew routing, and then solve the differential equation. In CMOS circuits, since the output fall/rise time dominates the delay time, the delay time can be estimated by half of the fall/rise time [19].

In this paper, we discuss only the fall time. The rise time can be estimated in a similar way.

3.1 Fall Time Estimation

First, we show a circuit equation at each internal node v, which is expressed by the load capacitance $C(v)$ and the propagation delay time $\tau(v)$.

Theorem 1 *For each node v and leaves $v_i^s \in S_v$ in a zero-skew routing,*

$$V(v) = V(v_i^s) + \tau(v)\frac{dV(v_i^s)}{dt}, \tag{3}$$

$$I(v) = C(v)\frac{dV(v_i^s)}{dt}, \tag{4}$$

where $V(v)$ and $I(v)$ are voltage and current values for the node v.

We prove the theorem in Appendix A.1.

Next, we construct a circuit equation to estimate the fall time at leaves. Suppose that the clock net is driven by a CMOS inverter whose nMOS has the MOS transistor gain factor:

$$\beta_n = \frac{\mu_n \epsilon}{t_{ox}}\left(\frac{W_n}{L_n}\right),$$

where μ_n is the effective surface mobility of the electrons in the channel, ϵ is the permittivity of the gate insulator, t_{ox} is the thickness of the gate insulator, W_n and L_n are the width and length of the channel. Then, the ideal (first order) equations [19] for this nMOS transistor is

$$I_{ds} = \begin{cases} 0 & V_{gs} - V_{t_n} \le 0 & \text{(cut-off)}, \\ \beta_n[(V_{gs} - V_{t_n})V_{ds} - \frac{V_{ds}^2}{2}] & 0 < V_{ds} < V_{gs} - V_{t_n} & \text{(linear)}, \\ \frac{\beta_n}{2}(V_{gs} - V_{t_n})^2 & 0 < V_{gs} - V_{t_n} < V_{ds} & \text{(saturation)}, \end{cases}$$

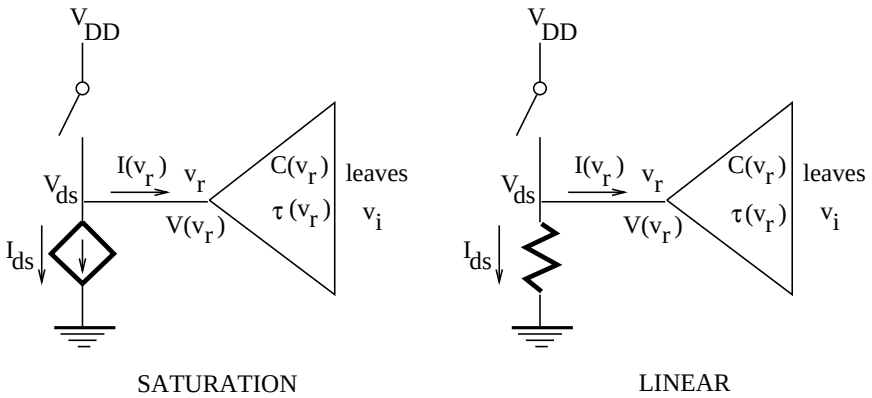


Figure 5: Equivalent Inverter Model for Fall Time Estimation.

where I_{ds} and V_{ds} are the drain-to-source current and voltage, V_{gs} is the gate-to-source voltage, V_{t_n} is the device threshold.

Since the driver is connected to the root v_r of the clock tree, $I_{ds} \equiv -I(v_r)$ and $V_{ds} \equiv V(v_r)$ (Fig. 5). Also, we assume that $V_{gs} = V_{DD}$. Consequently, from Theorem 1, for every leaf $v_i^s \in S$,

$$C(v_r)\frac{dV(v_i^s)}{dt} + \beta_n \left[(V_{DD} - V_{t_n})V(v_r) - \frac{V(v_r)^2}{2}\right] = 0 \quad \text{(linear)}, \tag{5}$$

$$C(v_r)\frac{dV(v_i^s)}{dt} + \frac{\beta_n}{2}(V_{DD} - V_{t_n})^2 = 0 \quad \text{(saturation)}, \tag{6}$$

where

$$V(v_r) = V(v_i^s) + \tau(v_r)\frac{dV(v_i^s)}{dt}. \tag{7}$$

By solving these equations, the fall time at leaves is estimated as follows:

$$t_f \approx 3.7\frac{C(v_r)}{\beta_n V_{DD}} + 1.4\tau(v_r). \tag{8}$$

The formula derivation is shown in Appendix A.2.

3.2 Delay Time Estimation

Since the delay time t_{d_f} for the fall can be estimated by half of the fall time t_f [19],

$$t_{d_f} \approx 1.85\frac{C(v_r)}{\beta_n V_{DD}} + 0.7\tau(v_r). \tag{9}$$

Similarly, the delay time t_{d_r} for the rise is estimated by

$$t_{d_r} \approx 1.85\frac{C(v_r)}{\beta_p V_{DD}} + 0.7\tau(v_r). \quad (10)$$

It is important to compare our estimation with a simple estimation, in which the propagation delay $\tau(v_r)$ is added to half of the fall time for an inverter with a capacitive load $C(v_r)$. By using a fall time estimation for an inverter with a capacitive load in [19], the delay time t'_{d_f} in the simple model is estimated by

$$t'_{d_f} \approx 2\frac{C(v_r)}{\beta_n V_{DD}} + \tau(v_r). \quad (11)$$

The similarity is that both formulae are a linear function of $C(v_r)$ and $\tau(v_r)$.

There are two differences between equations (9) and (11). First, the ratio of coefficients for two terms, $\frac{C(v_r)}{\beta_n V_{DD}}$ and $\tau(v_r)$, is different. In the next section, we show that the wire-width optimization is a function of this ratio. Thus, the difference of the ratio is crucial for the optimization.

Next, the coefficient values in t_{d_f} is smaller than those in t'_{d_f}. As we will see later, simulated values for t_{d_f} is still smaller than the estimated values. Experimental results will show that 60% of estimated values matchs with the circuit simulation SPICE (LEVEL3), in which more precise but complicated circuit equations are used [1].

In the next section, we optimize the wire width assuming that the delay time can be estimated by a linear function of $C(v_r)$ and $\tau(v_r)$.

4 Wire Width Optimization

For a wire segment on a clock net, large wire width may reduce the propagation delay when the wire segment is shared by many leaves, while smaller wire width may make the gate switching faster because of smaller wire capacitance. Since the total delay is estimated by a linear function of the propagation delay and the load capacitance, there should exist an optimum wire width. In this section, we derive an approximation formula for the optimum wire width to minimize the delay time.

4.1 Variable Wire Width Model at Internal Nodes

Consider an internal node v with children v_1 and v_2 in a zero-skew routing. Let l_1 (l_2) and w_1 (w_2) be the length and width of the wire segment from v to v_1 (v_2) (Fig. 6). Suppose that we are constructing the zero-skew routing in a bottom-up fashion. That is, the positions of v_1 and v_2 have already been determined, and $C(v_1)$, $C(v_2)$, $\tau(v_1)$ and $\tau(v_2)$ have been calculated. Thus, these values can be considered constants in the optimization.

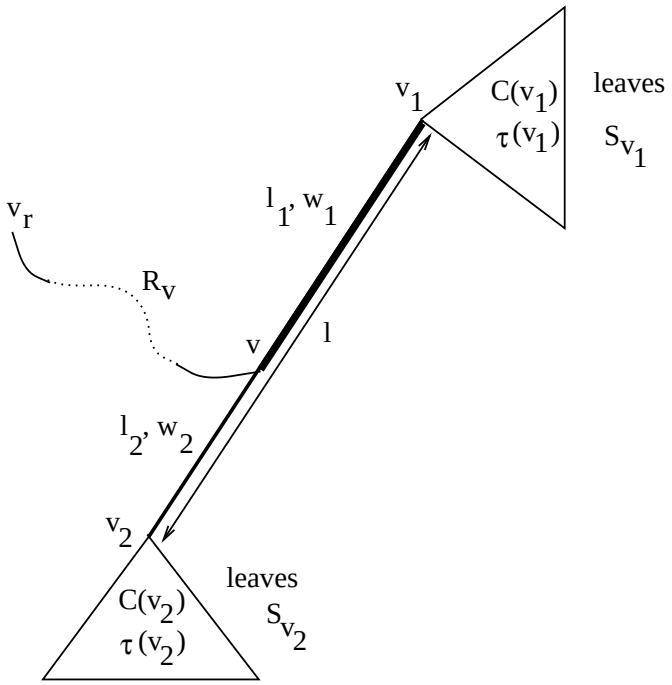


Figure 6: Wire Width Optimization for Zero-Skew Merge at v.

Now, at the next step, the position of v is to be determined. *The wire-width optimization* is to determine the values w_1 and w_2 so as to minimize the delay time. Note that l_1, l_2 and the position of v are expressed as functions of w_1 and w_2 in this optimization.

The length l is defined by the distance between v_1 and v_2, and w_{min} denotes the minimum width of a wire. These values are also constants. In addition, we define the sum R_v of ratio (length/width) over all wire segments p from the root v_r to v, i.e.,

$$R_v = \sum_{p \in \{\text{path from } v_r \text{ to } v\}} \frac{\text{length of } p}{\text{width of } p}. \tag{12}$$

Since we are constructing the routing in a bottom-up fashion, the value R_v can not be calculated. However, in this paper, we assume that R_v can be approximated from the structure of the clock tree and positions of v_1 and v_2, which are independent from w_1 and w_2. We discuss the approximation method in remarks.

Now, suppose that the delay time t_d can be expressed by the formula

$$t_d = \alpha_1 \frac{C(v_r)}{\beta V_{DD}} + \alpha_2 \tau(v_r). \tag{13}$$

In our estimation in equations (9) and (10), $\alpha_1 = 1.85$ and $\alpha_2 = 0.7$. We assume $\beta = \beta_n = \beta_p$ in this section.

Using the equations (1) and (2), the delay time t_d is rewritten in a function of w_1 and w_2 as follows:

$$\begin{aligned} t_d(w_1, w_2) &= \alpha_1 \frac{1}{\beta V_{DD}}(cl_1 w_1 + cl_2 w_2 + a_1) + \\ &\qquad \alpha_2 \left[rR_v(cl_1 w_1 + cl_2 w_2) + a_2 + \frac{rl_1}{w_1}\left(\frac{cl_1 w_1}{2} + C(v_1)\right) + \tau(v_1) \right] \qquad (14) \\ &= \left(\frac{\alpha_1 c}{\beta V_{DD}} + \alpha_2 rcR_v\right)(l_1 w_1 + l_2 w_2) + \alpha_2 r\left(\frac{cl_1^2}{2} + \frac{l_1 C(v_1)}{w_1}\right) + a_3, \qquad (15) \end{aligned}$$

where a_1, a_2 and a_3 are independent from w_1 and w_2.

4.2 Optimum Wire Width at Internal Nodes

In this section, we show the optimum width formulae derived from equation (15). Before deriving the formulae, we show a global optimality theorem for the equation. We use two conditions in the theorem. First, wire widths w_1 and w_2 are not less than the minimum wire width w_{min}. Next, $l_1 + l_2 = l$. Since the sum $l_1 + l_2$ of wire lengths from v to v_1 and v_2 can not be less than the distance l between v_1 and v_2, the optimum solution under this condition is the global optimum if the solution is feasible, i.e., both l_1 and l_2 are non-negative. We will classify into cases based on this feasibility.

Theorem 2 (Wire Width Optimization) *On the domain $w_1, w_2 \geq w_{min}$ and under a condition $l = l_1 + l_2$, the function $t_d(w_1, w_2)$ has the global minimum when*

$$w_1 = w_2 = w^* = \max\left(w_{min}, \sqrt{\frac{\alpha_2 r}{\frac{\alpha_1 c}{\beta V_{DD}} + \alpha_2 r c R_v}\left(\frac{C(v_1)C(v_2)}{C(v_1)+C(v_2)}\right)}\right). \tag{16}$$

The wire lengths for this optimum width are calculated by

$$l_1^* = \frac{(\tau(v_2) - \tau(v_1)) + rl(\frac{C(v_2)}{w^*} + \frac{cl}{2})}{r(cl + \frac{C(v_1)}{w^*} + \frac{C(v_2)}{w^*})}, \tag{17}$$

$$l_2^* = l - l_1^*. \tag{18}$$

The proof is shown in Appendix A.3. It is important to consider this optimum width as a function of the ratio α_2/α_1. This shows that, as we mentioned in the previous section, the accuracy for this ratio is critical for the wire width optimization.

Now, we show the optimum width formulae for zero-skew routings. Since, in the above theorem, we did not care about the condition that l_1 and l_2 should be non-negative value, we need to classify into the following three cases.

Case 1: $[l_1^* > l\ (l_2^* < 0)]$

In this case, we need to fix $l_2 = 0$, i.e., $v \equiv v_2$, for the delay minimization. From equations (17) and (18), for $\exists w_1^+ \leq w^*$,

$$\tau(v_2) - \tau(v_1) = rl(\frac{cl}{2} + \frac{C(v_1)}{w_1^+}).$$

There are two subcases.

Case 1.1: $[w_1^+ < w_{min}]$

In this case, whichever wire width ($\geq w_{min}$) is used, $\tau(v_2) - \tau(v_1)$ is too large to find a position for v on the shortest path between v_1 and v_2. Thus, we need to use a *detour* wire from v_1 to v $(= v_2)$, whose length is larger than l. In order to minimize the delay, w_1 should be w_{min} because the length of the detour wire needs to be minimized. Then, from equation (1), $l_1 = l_1^+$ such that

$$\tau(v_2) = \frac{rc}{2}(l_1^+)^2 + \frac{rC(v_1)}{w_{min}} l_1^+ + \tau(v_1).$$

Case 1.2: $[w_1^+ \geq w_{min}]$

In this case, we do not need any detour edge if $w_1 = w_1^+$ is used. In addition, it is easy to see that this is the case to minimize the delay. Therefore, $l_1 = l$ and $w_1 = w_1^+$.

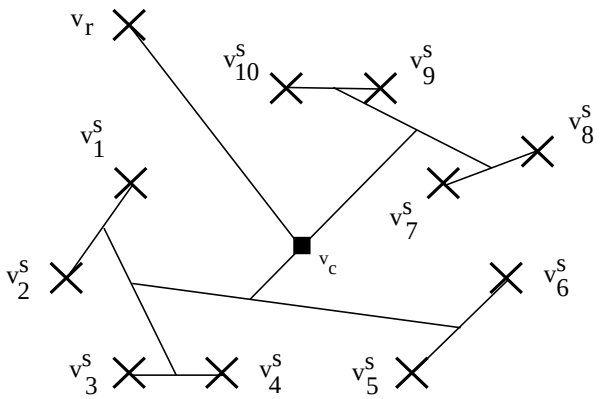


Figure 7: Root v_r and Center v_c in a zero-skew routing.

Case 2: $[0 \leq l_1^* \leq l \text{ and } 0 \leq l_2^* \leq l]$

Since our global optimum specified in equations (16), (17), and (18) is feasible, the optimum wire width and corresponding wire lengths are $w_1 = w_2 = w^*$, $l_1 = l_1^*$, and $l_2 = l_2^*$.

Case 3: $[l_2^* > l \ (l_1^* < 0)]$

This case can be analyzed in a similar way to Case 1. Thus, $l_1 = 0$ and $v \equiv v_1$. Also, there exists a w_2^+ such that

$$\tau(v_1) - \tau(v_2) = rl(\frac{cl}{2} + \frac{C(v_2)}{w_2^+}).$$

Case 3.1: $[w_2^+ < w_{min}]$

$w_2 = w_{min}$ and $l_2 = l_2^+$ such that

$$\tau(v_1) = \frac{rc}{2}(l_2^+)^2 + \frac{rC(v_2)}{w_{min}} l_2^+ + \tau(v_2).$$

Case 3.2: $[w_2^+ \geq w_{min}]$

$l_2 = l$ and $w_2 = w_2^+$.

4.3 Optimum Wire Width at Root

In many cases, the root v_r is not the point at which two children are merged. Instead, the root is connected by a wire to the highest-level merging point v_c on the clock tree (Fig. 7). We call the highest-level merging point the *center*. In this section, we derive the optimum width w_c^* of the wire connecting between v_r and v_c.

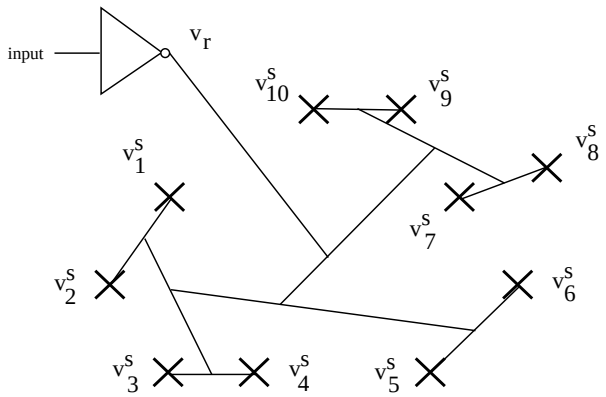


Figure 8: Test Circuit in Experiments. A drive inverter drives all output leaves. The total delay time is time difference between INPUT and leaves, and the propagation delay time is time difference between the root v_r and leaves.

It is easy to see that the length of the wire should be the distance l between v_r and v_c to minimize the delay time. Thus, from the equations (1), (2), and (13),

$$t_d(w_c) = \frac{\alpha_1}{\beta V_{DD}}(clw_c + C(v_c)) + \alpha_2 \left(rl(\frac{cl}{2} + \frac{C(v_c)}{w_c}) + \tau(v_c) \right), \tag{19}$$

where w_c is the width of the wire. Since l is independent from w_c,

$$\frac{dt_d}{dw_c} = \frac{\alpha_1 cl}{\beta V_{DD}} - \alpha_2 rl\frac{C(v_c)}{w_c^2}.$$

By solving $\frac{dt_d}{dw_c} = 0$, the optimum wire width w_c^* is

$$w_c^* = \max\left(w_{min}, \sqrt{\frac{\alpha_2 r \beta V_{DD}}{\alpha_1 c} C(v_c)} \right). \tag{20}$$

Again, we can consider this optimum width as a function of the ratio α_2/α_1.

5 Experimental Results

The proposed methods have been implemented with a clustering-based zero-skew routing algorithm described in section 2.4. The delay time estimation (9) or (10) is utilized for an objective function in the algorithm [DM], and the wire width optimization is applied to each wire segment where $\alpha_2/\alpha_1 = 0.7/1.85$.

The test circuit model in our experiment is depicted in Fig. 8. The total delay time was measured by time difference between INPUT transition (50% level) and the 50% level at leaves, which is estimated by the equations (9) and (10). The clock skew is defined by difference between the maximum and the minimum total delay time. In results generated by our algorithm, the clock skew should be zero theoretically. Also, the propagation delay time was measured by the time difference between the root v_r and leaves, which is estimated by $\tau(v_r)$.

For the drive inverter, we used $\beta_p = 20\left(\frac{W_p}{L_p}\right)[\mu A/V^2]$, $\frac{W_p}{L_p} = 280$, and $\beta_n = 40\left(\frac{W_n}{L_n}\right)[\mu A/V^2]$, $\frac{W_n}{L_n} = 140$ for all experiments. Also, $V_{DD} = 5[V]$.

The data we used are prim1-prim2 [14] and r1-r5 [20].

First, we compared our delay minimization method with a length minimization algorithm [LM]. In this experiment, we tested three algorithms: 1) the length-minimum selection algorithm [LM], 2) the delay-minimimum selection algorithm [DM], 3) the delay-minimum selection algorithm with the wire-width optimization [DM+WW]. Note that the length-minimum selection algorithm [LM] is the best-known algorithm in [10, 11].

The results are shown in Table 1. TWL is the total wire length, EPD is the estimated propagation delay time, and ETD is the estimated total delay time. The propagation delay is estimated by $\tau(v_r)$, and the total delay time is calculated by 60% of t_d in the equations (9) and (10). Table 1 shows that the wire width optimization is greatly effective to reduce the propagation delay time. In addition, it is observed that, for large data (r4 and r5), [DM] obtains smaller total delay time than [LM], while it generates longer total wire length than [LM]. This implies that minimizing the total wire length might not be minimizing the delay time.

Next, we justify our estimation with simulation results by the SPICE circuit simulator. Again, we compared our algorithm with the best-known algorithm [10, 11]. Table 2 shows the estimated and simulated values for the total wire length [TWL], the propagation delay time [PD], the total delay time [TD], and the clock skew [SKW].

Table 2 shows that our estimation fits with the simulation within 10% for most data. In addition, our methods accomplished 10%-50% shorter total delay time than the best-known algorithm.

Table 1: Total Wire Length [TWL], Estimated Propagation Delay Time [EPD], and Estimated Total Delay Time [ETD] for three algorithms [LM] ([10, 11]), [DM], and [DM+WW].

	#pins		[LM]	[DM]	[DM+WW](/[LM])	
prim1	269	TWL	131427	131125	131877	(1.00)
		EPD	2.34ns	2.75ns	0.36ns	(0.15)
		ETD	6.35ns	6.53ns	5.60ns	(0.88)
prim2	603	TWL	306053	315598	317296	(1.04)
		EPD	8.97ns	9.48ns	0.99ns	(0.11)
		ETD	15.90ns	16.13ns	12.77ns	(0.80)
r1	267	TWL	1289004	1288488	1288597	(1.00)
		EPD	1.13ns	1.23ns	0.71ns	(0.63)
		ETD	2.05ns	2.09ns	1.91ns	(0.93)
r2	598	TWL	2537488	2560231	2559898	(1.01)
		EPD	3.58ns	3.04ns	1.51ns	(0.42)
		ETD	4.79ns	4.57ns	4.06ns	(0.85)
r3	862	TWL	3227150	3286157	3266236	(1.01)
		EPD	4.70ns	4.50ns	1.79ns	(0.38)
		ETD	6.39ns	6.35ns	5.40ns	(0.85)
r4	1903	TWL	6588826	6661947	6657174	(1.01)
		EPD	14.92ns	10.89ns	4.00ns	(0.27)
		ETD	15.70ns	13.98ns	11.58ns	(0.74)
r5	3101	TWL	9867854	9994849	9952239	(1.01)
		EPD	33.42ns	21.14ns	5.32ns	(0.16)
		ETD	28.89ns	23.59ns	17.60ns	(0.61)

Table 2: Total Wire Length [TWL], Propagation Delay Time [PD], Total Delay Time [TD], and clock skew [SKW] for the algorithms [LM] ([10, 11]) and [DM+WW]. (SPICE (LEVEL3) was used in the simulation.)

	#pins		[LM]		[DM+WW]	
			estimated	simulated	estimated	sim. (ratio to [LM])
prim1	269	TWL	131427		131877	
		PD	2.34ns	2.63ns	0.36ns	0.37ns (0.14)
		TD	6.35ns	6.31ns	5.60ns	5.63ns (0.89)
		SKW	0.00ns	0.00ns	0.00ns	0.00ns
prim2	603	TWL	306053		317296	
		PD	8.97ns	10.89ns	0.99ns	1.01ns (0.09)
		TD	15.90ns	15.93ns	12.77ns	12.60ns (0.79)
		SKW	0.00ns	0.03ns	0.00ns	0.00ns
r1	267	TWL	1289004		1288597	
		PD	1.13ns	1.35ns	0.71ns	0.80ns (0.59)
		TD	2.05ns	2.19ns	1.91ns	2.04ns (0.93)
		SKW	0.00ns	0.00ns	0.00ns	0.00ns
r2	598	TWL	2537488		2559898	
		PD	3.58ns	4.58ns	1.51ns	1.69ns (0.37)
		TD	4.79ns	5.08ns	4.06ns	4.15ns (0.82)
		SKW	0.00ns	0.00ns	0.00ns	0.00ns
r3	862	TWL	3227150		3266236	
		PD	4.70ns	6.07ns	1.79ns	1.98ns (0.37)
		TD	6.39ns	6.71ns	5.40ns	5.44ns (0.81)
		SKW	0.00ns	0.01ns	0.00ns	0.00ns
r4	1903	TWL	6588826		6657174	
		PD	14.92ns	16.73ns	4.00ns	4.44ns (0.27)
		TD	15.70ns	17.10ns	11.58ns	11.47ns (0.67)
		SKW	0.00ns	0.02ns	0.00ns	0.00ns
r5	3101	TWL	9867854		9952239	
		PD	33.42ns	32.92ns	5.32ns	5.83ns (0.18)
		TD	28.89ns	33.23ns	17.60ns	17.30ns (0.52)
		SKW	0.00ns	0.05ns	0.00ns	0.01ns

6 Remarks

6.1 NonZero-Skew Routing

The zero-skew routing for given *dictinct* $\tau(v_i^s)$'s for leaves v_i^s is called the *nonzero-skew routing*. We show in this section that results for an exact nonzero-skew routing will generally be impractical.

Suppose that two leaves v_1^s and v_2^s are merged, $\tau(v_2^s) - \tau(v_1^s) = 1$ps, and $C(v_1^s) = C(v_2^s) = 30$fF. Also, $r = 0.003\Omega$ and $c = 0.02$fF for unit length ($= 0.1\mu$m) and minimum width ($= 1\mu$m) wire[3]. A solution for the equation (1) is $l_1 \approx 450\mu$m and $l_2 = 0$. It seems that a wire of length 450μm is impractical to adjust the time difference of only 1ps.

If the clock net is hierarchically designed, and the time difference of 1ps comes from 'lower-level' clock nets whose roots are v_1^s or v_2^s, the other adjustment method would be to lengthen the wire at the root in the clock net rooted by v_1^s, so that longer delay is expected for the clock net. From the equation (19), in order to add 1ps delay to the clock net, we have only to add 0.3μm wire at the root v_1^s, where we assume that $\beta = 20[\mu A/V^2]$.

Therefore, it would be better to adjust the clock skew in a hierarchical clock design by 'gate delay' than by 'propagation delay.' From this reason, we assumed that $\tau(v_i^s) = 0$ for all leaves v_i^s.

6.2 Approximation for Sum of Ratio R_v

We assumed in previous sections that we can approximate the sum R_v of ratio (length/width) over all wire segments p from the root v_r to an internal node v, i.e.,

$$R_v = \sum_{p \in \{\text{path from } v_r \text{ to } v\}} \frac{\text{length of } p}{\text{width of } p}.$$

In this section, we describe a two-phase approximation method for R_v.

In this method, we construct the zero-skew routing for the given root and leaves twice. In the first phase, we approximate the widths of all wire segments by w_1^* for v in the equation (16). Thus, in order to calculate R_v, we have only to know the path length from v_r to v. It has been proven that, in equi-distant routings, the path length from the root to all leaves is easily calculated from the positions of the root and leaves [12]. If we approximate the path length of a zero-skew routing by that of an equi-distant routing, the path length from v_r to v can easily be calculated because we have already known the average path length from v_1 (or v_2) to the leaves.

Once a zero-skew routing is generated, the value R_v can be calculated for each node v on the zero-skew routing. In the second phase, we use this value for the approximation of R_v. It is obvious

[3] These are typical values in benchmark data r1-r5 [20].

that more precise approximation will be achieved if the method in the second phase is repeated several times.

7 Conclusions

Delay-time estimation formulae and wire-width optimization methods have been proposed for zero-skew routings. Computational experiments showed that our methods with a clustering-based zero-skew routing algorithm achieved 10%-50% reduction of the total delay time compared with the best known algorithm.

References

[1] P. Antognetti and G. Massobrio: *Semiconductor Device Modeling with SPICE.* McGraw-Hill, New York, New York, 1987.

[2] H. B. Bakoglu: Optimal Interconnection Circuits for VLSI. *IEEE Transactions on Electron Devices*, Vol. ED-32 (1985), No. 5, pp.903-909.

[3] H. B. Bakoglu: *Circuits, Interconnections, and Packaging for VLSI.* Addison-Wesley, Reading, Massachusetts, 1990.

[4] K. D. Boese and A. B. Kahng: Zero-Skew Clock Routing Trees with Minimum Wirelength. *Proc. of IEEE International ASIC Conference*, 1992, pp.1-1.1 - 1-1.5.

[5] T. H. Chao, Y. C. Hsu, and J. M. Ho: Zero Skew Clock Net Routing. *Proc. of the 29th Design Automation Conference*, 1992, pp.518-523.

[6] T. H. Chao, Y. C. Hsu, J. M. Ho, K. D. Boese, and A. B. Kahng: Zero Skew Clock Routing with Minimum Wirelength. *IEEE Transactions on Circuits and Systems*, to appear.

[7] J. Cong, A. B. Kahng, and G. Robins: Matching-Based Methods for High-Performance Clock Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, to appear.

[8] J. Cong, K.-S. Leung, and D. Zhou: Performance-Driven Interconnect Design Based on Distributed RC Delay Model. *Technical Report CSD-920043*, Computer Science Department, UCLA, 1992.

[9] M. Edahiro and T. Yoshimura: Minimum Path-Length Equi-Distant Routing. *Proc. of 1992 IEEE Asia-Pacific Conference on Circuits and Systems*, 1992, pp.41-46.

[10] M. Edahiro: A Clustering-Based Optimization Algorithm in Zero-Skew Routings. *Technical Report CS-TR-416-93*, Department of Computer Science, Princeton University, 1993.

[11] M. Edahiro: A Clustering-Based Optimization Algorithm in Zero-Skew Routings. *Proc. of the 30th Design Automation Conference*, to appear, 1993.

[12] M. Edahiro: Equi-Spreading Tree in Manhattan Distance, submitted to *Algorithmica*.

[13] W. C. Elmore: The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers. *Journal of Applied Physics*, Vol. 10 (1948), pp.55-63.

[14] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh: Clock Routing for High-Performance ICs. *Proc. of the 27th Design Automation Conference*, 1990, pp.573-579.

[15] A. Kahng, J. Cong, and G. Robins: High-Performance Clock Routing Based on Recursive Geometric Matching. *Proc. of the 28th Design Automation Conference*, 1991, pp.322-327.

[16] Y. M. Li and M. A. Jabri: A zero-skew clock routing scheme for VLSI circuits. *Proc. of 1992 International Conference on Computer-Aided Design*, 1992, pp.458-463.

[17] J. Rubinstein, P. Penfield, and M. A. Horowitz: Signal Delay in RC Tree Networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. CAD-2 (1983), No. 3, pp.202-211.

[18] T. Sakurai: Approximation of Wiring Delay in MOSFET LSI. *IEEE J. of Solid-State Circuits*, Vol. SC-18 (1983), No. 4, pp.418-426.

[19] N. Weste and K. Eshraghian: *Principles of CMOS VLSI Design: A Systems Perspective.* Addison-Wesley, Reading, Massachusetts, 1985.

[20] R. S. Tsay: Exact Zero Skew. *Proc. of the 1991 International Conference on Computer-Aided Design*, 1991, pp.336-339.

[21] D. Zhou, F. P. Preparata, and S. M. Kang: Interconnection Delay in Very High-Speed VLSI. *IEEE Transactions on Circuits and Systems*, Vol. 38 (1991), No. 7, pp.779-790.

Appendix

A Proofs and Formula Derivations

A.1 Theorem 1

For each node v and leaves $v_i^s \in S_v$ in a zero-skew routing,

$$V(v) = V(v_i^s) + \tau(v)\frac{dV(v_i^s)}{dt}, \quad (3)$$

$$I(v) = C(v)\frac{dV(v_i^s)}{dt}, \quad (4)$$

where $V(v)$ and $I(v)$ are voltage and current values for the node v.

Proof: We prove it by induction.

i) v is a leaf. ($v = v_i^s$)

Since $S_v = \{v_i^s\}$ and $\tau(v) = \tau(v_i^s) = 0$, $V(v) \equiv V(v_i^s)$. Also, it is clear that $I(v) = C(v)\frac{dV(v_i^s)}{dt}$ because the only circuit element is the capacitive load $C(v) = C(v_i^s)$ (Fig. A1 (a)).

ii) v is an internal node with children v_1 and v_2

Figure A1 (b) depicts the equivalent circuit for this case, in which l_1 (l_2) and w_1 (w_2) are wire length and width from v to v_1 (v_2). Current values for wire capacitors are denoted by i_{11}, i_{12}, i_{21}, and i_{22}.

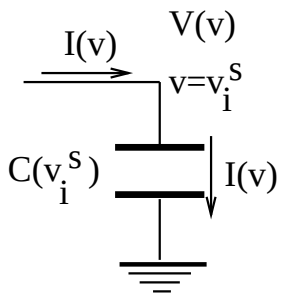
From the assumption of induction, the following formulae are satisfied:

$$\begin{aligned} V(v_1) &= V(v_i^s) + \tau(v_1)\frac{dV(v_i^s)}{dt}, \\ I(v_1) &= C(v_1)\frac{dV(v_i^s)}{dt}, \\ V(v_2) &= V(v_j^s) + \tau(v_2)\frac{dV(v_j^s)}{dt}, \\ I(v_2) &= C(v_2)\frac{dV(v_j^s)}{dt}, \end{aligned}$$

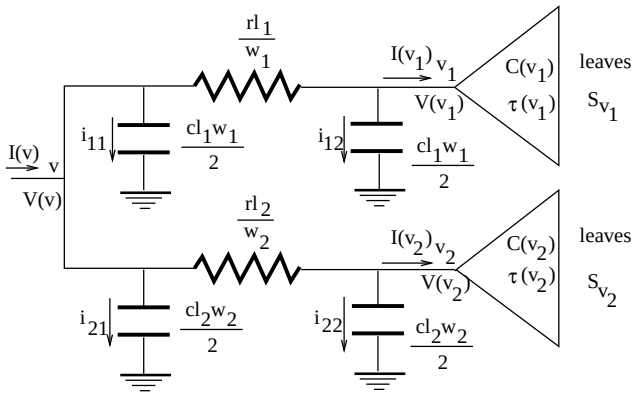
where $\forall v_i^s \in S_{v_1}$ and $\forall v_j^s \in S_{v_2}$.

Obviously,

$$i_{11} = \frac{cl_1w_1}{2}\frac{dV(v)}{dt},$$



(a) v is a leaf v_i^s



(b) v is an internal node

Figure A1: Circuit Model for Zero-Skew Merge at v.

$$
\begin{aligned}
i_{12} &= \frac{cl_1w_1}{2}\frac{dV(v_1)}{dt}, \\
i_{21} &= \frac{cl_2w_2}{2}\frac{dV(v)}{dt}, \\
i_{22} &= \frac{cl_2w_2}{2}\frac{dV(v_2)}{dt}, \\
V(v) &= \frac{rl_1}{w_1}(I(v_1)+i_{12})+V(v_1) \\
&= \frac{rl_2}{w_2}(I(v_2)+i_{22})+V(v_2), \\
I(v) &= I(v_1)+I(v_2)+i_{11}+i_{12}+i_{21}+i_{22}.
\end{aligned}
$$

Thus, in zero-skew routings, the following formula should be satisfied.

$$
\begin{aligned}
V(v) &= \frac{rl_1}{w_1}\left[C(v_1)\frac{dV(v_i^s)}{dt}+\frac{cl_1w_1}{2}\frac{dV(v_1)}{dt}\right]+V(v_1) \\
&= \frac{rl_1}{w_1}\left[C(v_1)\frac{dV(v_i^s)}{dt}+\frac{cl_1w_1}{2}\frac{d}{dt}\left(V(v_i^s)+\tau(v_1)\frac{dV(v_i^s)}{dt}\right)\right]+ \\
&\qquad V(v_i^s)+\tau(v_1)\frac{dV(v_i^s)}{dt} \\
&= V(v_i^s)+\left[\frac{rl_1}{w_1}\left(\frac{cl_1w_1}{2}+C(v_1)\right)+\tau(v_1)\right]\frac{dV(v_i^s)}{dt}+o(\varepsilon^2) \\
&= V(v_i^s)+\tau(v)\frac{dV(v_i^s)}{dt}.
\end{aligned}
$$

In real circuits, both $\frac{rl_1}{w_1}(\frac{cl_1w_1}{2}+C(v_1))$ and $\tau(v_1)$ are small values ($<\varepsilon=10^{-7}$sec). Therefore, the value $\frac{rl_1}{w_1}\frac{cl_1w_1}{2}\tau(v_1)$ is the second order term, which was ignored in the last formula.

Similarly,

$$
V(v) = V(v_j^s)+\tau(v)\frac{dV(v_j^s)}{dt}.
$$

Consequently, the equation (3) is proven, i.e., for each node v and leaves $v_i^s \in S_v (= S_{v_1} \cup S_{v_2})$ in a zero-skew routing,

$$
V(v) = V(v_i^s)+\tau(v)\frac{dV(v_i^s)}{dt}.
$$

Next, we prove that $I(v) = C(v)\frac{dV(v_i^s)}{dt}$.

Since the equation (3) should be satisfied universally for all $v_i^s \in S_v$,

$$\forall v_i^s, v_j^s \in S_v, \quad V(v_i^s) \equiv V(v_j^s).$$

Now, for all $v_i^s \in S_v$,

$$\begin{aligned} I(v) &= I(v_1) + I(v_2) + i_{11} + i_{12} + i_{21} + i_{22} \\ &= C(v_1)\frac{dV(v_i^s)}{dt} + C(v_2)\frac{dV(v_i^s)}{dt} + \frac{cl_1w_1}{2}\frac{dV(v)}{dt} + \frac{cl_1w_1}{2}\frac{dV(v_1)}{dt} + \\ &\quad \frac{cl_2w_2}{2}\frac{dV(v)}{dt} + \frac{cl_2w_2}{2}\frac{dV(v_2)}{dt} \\ &= (C(v_1) + C(v_2) + cl_1w_1 + cl_2w_2)\frac{dV(v_i^s)}{dt} + o(\varepsilon^2) \\ &= C(v)\frac{dV(v_i^s)}{dt}. \end{aligned}$$

■

A.2 Fall Time Estimation

The fall time is defined by time for a waveform to fall from $0.9V_{DD}$ to $0.1V_{DD}$ [19]. In order to estimate the fall time at a leaf v_i^s, we need to evaluate the voltage of v_i^s at which the transistor changes its state from the saturation to linear. In our model, at the turning point,

$$\begin{aligned} V(v_r) &= V_{DD} - V_{t_n}, \\ I(v_r) &= -\frac{\beta_n}{2}(V_{DD} - V_{t_n})^2. \end{aligned}$$

By assigning the above formulae to the equations (3) and (4),

$$V^*(v_i^s) = (1 + \frac{\beta_n \tau(v_r)}{2C(v_r)}(V_{DD} - V_{t_n}))(V_{DD} - V_{t_n}).$$

It is easy to see that, by assigning actual values to the above equation, $V^*(v_i^s)$ is more than $0.9V_{DD}$. Since the fall time at v_i^s is estimated using $V(v_i^s) \in [0.1V_{DD}, 0.9V_{DD}]$, we assume that the transistor is always on the linear state.

Now, from the differential equations (5) and (7) for the linear state, the fall time t_f is calculated by

$$\begin{aligned} t_f &= \int dt \\ &= \frac{2}{\beta_n}\int_{0.9V_{DD}}^{0.1V_{DD}} \frac{C(v_r) + \beta_n\tau(v_r)((V_{DD} - V_{t_n}) - V(v_i^s))}{V(v_i^s)^2 - 2V(v_i^s)(V_{DD} - V_{t_n})} dV(v_i^s). \end{aligned}$$

By using $x = (V_{DD} - V_{t_n}) - V(v_i^s)$,

$$t_f = \frac{2}{\beta_n} \int_{(V_{DD}-V_{t_n})-0.1V_{DD}}^{(V_{DD}-V_{t_n})-0.9V_{DD}} \frac{C(v_r) + \beta_n \tau(v_r) x}{x^2 - (V_{DD} - V_{t_n})^2} dx.$$

Since, for $a > 0$ and $a^2 > x^2$,

$$\begin{aligned} \int \frac{1}{x^2 - a^2} dx &= \frac{1}{2a} \ln\left(\frac{a-x}{a+x}\right), \\ \int \frac{2x}{x^2 - a^2} dx &= \ln(a^2 - x^2), \end{aligned}$$

we have

$$\begin{aligned} t_f &= \left[\left(\frac{C(v_r)}{\beta_n(V_{DD} - V_{t_n})} \ln \frac{(V_{DD} - V_{t_n}) - x}{(V_{DD} - V_{t_n}) + x}\right) + (\tau(v_r) \ln((V_{DD} - V_{t_n})^2 - x^2))\right]_{(V_{DD}-V_{t_n})-0.1V_{DD}}^{(V_{DD}-V_{t_n})-0.9V_{DD}} \\ &= \frac{C(v_r)}{\beta_n(V_{DD} - V_{t_n})} \left(\ln \frac{0.9V_{DD}}{2(V_{DD} - V_{t_n}) - 0.9V_{DD}} - \ln \frac{0.1V_{DD}}{2(V_{DD} - V_{t_n}) - 0.1V_{DD}}\right) + \\ &\quad \tau(v_r) \left(\ln(0.9V_{DD}(2(V_{DD} - V_{t_n}) - 0.9V_{DD})) - \ln(0.1V_{DD}(2(V_{DD} - V_{t_n}) - 0.1V_{DD}))\right) \\ &= \frac{C(v_r)}{\beta_n(V_{DD} - V_{t_n})} \left(\ln \frac{9(2(V_{DD} - V_{t_n}) - 0.1V_{DD})}{2(V_{DD} - V_{t_n}) - 0.9V_{DD}}\right) + \tau(v_r) \left(\ln \frac{9(2(V_{DD} - V_{t_n}) - 0.9V_{DD})}{2(V_{DD} - V_{t_n}) - 0.1V_{DD}}\right). \end{aligned}$$

If we make the assumption that $V_{t_n} \approx 0.2V_{DD}$,

$$t_f \approx 3.7 \frac{C(v_r)}{\beta_n V_{DD}} + 1.4\tau(v_r).$$

A.3 Theorem 2 [Wire Width Optimization]

On the domain $w_1, w_2 \geq w_{min}$ and under a condition $l = l_1 + l_2$, the function $t_d(w_1, w_2)$ has the global minimum when

$$w_1 = w_2 = w^* = \max\left(w_{min}, \sqrt{\frac{\alpha_2 r}{\frac{\alpha_1 c}{\beta V_{DD}} + \alpha_2 r c R_v} \left(\frac{C(v_1)C(v_2)}{C(v_1) + C(v_2)}\right)}\right). \tag{16}$$

The wire lengths for this optimum width are calculated by

$$l_1^* = \frac{(\tau(v_2) - \tau(v_1)) + rl(\frac{C(v_2)}{w^*} + \frac{cl}{2})}{r(cl + \frac{C(v_1)}{w^*} + \frac{C(v_2)}{w^*})}, \tag{17}$$

$$l_2^* = l - l_1^*. \tag{18}$$

Proof: Considering $l = l_1 + l_2$, from the equation (1),

$$\begin{aligned} l_1 &= \frac{(\tau(v_2) - \tau(v_1)) + rl(\frac{C(v_2)}{w_2} + \frac{cl}{2})}{r(cl + \frac{C(v_1)}{w_1} + \frac{C(v_2)}{w_2})} \\ l_2 &= l - l_1 . \end{aligned}$$

It follows that

$$\begin{aligned} \frac{\partial l_1}{\partial w_1} &= \frac{\partial l_1}{\partial \left(\frac{C(v_1)}{w_1}\right)} \frac{\partial \left(\frac{C(v_1)}{w_1}\right)}{\partial w_1} \\ &= \frac{\frac{C(v_1)}{w_1}}{cl + \frac{C(v_1)}{w_1} + \frac{C(v_2)}{w_2}} \left(\frac{l_1}{w_1}\right). \end{aligned}$$

Now, let A be $\left(\frac{\alpha_1 c}{\beta V_{DD}} + \alpha_2 rcR_v\right)$ and B be $(\alpha_2 r)$. Then, from the equation (15),

$$\frac{\partial t_d}{\partial w_1} = A\frac{\partial}{\partial w_1}(w_1 l_1 + w_2 l_2) + B\frac{c}{2}\frac{\partial (l_1^2)}{\partial w_1} + BC(v_1)\frac{\partial}{\partial w_1}\left(\frac{l_1}{w_1}\right).$$

By solving $\frac{\partial t_d}{\partial w_1} = 0$,

$$A\left((2 - \frac{w_2}{w_1})\frac{w_2}{w_1}C(v_1) + C(v_2) + clw_2\right) w_1^2 = BC(v_1)(C(v_2) + cl_2 w_2).$$

In this equation, we ignore the wire capacitance clw_2 and $cl_2 w_2$ because they are small enough compared with the load capacitance $C(v_1)$ and $C(v_2)$. It follows that

$$A\left((2 - \frac{w_2}{w_1})\frac{w_2}{w_1}C(v_1) + C(v_2)\right) w_1^2 = BC(v_1)C(v_2).$$

Similarly, by solving $\frac{\partial t_d}{\partial w_2} = 0$,

$$A\left((2 - \frac{w_1}{w_2})\frac{w_1}{w_2}C(v_2) + C(v_1)\right) w_2^2 = BC(v_1)C(v_2).$$

The global minima must satisfy the both equations. From the above two equations, we obtain a necessary condition:

$$(w_1 - w_2)(C(v_2)w_1 + C(v_1)w_2) = 0.$$

It is easy to see that only $w_1 = w_2$ satisfies the both equations and that the solution gives the minimum. Considering boundary conditions, the optimum wire width formula is

$$w_1^* = w_2^* = w^* \quad = \quad \max\left(w_{min}, \sqrt{\frac{\alpha_2 r}{\frac{\alpha_1 c}{\beta V_{DD}} + \alpha_2 r c R_v}\left(\frac{C(v_1)C(v_2)}{C(v_1)+C(v_2)}\right)}\right).$$

■