A CLOCK DISTRIBUTION SCHEME FOR NON-SYMMETRIC VLSI CIRCUITS

P. Ramanathan and Kang G. Shin

Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109–2122.

ABSTRACT

The control of clock skew to an acceptable small fraction of the clock period is essential for the correct and dependable operation of any digital system. Within a VLSI circuit, the foremost factor responsible for this clock skew is the difference in the length of clock lines to different functional elements in the circuit. In this paper we propose a clock distribution scheme that minimizes the difference in the length of clock lines.

The proposed scheme uses the hierarchy created by the clock buffers to parallelize the distribution of the clock signal. At each hierarchical level, an exhaustive search of paths with intelligent pruning is used to determine the optimal layout of clock lines at that level. Unlike other related work in this area, both delay and skew are taken into account in determining the layout.

1 Introduction

Within most VLSI circuits, data transfer between the functional elements is synchronized by a single control signal, the processing clock. This signal typically constrains the timing and performance of the circuit. It is, therefore, imperative to develop a design methodology for distributing the clock signal without creating any timing uncertainties.

There are four main factors responsible for the timing uncertainties in a VLSI circuit: (i) differences in the length of lines that deliver the clock signal to two different functional elements, (ii) differences in delays through any active elements inserted in the clock lines (e.g., buffers), (iii) differences in parameters such as resistivity and dielectric constant that determine the line time constant, and (iv) differences in threshold voltages of different elements. These factors can be eliminated by using an appropriate clock distribution scheme.

Wann and Franklin [1] proposed a distribution scheme that ensures equal lengths for all paths when the elements are identical and placed in the form of a symmetric array. This scheme, however, is not suitable for distributing the clock signal in a general VLSI circuit where the functional elements are of different sizes and where the placement of the elements is governed by the interconnection between the elements and is usually not in an array form. Likewise, the existing schemes for eliminating the other factors [2, 3, 4] are also not suitable for distributing a clock in a general VLSI circuit because they are sensitive to the fabrication process.

In this paper we propose a non-symmetric distribution scheme that will minimize the clock skew due to line lengths for functional elements with different sizes and arbitrary placement. It also accounts for the difference in delays caused by the clock buffers. Unlike the other related work in this area, both delay and skew are considered in determining the layout of the clock lines. This is important because layouts that have minimum skew might have long lines that result in large delays, and hence, degraded performance. On the other hand, layouts that have minimum delay might have large skews that could affect the correctness of the circuit. The objective used in this paper for determining the clock layout minimizes the skew subject to minimum longest delay.

The scheme proposed here assumes that a floorplan of the modules is given. This floorplan would have been determined by using a placement algorithm based on the other signals interconnecting the sub-modules that constitute that module. For this floorplan, the scheme identifies an entry point from which the clock is distributed and the route of the clock from the entry point to each of the sub-modules. The entry point is selected in such a way that the delay to the farthest sub-module from that entry point is minimum. This will minimize the maximum delay in distributing the clock signal to the sub-modules. The optimal layout of the clock lines from the selected entry point is determined by an exhaustive search of all paths with intelligent pruning. The main advantage of the scheme is that the computation of the optimal layout can be easily parallelized.

This paper is organized as follows. In Section 2, the clock layout problem is defined formally and some notations and terms are introduced. An algorithm to select the entry point is presented in Section 3. Section 4 describes an algorithm to determine the optimal clock layout from the selected entry point. An implementation of the scheme is discussed in Section 5. The paper concludes with Section 6.

2 Problem Formulation

A *module* is a group of functional elements to which the clock signal is being distributed. For clarity, the elements contained in a module will be referred to as the *sub-modules*. A *floorplan* is a relative placement of the sub-modules in the routing area. A *channel* is a rectangular zone of empty area between the sub-modules through which the signals can be routed. A *track* is a sub-unit within a channel through which a single signal can be routed. A channel is said to be *horizontal (vertical)* if the tracks within the channel are horizontal (vertical). A channel is said to be *peripheral* if it lies in the periphery of the floorplan. There are three possible types of intersection between a horizontal and a vertical channel: L-type, T-type, and $+$-type. A *decision point* is the point of a T-type or a $+$-type intersection.

The work reported here is supported in part by the NASA under grants NAG–1–296 and NAG–1–981. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NASA.

The floorplan can be represented by an undirected, weighted graph referred to as a *placement graph*. The vertices of the placement graph are the decision points in the floorplan, the input clock terminals of all the sub-modules, and the output clock terminals of all clock buffers. Clock buffers are sub-modules whose output drives the input clock signals of other sub-modules. Vertices v_l and v_k are connected by an edge if and only if there is a channel from v_l to v_k not containing any other vertex of the placement graph. In other words, there is a one-to-one correspondence between the edges of the placement graph and the channels in the floorplan. The weight of an edge $\{v_l, v_k\}$ is equal to the physical distance in the channel between the two points corresponding to v_l and v_k except when either v_l or v_k is an input clock terminal of one of the sub-modules. It is therefore proportional to the delay that will be induced on a clock signal if that channel is used as a part of the clock layout. The weight of edges $\{v_l, v_k\}$ in which one of v_l or v_k is an input clock terminal will be used to account for the delays induced by clock buffers. The assignment of weights to these edges will be discussed later in this section. One needs to consider not only the edges of the placement graph but also the line segment between two arbitrary points on the same edge. The line segment between two points u and v on the same edge will be denoted by $[u, v]$. The weight of $[u, v]$ is equal to the physical distance in the channel between the points corresponding to u and v. The weight of $[u, v]$ is sometimes referred to as the distance between u and v and is denoted by $d(u, v)$.

Let P be a floorplan of a module M that is comprised of m sub-modules M_1, M_2, ..., and M_m. There is an entry point into the floorplan from which the clock is distributed to all the sub-modules. However, all sub-modules may not receive their clock signal directly from this entry point. Instead, some of the sub-modules will receive their clock signal indirectly from the output of some other sub-module (i.e., clock buffer) in M. For simplicity of presentation, introduce a fictitious clock buffer that drives the entry point to the floorplan. For each sub-module M_i, $1 \leq i \leq m$, it is then possible to associate a unique level, *Level(M_i)*, that is equal to the number of clock buffers in the route from the clock entry point to the sub-module. In other words, sub-modules directly driven from the entry point are said to be in *Level 1*, while those driven by sub-modules in Level 1 are said to be in *Level 2*, and so on. Let *max_level* $= \max_{1 \leq i \leq m}$ *Level(M_i)*, and *Buffer(M_i)* be the clock buffer, say, M_j that drives the clock signal of M_i. Define *Buffer(M_i)* of a sub-module M_i at level 1 as the fictitious clock buffer that drives the clock entry point to the floorplan.

Definition 1: In a placement graph, a *simple path* from s to t is a finite sequence of distinct points $u_0 u_1 \cdots u_r$, $r \geq 1$, such that: (i) $u_0 = s$ and $u_r = t$, (ii) if $r > 1$, then $u_1, u_2, \ldots, u_{r-1}$ are vertices of the placement graph, and (iii) $\forall\, j \in \{0, \ldots, r-1\}$, u_j and u_{j+1} lie on a common edge. The term "a simple path from M_i to M_j" will be used to refer to the simple path from the output clock terminal of M_i to the input clock terminal of M_j.

Definition 2: A *c-route* from an entry point e to a sub-module M is a concatenation of simple paths from e to B_1, B_1 to B_2, B_2 to B_3, ..., B_{l-1} to B_l, and B_l to M, where $B_1, B_2, \ldots, B_l$ are such that B_l = *Buffer(M)* and B_{j-1} = *Buffer(B_j)* for all $l \geq j \geq 2$.

Definition 3: The sum of the distances between adjacent points in a c-route will be referred to as its *c-length*.

Definition 4: A *c-layout* from an entry point e is a tuple of c-routes from e to all sub-modules in the floorplan.

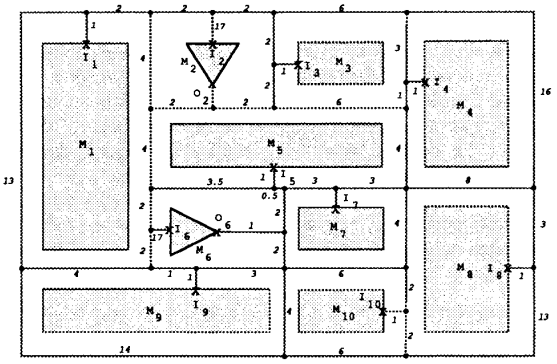

Figure 1: Floorplan of sub-modules in M.

Let SM_l denote the set of all sub-modules in level l. Let $\eta_i^{P,e}$ denote the c-length of the shortest simple path from *Buffer(M_i)* to M_i in the floorplan P when e is the clock entry point. Define $\mu_l^{P,e}$ to be $\max_{i \in SM_l} \eta_i^{P,e}$. We are now in a position to define the weights of the edges $\{v_l, v_k\}$ in the placement graph in which either v_l or v_k is an input clock terminal of a sub-module. Let I_j be the input clock terminal of sub-module M_j. If *Level(M_j)* is *max_level* or if M_j is not a clock buffer, then the weight of the edge $\{v_k, I_j\}$ is equal to the physical distance between v_k and I_j in the channel, else the weight of the edge is equal to $\mu_{l+1}^{P,e} + b + c$, where l = *Level(M_j)*, b is a constant corresponding to the delay introduced by M_j, and c is equal to the distance in the channel between the points v_k and I_j. The intuitive reason for assigning the weights to these edges in this manner is explained later in Example 1.

There are two main objectives in determining a c-layout. The first objective is to minimize the maximum difference in the c-length to the different sub-modules. The second objective is to minimize the c-length to all sub-modules. More formally, let E be the set of all points in the periphery of the floorplan. It is the set of points from which an entry point is to be selected. Let $Y^{P,e}$ denote a c-layout from e in the floorplan P. Let $L_i(Y^{P,e})$ denote the c-length of the simple path from *Buffer(M_i)* to M_i in the c-route from e to M_i in $Y^{P,e}$. Then the clock distribution problem is to determine an entry point e such that $\mu_1^{P,e} \leq \mu_1^{P,s}$, $\forall s \in E$ and then from that entry point determine a c-layout $Y^{P,e}$ that minimizes $\sum_{i \in SM_l} \left| L_i(Y^{P,e}) - \mu_l^{P,e} \right|$, $\forall\, 1 \leq l \leq$ *max_level*. Since $\mu_1^{P,e}$ represents the delay from the entry point to the farthest (from that entry point), the first criterion minimizes the longest delay. The second criterion tries to increase the length of clock lines from the entry point to all the sub-modules to equal line length to the farthest sub-module. Consequently, the overall objective identifies a layout of clock lines that has minimum skew subject to minimum worst case delay.

Example 1: Consider the floorplan in Figure 1. The module M is comprised of ten sub-modules M_1, M_2, ..., M_{10}. Sub-modules M_2 and M_6 are clock buffers whose input is driven directly from the clock entry point of M. In addition, the input of sub-module M_8 is also driven directly from the entry point. M_2 drives the input clock signals of sub-modules M_1, M_3, M_4, and M_5 while M_6 drives the

input clock signals of sub-modules M_7, M_9, and M_{10}. The sub-modules have been placed within the given area in such a way that there is ample space to route all the signals between them. The placement graph for this floorplan is also shown in Figure 1. In this graph the vertices corresponding to the decision points are indicated by • whereas those corresponding to the clock terminals of the sub-modules are indicated by ×. The weight of the edges in this graph are indicated in italics. The problem is to determine a clock entry point for the module M, the route of the clocks from this entry point to the sub-modules M_2, M_6 and M_8, and the route of the clocks from the output of M_2 and M_6 to the input clock terminals of the other sub-modules.

In this example it is easy to illustrate the reason for assigning larger weights to the edges $\{v_j, v_k\}$ in which either v_j or v_k is an input terminal of a clock buffer. Consider the sub-module M_7. The clock input to M_7 is not directly driven by the clock entry point but by the output of M_6. Therefore, total delay from the clock entry point to the clock input of M_7 is equal to the sum of the delays from the entry point to the input of M_6 ($\approx \mu_2^{P,e}$), delay introduced by M_6 ($= b$), and the delay from the output of M_6 to the input of M_7 ($= c$). By assigning $\mu_2^{P,e} + b + c$ as the weight of the edge incident on I_6 we can determine the c-route from the entry point to M_6 in parallel with the c-route from M_6 to M_7. In this example, b is assumed to be 5 and c is assumed to be 1. ■

3 Selection of the Entry Point

The goal of this section is to select a point in E that minimizes the longest delay, i.e., select $e \in E$ such that $\mu_1^{P,e} \leq \mu_1^{P,s}$ for all $s \in E$. Without loss of generality any point in E can be chosen as the origin and the other points can be represented as the distance from the origin along the periphery of the floorplan. In other words, E can be represented as a set of real numbers from 0 to $\|E\|$, where $\|E\|$ is the perimeter of the floorplan. Let $D(s_1)$ denote the distance of s_1 from the origin along the periphery.

Theorem 1: If $s_1 \in E$ and $s_2 \in E$ are two points on the same edge of a placement graph with $D(s_1) < D(s_2)$, then for all $x \in [s_1, s_2]$ and all $M_i \in SM_1$

$$\text{if } \left[D(x) - D(s_1) \leq \frac{\eta_i^{P,s_2} - \eta_i^{P,s_1} + D(s_2) - D(s_1)}{2} \right] \text{ then}$$
$$\eta_i^{P,x} = \eta_i^{P,s_1} + D(x) - D(s_1)$$
else
$$\eta_i^{P,x} = \eta_i^{P,s_2} + D(s_2) - D(x).$$

Proof: Outlined in [5]. ■

Entry, an algorithm for identifying the best entry point is based on Theorem 1. It can be informally explained as follows. Let $s_1, s_2, \ldots, s_n$ be the decision points in E such that $D(s_i) < D(s_{i+1})$ for $1 \leq i \leq n-1$. Let $s_{n+1} \equiv s_1$. First determine the shortest c-routes to the sub-modules in level 1 from each of the points $s_1, s_2, \ldots, s_n$. Then start at s_1 and move along the periphery of the floorplan from s_1 to s_2, s_2 to s_3, $\ldots$ s_{n-1} to s_n, and then back to s_1. To go from s_k to s_{k+1} it may be necessary to move through several intermediate points. Each move is comprised of determining the best entry point based on the path already traveled and then selecting the next point to move.

For example, let $u \in [s_k, s_{k+1}]$ be an intermediate point to which the algorithm has moved while going from s_k to s_{k+1}. If M_i is the farthest sub-module from u, then the next point s to which the algorithm moves is given by

$$\text{if } \left[D(s) - D(u) < \frac{\eta_i^{P,v} - \eta_i^{P,u} + D(v) - D(u)}{2} \right] \text{ then}$$
$$s = D^{-1}\left[D(u) + \frac{\eta_i^{P,v} - \eta_i^{P,u} + D(v) - D(u)}{2} \right];$$
else
$$s = \min\{z_{ij} : M_j \in SM_1, z_{ij} > s\},$$

where z_{ij} is the point of intersection between $\eta_i^{P,x}$ and $\eta_j^{P,x}$ in $[s_k, s_{k+1}]$. After moving to s the algorithm updates the current best entry point and then repeats the above steps until it reaches s_{k+1}.

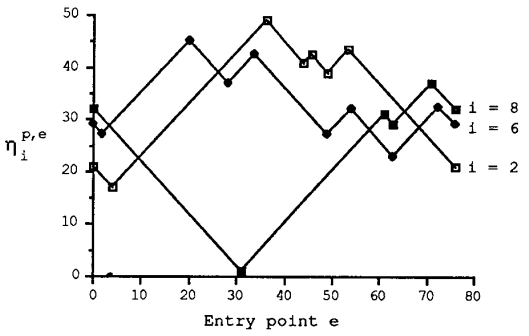

Figure 2: Variation in $\eta_i^{P,x}$ for the floorplan in Figure 1

Example 1 (cont'd.): Consider the floorplan in Figure 1. Recall that only sub-modules M_2, M_6 and M_8 are in level 1. Figure 2 shows the variation in $\eta_i^{P,x}$ for the three sub-modules in this floorplan. The plots are based on vertex 0 (see Figure 1) as the origin. Clearly, $\eta_i^{P,x}$ for $i = 2, 6$ and 8 are comprised of straight lines of slopes $+1$ and -1 as indicated by Theorem 1. Since in this example $\mu_1^{P,x} = \max\{\eta_2^{P,x}, \eta_6^{P,x}, \eta_8^{P,x}\}$ and $\mu_{opt} = \min_{x \in [0,76]} \mu_1^{P,x}$, it follows by either graphically or by using algorithm **Entry** that the optimal entry point is at distance 3.5 units along the periphery from the origin and $\mu_{opt} =$ 28.5 units. ■

4 Optimization Problem

The problem addressed in this section can be stated as follows: given a floorplan P and an entry point e determine a c-layout $Y^{P,e} \equiv (R_1^{P,e}, R_2^{P,e}, \ldots, R_m^{P,e})$ that minimizes $\sum_{i \in SM_l} \left| L_i(Y^{P,e}) - \mu_l^{P,e} \right|$ for all $1 \leq l \leq max\_level$. Since there is no inter-dependence between the terms of the summation,

$$\min_{Y^{P,e}} \sum_{i \in SM_l} \left| L_i(Y^{P,e}) - \mu_l^{P,e} \right| \iff \min_{R_i^{P,e}} \left| L(R_i^{P,e}) - \mu_l^{P,e} \right| \tag{4.1}$$

for all i such that $M_i \in SM_l$, and where $L(R_i^{P,e}) = L_i(Y^{P,e})$ is used just to emphasize that the right-hand side depends only on M_i.

Consider the placement graph G^P of a floorplan P. Let e be the selected entry point. Without loss of generality we can assume that e is a vertex of G^P. (If e is not a vertex of G^P, then modify G^P to include e as a vertex by augmenting the vertex set of G^P with e and replacing the edge $\{v_1, v_2\}$ containing e by two edges $\{v_1, e\}$

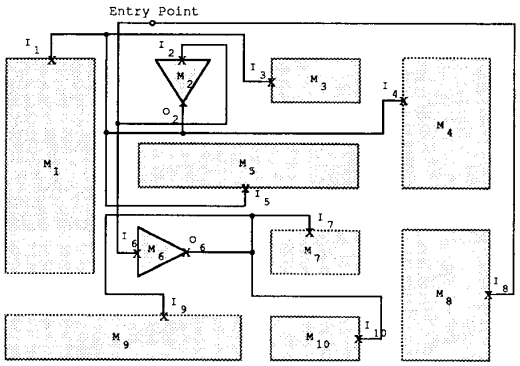

Figure 3: Optimal clock layout for floorplan in Figure 1

and $\{e, v_2\}$ of weights $d(v_1, s)$ and $d(e, v_2)$, respectively.) Since determining an $R_i^{P,e}$ that minimizes $\left|L(R_i^{P,e}) - \mu_l^{P,e}\right|$ is equivalent to finding a simple path in a graph of length closest to a given length $\mu_l^{P,e}$, the problem is NP-hard [6]. An exhaustive search (with intelligent pruning) of the paths from the entry point to the clock terminals of all the sub-modules is therefore used to find the optimal c-layout.

Route, the proposed algorithm for searching the paths, is recursive in nature. The basic idea of the algorithm is as follows. Consider a clock buffer B_l at level l. It drives the clock signals of some of the sub-modules in level $l+1$. Let *Drives(B_l)* denote the set of sub-modules driven by B_l. For clarity of presentation, it is convenient to assume that there is only one sub-module, say M_i, in *Drives(B_l)*. Extending the description to the case when *Drives(B_l)* contains more than one sub-module is relatively simple.

The algorithm starts out with a partial path that contains only the output clock terminal of B_l. At each step of the algorithm a vertex is added to the partial path until either the clock terminal of M_i is reached, or it is clear that there is no completion of the partial path that will result in a c-route with better objective value than the best known objective at that step. The algorithm terminates when every path from the output of B_l to the clock terminal of M_i has been either investigated or pruned. The pruning of the search space occurs while forming the set of possible vertices that can be appended to the current partial path. Vertices are disregarded if the shortest c-route to M_i from that vertex conclusively shows that there is no better completion of the current partial path.

5 Implementation

Entry and **Route** were implemented in C on an Apollo DN4000. The placement graph was the input to the program and the output was the optimal entry point and the route of clocks to all the sub-modules. This implementation was used to distribute clocks in several examples. In this paper we show the results for two examples.

The first example is that of the floorplan in Figure 1. As discussed in Section 3, the optimal entry point for this floorplan is at a distance 3.5 units from vertex 0. The route of the clocks from this entry point is shown in Figure 3. In this layout it is easy to see how the increased weight on the edges incident I_2 and I_6 account for the delays induced by the clock buffers M_2 and M_6, respectively. For instance, the delay to M_8 from the entry point is 28.5 units. The delay to M_1 from the entry point is the delay to M_2 (16.5 units) plus the delay introduced by M_2 (5 units) plus the delay from M_2 to M_1 (10 units), i.e., a total delay of 31.5 units. Being a small example the CPU time required to determine all these routes was less than a second.

The second example was a custom VLSI chip called the *routing controller* [7]. It has been implemented using the CONCORDE™ silicon compiler in a $2\mu m$ CMOS process and contains around 20,000 transistors. The routing of clock signals to the sub-modules were initially carried out using the placement and routing tools in the compiler. Since these placement and routing tools do not take into account the clock skew problem, the skew (in terms of difference in line lengths) was around 4000 microns between the sub-modules driven by one of the clock buffers. By using **Route**, the skew can be reduced to 20 microns for the same set of sub-modules and the algorithm requires only 4.5 seconds of CPU time on an Apollo DN4000 with 16 MB of memory. This should be contrasted to the several hours of CPU time that was required to route the other signals in the floorplan.

These two examples clearly indicate that the algorithm proposed in this paper can be used to distribute clocks even in large VLSI circuits.

6 Conclusion

A hierarchical clock distribution scheme suitable for general VLSI circuits is presented in this paper. The unique features of the proposed scheme are: (i) both delay and skew are taken into account in determining the clock layout, and (ii) the proposed scheme can be easily parallelized.

References

[1] D. F. Wann and M. A. Franklin, "Asynchronous and clocked control structures for VLSI based interconnection networks," *IEEE Trans. Comput.*, vol. C-32, no. 3, pp. 284–293, March 1983.

[2] K. D. Wagner, "A survey of clock distribution techniques in high-speed computer systems," Technical Report CRC 86-20, Center for Reliable Computing, Stanford University, December 1986.

[3] E. G. Friedman and S. Powell, "Design and analysis of a hierarchial clock distribution system for syncrhonous standard cell/macrocell VLSI," *IEEE Journal of Solid-State Circuits*, vol. SC-21, no. 2, pp. 240–246, April 1986.

[4] M. Shoji, "Elimination of process-dependent clock skew in CMOS VLSI," *IEEE Journal of Solid-State Circuits*, vol. SC-21, no. 5, pp. 869–880, October 1986.

[5] P. Ramanathan and K. G. Shin, "A clock distribution scheme for non-symmetric VLSI circuits," Technical report, Real-Time Computing Laboratory, Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, 1989.

[6] M. R. Garey and D. S. Johnson, *Computer and intractability: A guide to the theory of NP–completeness*, Freeman, San Francisco, 1979.

[7] J. W. Dolter, P. Ramanathan, and K. G. Shin, "A microprogrammable VLSI routing controller for HARTS," Technical Report CSE-TR-12-89, Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, 1989.

™CONCORDE is a trademark of Seattle Silicon Technology Inc.