

Simultaneous Clock Scheduling and Buffered Clock Tree Synthesis

Ivan S. Kourtev and Eby G. Friedman
 Department of Electrical Engineering
 University of Rochester
 Rochester, NY 14627

ivan@ee.rochester.edu, friedman@ee.rochester.edu

Abstract— This paper considers the problem of designing the topology of a clock distribution network in a synchronous digital system so as to enforce non-zero clock skew. A methodology and related algorithm for synthesizing the topology of the clock distribution network from a clock skew schedule derived from the circuit timing information is presented. The application of the algorithm to benchmark circuits shows that improvements of the minimum clock period ranging up to 64% can be achieved. These improvements are attained by exploiting non-zero clock skew throughout the synchronous system. Mathematically, the problem of designing the clock distribution network is formulated as an integer linear programming problem which is efficiently solvable. The algorithm for synthesizing a clock tree is demonstrated on an example circuit.

Keywords—Clocking, clock skew, linear programming, optimization, synchronous systems, VLSI.

I. INTRODUCTION

The prevailing synchronization paradigm for most digital VLSI systems is a fully synchronous environment. These *synchronous digital systems* require a global clock signal to synchronize and temporally order the functional events. For the system to function properly, the clock signal must be delivered to every register at a precise relative time. This delivery function is accomplished by the *clock distribution network*. For a modern VLSI-complexity circuit containing millions of transistors, the clock distribution network may be required to transmit the clock signal to several tens of thousands of registers scattered throughout the integrated circuit. A clock signal from a single clock source is typically delivered to every clocked register by a circuit and interconnect system which structurally resembles a tree. Thus, clock distribution networks are often referred to as *clock trees*.

As feature sizes continue to decrease and die area increases, the quality of the on-chip clock signals has become one of the primary factors limiting circuit performance. Interconnect parasitic impedances degrade waveform shapes, effectively increasing inter-

connect delay, such that the interconnect delay can be greater than the gate delay [1, 2]. Also, as the total capacitance driven by the clock distribution network increases, the power dissipated within the clock distribution network has become a prohibitively large portion of the total on-chip power budget, *e.g.*, 40% of the power of the DEC Alpha chip is dissipated within the clock distribution network [3]. Additionally, avoiding clock hazards at circuit frequencies significantly greater than 200 MHz can be an enormous design challenge. As a consequence, the design of the clock distribution network, particularly in high performance applications, may require a significant investment of both time and effort in order to achieve these high performance goals [3].

Many different approaches to designing the clock distribution network of a synchronous digital system have been proposed and applied. With a few exceptions [4–6], these approaches have as a final objective a clock tree with minimal or zero global clock skew. This objective may be achieved by the application of different routing strategies [7–10], by buffered clock tree synthesis, the use of symmetric n -ary trees (most notably H-trees), or a distributed series of buffers connected as a mesh [3, 11].

This paper is organized as follows. First, relevant notations are introduced and the problem is formulated in section II. The proposed methodology and algorithm are presented in section III. The application of this algorithm is demonstrated on a practical circuit in section III, while some concluding remarks are offered in section IV.

II. PROBLEM FORMULATION

In this section, certain important properties of a synchronous digital system are outlined, the model used in this paper to describe these systems is formulated, and the notations used throughout this paper are introduced. Specifically, in section II.A, the concepts of clock scheduling are reviewed, in section II.B, the tree structure of the clock distribution network is described, and in section II.C, the optimization problem for synthesizing the clock tree is formulated.

This research was supported in part by the National Science Foundation under Grant No. MIP-9208165 and Grant No. MIP-9423886, the Army Research Office under Grant No. DAAH04-G-0323, a grant from the New York State Science and Technology Foundation to the Center for Advanced Technology—Electronic Imaging Systems, and by a grant from the Xerox Corporation.

A. Clock Scheduling

Previous research has established the possibility of permitting a synchronous digital system to operate at higher clock frequencies by exploiting non-zero clock skew within each local data path [4–6, 11–13]. Fishburn suggested an approach in [14] for computing a set of clock signal delays in a synchronous digital system. This set of delays is called a *clock schedule* which can be used to improve circuit performance while reducing the likelihood of creating race conditions.

A *sequentially-adjacent* pair of registers or a local data path (a pair of registers with only combinational logic between the registers) is shown in Figure 1a. The

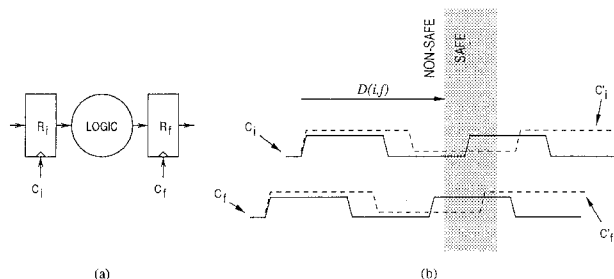


Fig. 1. Illustration of clock skew affecting the performance of a local data path. (a) A local data path. (b) Timing diagram of a local data path with positive clock skew.

clock signals C_i and C_f synchronize the registers R_i and R_f , respectively. These clock signals originate at the clock source and are delivered to the registers R_i and R_f by the clock distribution network with delays $T_d(i)$ and $T_d(f)$, respectively. Positive edge-triggered flip-flops are assumed.

As shown in Figure 1b the clock signals C_i and C_f may not arrive at their respective destination registers at the same time, thereby creating clock skew. Clock skew can be defined as follows:

Definition 1: Let R_i and R_f be a sequentially-adjacent pair of registers synchronized by the clock signals C_i and C_f , respectively. The clock skew between R_i and R_f is defined as

$$T_S(i, f) = T_d(i) - T_d(f), \quad (1)$$

where $T_d(i)$ and $T_d(f)$ are the delays of the signals, C_i and C_f , from the clock source to the registers R_i and R_f , respectively.

Returning to Figure 1, the rising edge of C_f must *not* arrive at R_f before the data has propagated through the logic and successfully latched within R_f . If the maximum propagation delay of the logic between the pair of registers is denoted by $D(i, f)$, C_f must clock R_f no earlier than $D(i, f)$ time after R_i has been clocked by C_i , *i.e.*, the clocking edge of C_f should occur after the non-safe (lightly-shaded) region in Figure 1b. If $T_d(i) \neq T_d(f)$, a non-zero clock

skew exists within this specific local data path. In this particular case, $T_d(i) > T_d(f)$, thus creating a positive clock skew. If the condition is not satisfied, a clock hazard has been created. This hazard is known as *zero clocking* [11, 14]. To remove this hazard, the minimum clock period must be increased.

Analogously, a situation may arise (not shown in Figure 1), where $T_d(i) < T_d(f)$, *i.e.*, the clock skew $T_S(i, f)$ is negative. If the minimum logic delay between a sequentially-adjacent pair of registers is denoted by $d(i, f)$, and $T_d(i) + d(i, f) < T_d(f)$, it is possible to clock the identical data signal through two registers with the same clock edge. This type of clock hazard is called *double clocking* [11, 14], and causes a catastrophic race condition to exist.

If the clock signal delays $T_d(i)$ and $T_d(f)$ are such that a clock hazard is created, some measures may be taken to eliminate this condition. For example, in Figure 1b a zero clocking condition is created since the clocking edge of C_f arrives before the data signal arrives. However, by increasing the length of the clock period—clock signals C'_i and C'_f in Figure 1b—it is possible to latch the data signal into R_f before C_f arrives. Increasing the clock period is equivalent to reducing the frequency of the clock signal, thereby decreasing system performance.

If the clock period is T_{CP} , the following two relationships must be satisfied for every local data path in the circuit in order to avoid these clock hazards:

$$T_S(i, f) = T_d(i) - T_d(f) \geq -d(i, f) \quad (2)$$

$$-T_S(i, f) = T_d(f) - T_d(i) \geq D(i, f) - T_{CP} \quad (3)$$

B. Organization of the Clock Distribution Network

The clock distribution network is typically organized as a tree structure [4, 11, 15] as illustrated in Figure 2. The unique clock source is the root of the

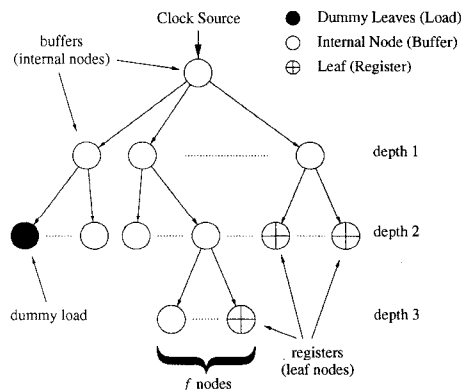


Fig. 2. Clock distribution network with *tree* structure.

tree. Every node of the tree branches to a fixed number of successor nodes. This number is called the

branching factor and is denoted by f . In the methodology presented in this paper, every node of the clock tree can either have f successors (*i.e.*, an *internal node*), or not have any successors at all (*i.e.*, an *external node*). In the latter case, the node is a *leaf* of the tree. The *branching depth* b_i of a specific node i is the number of branches on the unique path that exists between the root of the tree and a leaf node. Note that the depth b_i is also equal to the number of nodes on the path between the root of the tree and the node i , excluding i itself. In Figure 2, internal nodes are represented by empty circles and leaves are represented by crossed circles.

The physical interpretation of the different elements of the tree is as follows. All internal nodes represent buffer elements which drive either successive buffers in the clock tree or registers in the synchronous system—the leaves of the clock tree therefore correspond to the registers. The branches of the clock tree represent interconnections between the outputs of buffers and the inputs of other buffers (or registers). The black circles in Figure 2—called *dummy nodes*—are buffers which do not drive any successive nodes. These dummy nodes are necessary to ensure that the branching factor of each internal node is f . Note that the number of dummy nodes at any given depth of the clock tree can be an integer in the interval $\{0, \dots, f - 1\}$.

C. Timing Constraints and Problem Formulation

The timing model is based on the following assumption: *the signal propagation delay through a node, i.e., a buffer, is a constant and is denoted by Δ_b* . Note that the quantity Δ_b includes both the gate delay of the buffer and any interconnect delay of the branch immediately preceding this buffer. Therefore, the propagation delay from the clock source to any node i at depth b_i is $\delta_i = b_i \times \Delta_b$. If the node i is a leaf, *i.e.*, i is a register, the delay from the clock source to this register is the clock signal propagation delay $T_d(i)$, *i.e.*, $T_d(i) = \delta_i = b_i \times \Delta_b$. Substituting this expression into (2) and (3), the necessary conditions to avoid either clock hazard can be rewritten as follows:

$$T_S(i, f) = (b_i - b_f)\Delta_b \geq -d(i, f) \quad (4)$$

$$-T_S(i, f) = (b_f - b_i)\Delta_b \geq D(i, f) - T_{CP}. \quad (5)$$

Therefore, the problem of designing the topology of the clock distribution network can be formulated as an optimization problem of minimizing T_{CP} subject to the constraints (4) and (5).

The quantities b_i and b_f are integers, since they denote the number of branches (buffers) from the root of the clock tree to a particular leaf (*i.e.*, register). In the general case, this optimization problem can be described as a *mixed-integer* linear programming

problem (since T_{CP} can be any real, positive number), and is difficult to solve. However, if a fixed value for the clock period T_{CP} is chosen, the problem changes as follows. Given a value for T_{CP} , find a set of integers $\{b_1, b_2, \dots, b_i, \dots\}$ such that

$$\begin{aligned} (b_i - b_j)\Delta_b &\geq -d(i, j) \\ \text{and } (b_j - b_i)\Delta_b &\geq D(i, j) - T_{CP} \end{aligned} \quad (6)$$

for every sequentially-adjacent pair of registers (R_i, R_j) , or determine that no such set of integers exist.

Once (6) has been solved for a particular synchronous digital system, a clock tree topology such as the network shown in Figure 2 can be implemented. Each register R_i of the circuit receives its clock signal from a leaf of the clock tree at branching depth $b = b_i$, where b_i is the integer obtained from solving (6). Different clock tree structures can be designed by choosing different values of the branching factor f .

III. SOLUTION AND EXPERIMENTAL RESULTS

Leiserson and Saxe demonstrate in [16] that an algorithm exists for efficiently solving optimization problems such as represented by (6). The run time of this algorithm is $O(V(E + V) \log V)$, where V and E denote the number of registers and the number of sequentially-adjacent pairs of registers, respectively. This algorithm is used in this paper as part of the design methodology for constructing the topology of the clock tree.

A binary search of the feasible range for the clock period is performed to determine the minimum possible clock period such that (6) has a feasible solution. The boundary values of the feasible clock period to be searched consists of the interval from T_{min} to T_{max} . The values T_{min} and T_{max} are computed from information describing the short path and long path propagation delays of each local data path.

The algorithm has been implemented in a 3,300-line program written in the C++ high-level programming language. This program has been executed on the ISCAS'89 suite of benchmark circuits. A summary of the results for these benchmark circuits is shown in Table I. These results demonstrate that by applying the proposed algorithm to schedule the clock delays to each register, up to a 64% decrease in the minimum clock period can be achieved.

Once the clock scheduling has been computed, different implementations of the clock tree are possible for different values of the branching factor f . One possible implementation of the clock tree of the circuit s400 for $f = 3$ is shown in Figure 3.

TABLE I. EXPERIMENTAL RESULTS. For each ISCAS'89 circuit the number of registers, the bounds of the searchable clock period, the optimal clock period (T_{opt}), and the performance improvement (in per cent) are shown.

Circuit	Regs	T_{min}	T_{max}	T_{opt}	% Imp.
s1196	18	7.80	20.80	13.00	17%
s13207	669	60.40	85.60	60.45	29%
s1423	74	75.80	92.20	79.00	14%
s1488	6	31.00	32.20	31.00	4%
s15850	597	83.60	116.00	83.98	28%
s208.1	8	5.20	12.40	5.48	56%
s27	3	5.40	6.60	5.40	18%
s298	14	9.40	13.00	10.48	19%
s344	15	18.40	27.00	18.65	31%
s349	15	18.40	27.00	18.65	31%
s35932	1728	34.20	34.20	34.20	0%
s382	21	8.00	14.20	8.88	37%
s38417	1636	42.20	69.00	42.82	38%
s38584	1452	67.60	94.20	67.65	28%
s386	6	17.00	17.80	17.80	0%
s400	21	8.40	14.20	8.88	37%
s420.1	16	5.20	16.40	7.45	55%
s444	21	8.40	16.80	10.17	39%
s510	6	14.80	16.80	15.20	10%
s526	21	9.40	13.00	10.48	19%
s526n	21	9.40	13.00	10.48	19%
s5378	179	20.40	28.40	22.29	22%
s641	19	71.00	88.00	71.03	19%
s713	19	79.20	89.20	72.23	19%
s820	5	19.20	19.20	19.20	0%
s832	5	19.80	19.80	19.80	0%
s838.1	32	5.20	24.40	8.76	64%
s9234.1	211	54.20	75.80	54.24	28%
s9234	228	54.20	75.80	54.24	28%
s953	29	16.40	23.20	18.96	18%

IV. CONCLUDING REMARKS AND FUTURE WORK

The problem of synthesizing the topology of a buffered clock distribution network from a top-down clock skew schedule is examined in this paper. An integer linear-programming approach based on local timing information is presented for determining the clock skew schedule. Different forms of the clock distribution network are possible for different values of the branching factor f . The effects of f on the total area and depth of the clock tree is presently under investigation as well as adding the capability of varying f within a clock tree.

The methodology presented in this paper can be applied to the top-down design of synchronous digital systems. In this methodology, the buffered clock tree synthesis process is integrated with the clock scheduling process to achieve a minimum clock period.

REFERENCES

- [1] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley Publishing Company, 1990.
- [2] S. Bothra, B. Rogers, M. Kellam, and C. M. Osburn, "Analysis of the Effects of Scaling on Interconnect Delay in ULSI Circuits," *IEEE Transactions on Electron Devices*, Vol. ED-40, pp. 591-597, March 1993.

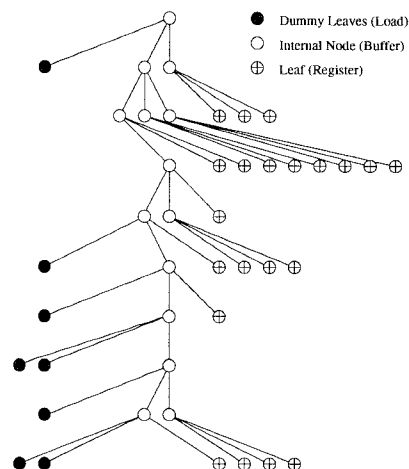


Fig. 3. Buffered clock tree for the benchmark circuit s400. Branching factor $f = 3$, total of 14 buffers and 21 registers.

- [3] W. J. Bowhill *et al.*, "Circuit Implementation of a 300-MHz 64-bit Second-generation CMOS Alpha CPU," *Digital Technical Journal*, Vol. 7, No. 1, pp. 100-118, 1995.
- [4] J. L. Neves and E. G. Friedman, "Topological Design of Clock Distribution Networks Based on Non-Zero Clock Skew Specification," *Proceedings of the 36th Midwest Symposium on Circuits and Systems*, pp. 468-471, August 1993.
- [5] J. G. Xi and W. W.-M. Dai, "Useful-Skew Clock Routing With Gate Sizing for Low Power Design," *Proceedings of the 33rd ACM/IEEE Design Automation Conference*, pp. 383-388, June 1996.
- [6] J. L. Neves and E. G. Friedman, "Design Methodology for Synthesizing Clock Distribution Networks Exploiting Non-Zero Localized Clock Skew," *IEEE Transactions on VLSI Systems*, Vol. VLSI-4, pp. 286-291, June 1996.
- [7] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, "Clock Routing for High-Performance ICs," *Proceedings of the 27th ACM/IEEE Design Automation Conference*, pp. 573-579, June 1990.
- [8] R.-S. Tsay, "An Exact Zero-Skew Clock Routing Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. CAD-12, pp. 242-249, February 1993.
- [9] N.-C. Chou and C.-K. Cheng, "On General Zero-Skew Clock Net Construction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. VLSI-3, pp. 141-146, March 1995.
- [10] N. Ito, H. Sugiyama, and T. Konno, "ChipPRISM: Clock Routing and Timing Analysis for High-Performance CMOS VLSI Chips," *Fujitsu Scientific and Technical Journal*, Vol. 31, pp. 180-187, December 1995.
- [11] E. G. Friedman, *Clock Distribution Networks in VLSI Circuits and Systems*. IEEE Press, 1995.
- [12] E. G. Friedman, "The Application of Localized Clock Distribution Design to Improving the Performance of Retimed Sequential Circuits," *Proceedings of the IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 12-17, December 1992.
- [13] J. L. Neves and E. G. Friedman, "Optimal Clock Skew Scheduling Tolerant to Process Variations," *Proceedings of the 33rd ACM/IEEE Design Automation Conference*, pp. 623-628, June 1996.
- [14] J. P. Fishburn, "Clock Skew Optimization," *IEEE Transactions on Computers*, Vol. C-39, pp. 945-951, July 1990.
- [15] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. MIT Press, 1989.
- [16] C. E. Leiserson and J. B. Saxe, "A Mixed-Integer Linear Programming Problem Which is Efficiently Solvable," *Journal of Algorithms*, Vol. 9, pp. 114-128, March 1988.