Research in High Speed, Low Power Synchronous Digital and Mixed-Signal Systems

Eby G. Friedman

High Performance VLSI/IC Design and Analysis Laboratory
Department of Electrical Engineering
University of Rochester
Rochester, New York 14627

friedman@ee.rochester.edu

Abstract - The intention of this paper is to summarize research currently under development in the High Performance VLSI/IC Design and Analysis Laboratory within the Department of Electrical Engineering at the University of Rochester. Research in a variety of related fields dealing with high performance microelectronic design and analysis is reviewed such as a) the automated synthesis of high speed and low power clock distribution networks, b) circuit models and algorithms for the practical application of retiming, c) CMOS circuit design techniques that optimize and/or tradeoff speed, area, power, and reliability, and d) design techniques to minimize noise effects between the analog and digital portions of mixed-signal VLSI-based systems. These areas represent some of the primary topics currently under development in this laboratory to advance the nation's capability in building next generation VLSI/ULSI-complexity high performance synchronous digital and mixed-signal systems.

I. Significance of synchronous digital systems

The vast majority of modern VLSI/ULSI-based systems utilize fully synchronous timing, requiring a distributed clock signal to synchronize events. This clock signal functions as a system-wide time reference, coordinating the temporal operation of a digital synchronous circuit. Conceptually, arbitrarily complex VLSI circuits can be assembled from pieces of combinational logic, each of which performing fairly simple arithmetic operations. Increasingly sophisticated timing, however, is required when hundreds of thousands of these elementary operations are executed in a particular sequence controlled by the clock signal and specified by the VLSI algorithm being implemented. Furthermore, in the design of a synchronous VLSI system, storage registers are inserted between the blocks of logic. These registers are synchronized by the clock signal and functionally align in time the data signals at the inputs of the logic, thereby increasing the system-wide clock frequency. This process of inserting registers is called pipelining. The process of automated pipelining in which the latency is held constant is called retiming

The standard methodology for designing complex VLSI systems is top-down. The design process begins at the topmost, behavioral level of abstraction, where the required system functionality is initially specified. During each subsequent design phase, the specifications at the current level of abstraction are

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. DAAH04-93-G-0323, and by a grant from the Xerox Corporation.

transcribed into the structural properties of the next lower level in the design hierarchy until a mask layout is generated and the circuit is submitted for fabrication. As new semiconductor technologies constantly push against the limit of circuit performance and complexity, a variety of tools is introduced to handle these correspondingly more complex design problems, permitting the development of higher performance (both high speed and low power) VLSI/ULSI complexity circuits.

Furthermore, a large portion of the problems arising in VLSI design are intractable from a computational perspective. These design tools often trade off quality and/or performance of the final product with the computational efficiency of the design algorithm, thereby not necessarily produce optimally performing circuits. With ten million transistor circuits operating at clock speeds of many hundreds of megahertz, the design complexity of these systems is growing exponentially.

In response to these design bottlenecks, a variety of design techniques, optimization strategies, and design methodologies are under investigation within this laboratory. These research efforts all have the objective of speeding up design turnaround time and improving circuit performance without sacrificing quality. The focus is to target those circuit issues that limit the performance (both speed and power) in synchronous digital and mixed-signal systems. These efforts are comprised of a number of interrelated research objectives which are summarized in this paper. The automated synthesis of high performance clock distribution networks including clock scheduling is described in section 2. For fully synchronous digital systems, an optimization technique that has attracted significant attention recently is retiming. This topic is discussed in section 3. Each of these systems must be physically implemented in a semiconductor technology, where CMOS is currently the most common technology of choice. Every application requires a different tradeoff among a variety of design criteria. Research in evaluating and exploiting these speed/area/power design criteria in CMOS circuits is discussed in section 4. Recent effort has focused on integrating both digital and analog elements of a system onto a single substrate. Often these analog and digital signals interact, degrading the performance of these mixed-signal systems. Recent research within this laboratory has focused on mitigating the effects of substrate noise coupling in mixed-signal systems. This effort is briefly reviewed in section 5. Finally, some general summarizing comments are offered in section 6.

2. Synthesis of Clock Distribution Networks based on Non-Zero Clock Skew Specifications

The existence of registers in synchronous digital systems requires a temporal reference signal to control the sequence of events being executed by the system. This reference signal must be distributed throughout the circuit and is typically called the clock distribution network. The design of the clock distribution network is one of the limiting factors in performance and reliability of a synchronous digital system and is, therefore, worthy of significant investigation.

These clock signals typically do not arrive at the input of the registers at the same time, creating clock skew between sequentially-adjacent registers. This clock skew is manifested by a lead/lag relationship

between the clock signals that control the signal flow in a local data path. The existence of clock skew is caused by differences in propagation delay from the clock source to the destination registers due to asymmetric clock signal paths. Typically, the clock skew is minimized during the design of the clock distribution network, although localized clock skew can be used to increase the system-wide clock frequency [1,2]. A new methodology is under development within this laboratory for the design of clock distribution networks with distributed buffers which exploits non-zero localized clock skew to increase circuit speed. This design methodology consists of four phases [3-7]. These phases are:

1. Optimal clock scheduling phase to determine the minimum set of clock skew values that maximize speed performance while avoiding all race conditions.

2. Topological design phase to determine the hierarchical structure of the clock tree and to determine the minimum delay values of each branch such that the non-zero clock skew specifications are satisfied.

3. Circuit design phase to design the circuit structures that accurately emulate the delay values assigned to each branch of the clock tree.

4. Layout design phase to determine the physical position of the CMOS buffers and to determine the routing of the clock tree.

This research project has focused primarily on the first three phases. The optimal clock skew scheduling phase is discussed in greater detail in section 2.1. In phase two, the topology of the clock distribution network is implemented and delays values are assigned to each branch of the clock tree [3]. In the third phase, a design strategy has been successfully developed to determine the geometric size and output load of each CMOS buffer. Furthermore, an algorithm has been developed to determine the minimum number of CMOS buffers of each branch of the clock tree such that the polarity of the clock signal at the clock input of an edge-triggered flip-flop is preserved. The circuit synthesis technique that implements the branch delay values is highly accurate, producing clock distribution networks that agree within 3% of simulated experiments [5]. This clock distribution network circuit synthesis phase has also been extended to minimize the power dissipated within the clock tree [7].

2.1 Clock Skew Scheduling

In submicrometer technologies, the requirement of equal clock delay has become increasingly difficult to meet. Higher global interconnect delays, parasitic impedances, and process parameter variations are among the principal factors that have lead to non-uniformity among the clock signal delays. These factors create non-deterministic differences in the clock skew that may drastically limit the maximum frequency of operation of the synchronous digital circuit. Nevertheless, localized non-zero clock skew may be used to improve the overall performance of a synchronous VLSI system. The use of non-zero clock skew to improve the performance of VLSI systems is discussed in section 2.1.1.

Furthermore, techniques have been developed that tolerate variations of the delays within the clock distribution network due to process parameters variations [8-10]. These techniques are summarized in section 2.1.2.

2.2.1 Targeted clock delays

A sequentially-adjacent pair of registers, *i.e.*, a pair of registers with only combinational logic between them is shown in Figure 1. The clock signals C_i and C_f which synchronize the registers R_i and R_f, respectively, are delayed by the clock distribution network by T_{CDi} and T_{CDf} respectively. The difference in arrival time of the clock signals between a pair of sequentially-adjacent registers is $T_{Skew}=T_{CDi}-T_{CDf}$ [1,2].

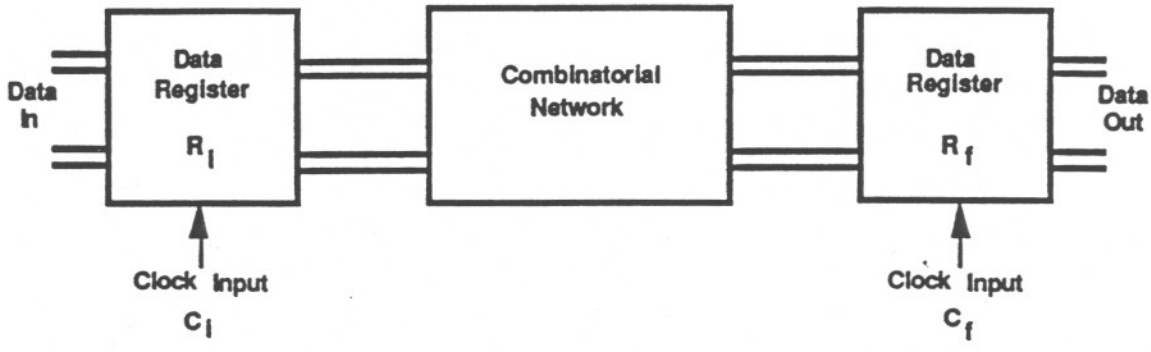

Figure 1: A local data path composed of a sequentially-adjacent pair of registers

The operation of the circuit shown in Figure 1 is as follows. At the rising edge of the clock signal, a new data signal begins to propagate through the logic. The data signal must arrive at the input of R_f and successfully latch into R_f before the arrival of the next clocking edge at C_f. The latching characteristics determine the set-up time $T_{Set\text{-}up}$ and hold time T_{Hold} of the registers, which describe the relative temporal relationships of the clock and data signals at the input of R_f [1,2,11]. The time required for the data signal to propagate from R_i upon the arrival of C_i through the logic and registers and successfully latch into R_f before the arrival of C_f is the minimum clock period T_{CP}. If the clocks C_i and C_f are perfectly aligned in time (*i.e.*, zero clock skew), the minimum clock period (or maximum clock frequency) of the circuit is only constrained by the data path delay.

The signals C_i and C_f may arrive at the destination register with different delays. If C_f arrives at R_f before C_i arrives at R_i, a positive clock skew is created. The rising edge of C_f must not arrive while the data is still propagating through the logic. It is, however, safe to clock R_f once the data signal has propagated through the logic. With positive clock skew, T_{CP} may not be sufficiently long for the data signal to propagate through the logic and latch within R_f. This situation can be corrected by increasing the clock period by an

amount represented by the positive clock skew, thereby decreasing the maximum system-wide clock frequency.

Negative clock skew has a different effect on circuit performance. Negative clock skew occurs when C_i leads C_f, essentially giving the data signal at the output of R_i a "head start" [2]. A certain amount of negative clock skew can be exploited to increase the circuit performance by "stealing" time from the fast data paths, shifting the time into the slower data paths and relieving the timing constraints on these slow data paths [2]. This characteristic of negative clock skew is one of the primary reasons for scheduling the clock signals. Excessively large amounts of negative clock skew, however, may have an adverse effect, creating a race condition known as double clocking [12]. In the latter case, the data signal which is latched within the final register R_f may be overwritten by a new data signal and thereby lost. These characteristics of the clock skew suggest the local values of clock skew must be chosen to maximize the speed of a system while removing any race conditions from the system. This approach, called clock scheduling [2,8-10,12], exploits non-zero clock skew rather than minimizes the clock skew. Since clock scheduling does not encompass any architectural changes, the function of the original circuit is preserved. From a computational perspective, clock scheduling can be efficiently handled as a linear programming problem. Such a formulation is based on a simplified sequential circuit model and was first introduced by Fishburn in 1990 [12].

2.1.2 Clock skew tolerant to process parameters variations

As discussed in section 2.1.1, non-zero positive clock skew may seriously degrade the performance of a circuit by limiting its maximum operating frequency. Unintentional factors, such as process parameter variations, can further increase the amount of non-zero clock skew, as well as create race conditions independent of clock frequency, thereby leading to circuit failure. Therefore, the design of process insensitive clock distribution networks is a critical phase in the design of high speed, high performance synchronous digital circuits.

To further illustrate the significance of process parameter variations, consider the local data path shown in Figure 1. A local data path L_{ij} is a set of sequentially-adjacent pair of registers, R_i and R_j. Each local data path can be characterized by a minimum and maximum delay (T_{PDmin} and T_{PDmax}) through the combinational logic block and registers. A region of valid clock skew for each local data path, called the permissible range $Pl(L_{ij})$ [8-10], can be described as shown in Figure 2. The bounds of $Pl(L_{ij})$ are determined from the timing constraints of each local data path for a given clock period T_{CP}. These local constraints ensure that the data signal is correctly latched into the registers of a local data path, and can be expressed as

$$T_{Skew}(L_{ij}) > T_{Holdj} - T_{PD(min)} + \zeta_{ij}\,, \tag{1}$$

$$T_{CP} > T_{Skew}(L_{ij}) + T_{PD(max)}, \tag{2}$$

where ζ_{ij} is a safety term to provide some margin in a local data path against race conditions due to process parameter variations. In particular, (1) prevents latching the incorrect data signal into R_j by the clock pulse that latched the same data into R_i, thereby avoiding a race condition. Constraint (2) guarantees that the data signal latched in R_i is latched into R_j by the following clock pulse. Finally, the width of a permissible range is defined as the difference between the maximum ($T_{Skewij(max)}$) and the minimum ($T_{Skewij(min)}$) clock skew.

Figure 2: Permissible range of the clock skew of a local data path

Satisfying the clock skew constraints of each individual local data path does not guarantee that the clock skew between the initial and final registers of a global data path is satisfied, particularly if there are multiple parallel and feedback paths between these two registers. Since any two registers connected by more than one global data path are each driven by a single clock path, the clock skew between these two registers is unique and the permissible range of every path connecting the two registers must contain this clock skew value to ensure that the circuit will function correctly.

This clock scheduling design system has been demonstrated on a variety of synchronous circuits, reducing the minimum clock period by up to 42% by exploiting intentional non-zero clock skew. Furthermore, an 18% improvement in clock frequency with up to a 30% variation in the nominal clock skew, and a 33% improvement in clock frequency with up to an 18% variation in the nominal clock skew have been demonstrated for a set of standard benchmark circuits [6,8-10].

3. Retiming with localized clock distribution, interconnect, and register delays

Retiming is a powerful architectural-level transformation technique that optimizes the performance of VLSI systems by relocating the storage elements within these systems without affecting the functionality of the original circuit. With retiming the temporal distribution of the data path delays is more balanced, yielding a circuit with a higher maximum clock frequency than the original circuit. In the example shown in Figure 3, a combinational logic block v is retimed with a lag of 1 by removing one storage element from each output wire of v and by inserting one storage element onto each input wire of v. This transformation amounts to shifting a stage of storage elements from the outputs of v to the inputs of v. Since storage elements are inserted at the input wires of v, the three inputs of v are delayed by one clock cycle. The result of the computation at v does not change, however, because the three inputs still arrive within the same clock cycle. Therefore, the functionality of v has not been affected by the relocation of the storage elements. Consequently, when retiming is applied to every combinational logic block of a circuit, the functionality of the circuit is maintained.

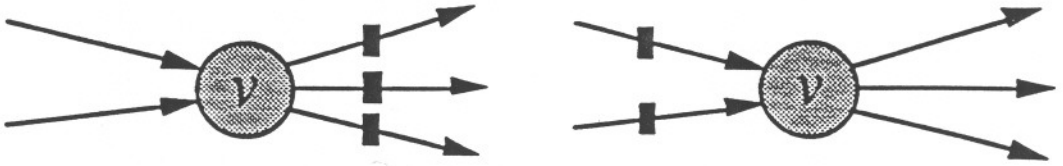

Figure 3: Illustration of retiming for a single block of combinational logic v. The circle represents a block of combinational logic and the rectangles represent storage elements. Retiming moves storage elements across the logic without affecting the functionality of the VLSI system.

A linear programming formulation of retiming for reducing the clock period of an edge-triggered system was first presented by Leiserson and Saxe in [13]. By assuming zero clock skew, Leiserson and Saxe exploited the structure of their linear program to design an $O(VE)$-time algorithm for retiming an edge-triggered system to achieve a given clock period, where V is the number of combinational logic blocks and E is the number of wires in the system. Leiserson and Saxe also provided an $O(VE \lg V)$-time algorithm for retiming to achieve the minimum possible clock period.

Existing research into retiming has so far dealt with highly idealized circuit models that are unsuitable for practical applications. With certain exceptions, this existing research has failed to address important electrical issues that are crucial to contemporary VLSI circuit design. These issues represent a variety of problems inherent to emerging submicrometer VLSI/ULSI technologies. Therefore, in order to permit the successful practical application of retiming, any retiming algorithm must consider these electrical problems, which include RC interconnect delay, variable logic and register delay, process parameter variations, physical area, and power dissipation.

Previous results in the area of retiming algorithms ignore non-zero clock skew and variable register and interconnect delay. Data paths are defined in these algorithms from vertex-to-vertex and the delay of these paths are used to determine those critical paths that affect circuit performance. Only long paths are considered, since without considering clock skew, paths with negative delay (short paths) cannot be analyzed. Therefore, previous work in the field of retiming has assumed that registers are ideal, the clock distribution network is assumed to have zero clock skew, the registers are assumed to have zero or constant delay, and the interconnect delay between the registers and the logic elements is assumed to be negligible.

In an actual synchronous circuit, however, all of these electrical delay components can significantly affect circuit performance. Without including these delay components, these earlier retiming algorithms have not been sufficiently accurate to permit their use in synthesizing practical high speed circuits. For this reason, clock distribution, variable register, and interconnect delay have been integrated into the retiming process in order to ensure that retiming becomes a practical and useful element of the overall VLSI design methodology.

An algorithm *RETSAM* has been developed within this laboratory in which electrical characteristics are attached to the edges of the graph representing the circuit, thereby incorporating delay components [14,15]. These attached values are called Register Electrical Characteristics or RECs. Attaching RECs to the edges permits redefining the path delays to be from edge-to-edge rather than vertex-to-vertex, as in previous retiming algorithms. The algorithm *RETSAM* uses a matrix called the Sequential Adjacency Matrix (SAM) which contains all edge-to-edge path delays. Four different types of timing constraints are derived from this matrix: long path constraints, short path constraints, negative edge weight constraints, and internal long path constraints. Race conditions in the circuit appear as negative entries in the SAM. A binary search is performed among all possible clock periods and the minimum clock period is determined. A circuit which operates at the minimum possible clock period and contains no race conditions is determined with this retiming algorithm. With *RETSAM*, race conditions in the graph are detected on a path-by-path basis and removed as a condition imposed on the retiming process. This result cannot be assumed in other retiming algorithms and demonstrates an important capability of this retiming algorithm. Furthermore, a set of monotonicity constraints has been developed that can be imposed on the REC values so as to permit the use of existing linear programming methods [16]. Thus, a practical and useful retiming algorithm that operates in polynomial time has been developed.

In summary, a new retiming algorithm has been developed which considers the effects of variable clock distribution, register, and interconnect delay on local data paths. This algorithm represents a significant generalization of existing retiming algorithms, permitting the use of retiming for the automated synthesis of higher speed, more reliable pipelined digital systems. This capability with the clock distribution network synthesis system described in section 2 provides an integrated pipelining methodology composed of both retiming and clock distribution network design for the automated synthesis of high performance synchronous digital systems.

4. CMOS Circuit Structures for Optimizing Speed/Power/Area/Reliability Tradeoffs

Another related research area is the field of specialized structures for high performance CMOS integrated circuits. Recently, three specific structures have been studied: tapered serial MOSFET chains, CMOS tapered buffers, and CMOS repeaters. All of these structures are common to standard VLSI circuits and are of fundamental importance to the design of fast, low power integrated circuits.

4.1 Tapered Serial MOSFET Chains

Many CMOS logic structures contain chains of MOSFETs serially connected between a power supply and the output of the subcircuit. These serially connected MOSFETs are a major source of both delay and power dissipation. Exponential tapering of serially connected MOSFET chains had previously been presented qualitatively as a method to reduce the delay of these chains. With exponential tapering, the channel width of a transistor in a serial chain is a constant multiple less than unity of the channel width of

the transistor to which its source terminal is connected. In this manner, the ratio of channel widths between any two adjacent transistors along the chain remains fixed. This research into tapering has concentrated on explaining the mechanisms by which tapering works, and exploring the power dissipation characteristics of tapering. Tapered chains of serially connected MOSFETs have been shown to simultaneously decrease power dissipation, physical area, and under certain load circumstances, circuit delay as compared to that of constant width serially connected MOSFET circuits [17].

Specifically, this research has led to the classification of serially connected MOSFET chains into three categories, each providing specific tradeoffs among speed, area, and power dissipation. The first category includes those serial chains for which tapering reduces propagation delay, short-circuit power dissipation, dynamic power dissipation, and physical area. In the second category, similar short-circuit and dynamic power dissipation reductions occur. However, in this category, the propagation delay is somewhat increased. An important phenomenon which distinguishes tapered serial chains in this category from untapered serial chains of similar delay is the lack of signal degradation in the tapered chains. That is, by tapering a chain of Category 2 MOSFET transistors, the 50% delay is increased, but the slope of the output waveform is also increased, making the output waveform appear more rectangular in shape. It is this effect which reduces the short-circuit power dissipation in the stage following the serial chain, an added advantage over that of an untapered chain of similar delay. The third category includes those tapered serial chains in which delay is increased, and the quality of the output waveform is diminished. There is no particular advantage to tapering those serial chains which fall into this category.

An automated design system for exploiting the speed and power dissipation advantages of tapered serial chains has been developed as part of this research effort. This system generates SPICE circuit simulation files and automatically generates a physical layout of the tapered serial MOSFET chain. Finally, a test circuit was successfully fabricated, experimentally demonstrating the power dissipation characteristics of tapering serial MOSFET chains.

4.2 Cascaded CMOS Buffers

Methodologies have also been developed for the design of CMOS cascaded buffers. Specifically, a unified tapered buffer design methodology has been developed which simultaneously considers circuit speed, power dissipation, physical area, and hot-carrier reliability [18]. A methodology has also been developed for tapered buffer design which considers the effects of local interconnect capacitance [19].

The CMOS tapered buffer is a common circuit structure in modern VLSI-based systems. In CMOS integrated circuits, large capacitive loads occur both on-chip where high localized fan-out is often encountered, and off-chip, where highly capacitive chip-to-chip communication lines exist. In order to drive these large capacitive loads at high speeds, buffer circuits are required which must source and sink relatively large currents quickly, while not degrading the performance of previous stages. In CMOS, a tapered buffer is often used to perform this task.

Many different approaches to tapered buffer design have been described in the literature. The most commonly addressed criteria in tapered buffer design are propagation delay, power dissipation, physical area, and circuit reliability. However, while many disparate approaches have been developed to deal with these performance issues, no single design methodology exists which considers all four of these criteria simultaneously. Such a methodology has been developed.

The development of this unified design methodology has progressed in two phases. In the first phase, expressions for each of the four performance criteria have been developed using a single nomenclature, allowing, in the second phase, for the unification of these expressions into a single methodology.

As the physical geometries of CMOS processes continue to decrease, the presence of short-channel effects has become unavoidable in modern technologies. Thus, it is necessary to consider these short-channel effects when developing analytical performance models. Previously developed performance expressions described in the literature have been based upon long-channel models. In this work, the first analytical expressions have been developed for propagation delay and power dissipation of tapered buffers based upon a short-channel transistor model, specifically the Sakurai-Newton alpha-power model [20]. The split-capacitor model [21] is also used to incorporate the parasitic capacitance associated with each buffer stage. An important finding of this research is that while short-channel effects change the absolute delay and power dissipation of a tapered buffer, these effects do not change the relative values of these performance criteria as functions of tapering factor. Thus, much of the previous work in the field, which was developed for long-channel transistors, is equally applicable to tapered buffers with short-channel transistors.

An analytical expression for the hot-carrier reliability of a tapered buffer system has been developed based upon the split-capacitor model. This is a relatively new design criteria, and the only previous expression describing the hot-carrier reliability of a tapered buffer is based upon a less accurate single capacitor model.

These expressions have been unified into a single, efficient design methodology permitting an application-specific optimal buffer implementation based on a simple look-up table. These look-up tables are generated utilizing weighted products of the previously described performance expressions. A methodology for applying the look-up tables in conjunction with system performance constraints has also been developed. This methodology determines the optimal design, considering all four performance criteria. Thus, previously disparate design approaches for CMOS tapered buffers have been integrated into a unified tapered buffer design methodology.

Furthermore, a tapered buffer design methodology which considers the effects of local stage-to-stage interconnect capacitance has been developed. Previous tapered buffer methodologies presented in the literature consider the effects of stage-to-stage interconnect capacitance to be negligible. However, in many cases, it is both not negligible and nonlinear with respect to the geometric size of the buffer stages. For instance, in semi-custom integrated circuits where cells are placed in rows and automated routing occurs within channels, long stage-to-stage interconnect distances, and thus large interconnect capacitances, may

develop. Initial, smaller stages of the tapered buffer system are particularly sensitive to this local interconnect capacitance as it may be similar in magnitude to the output and input capacitances associated with these smaller buffer stages.

This tapered buffer methodology is based upon maintaining the capacitance to current ratio constant for every stage of the buffer system. This has been shown to be delay optimal in a tapered buffer system, and standard tapered buffer approaches also maintain this ratio. However, standard approaches ignore the stage-to-stage local interconnect capacitance. This method, denoted as Constant Capacitance-to-Current Ratio Tapering (C^3RT), includes the interconnect capacitances in determining the optimal buffer implementation. The C^3RT methodology results in tapered buffers with reduced propagation delay, power dissipation, and active area requirements as compared with conventional Fixed Tapered (FT) approaches. Delay reductions of 1%-2%, power dissipation reductions of up to 25%, and active area reductions of up to 45% have been demonstrated.

4.3 CMOS Repeater Design

As the die size of CMOS integrated circuits continues to increase, interconnections have become increasingly significant. With a linear increase in length, interconnect delay increases quadratically due to a linear increase in both interconnect resistance and capacitance. Large interconnect loads not only affect performance but also cause excess power to be dissipated. A large RC load degrades the waveform shape, dissipating excessive short-circuit power in the following stages loading a CMOS logic gate.

An analytical expression for the transient response of a CMOS inverter driving a lumped RC load has been developed. Sakurai's alpha-power law [20] is used to describe the circuit operation of the CMOS transistors rather than the classical Shichman-Hodges model. The alpha-power law model considers short channel behavior, permitting improved accuracy and generality in the delay and power expressions.

The model can be used to design and analyze those CMOS inverters that drive a large RC load when considering both speed and power. Expressions have been developed for estimating the propagation delay and transition time which exhibit less than 27% discrepancy from SPICE for a wide variety of RC loads. Expressions have also been developed for modeling the short-circuit power dissipation of a CMOS inverter driving a resistive-capacitive interconnect line which are accurate to within 15% of SPICE for most practical loads [22,23]. Therefore, due to the simplicity and accuracy of these expressions, the delay and power characteristics of a CMOS inverter driving a high impedance RC interconnect line can be efficiently estimated.

5. Mixed Signal Systems

As chip dimensions decrease and therefore the spacing between devices become smaller, noise coupling between on-chip components will increase, degrading signal integrity and creating reliability problems. Furthermore, as chip die sizes increase, parasitic coupling capacitances will also increase.

greatly affecting signal quality, particularly for increasingly higher speed signals. Finally, mixed on-chip digital and analog signals are particularly sensitive to crosstalk induced noise coupling. These technology trends indicate that noise coupling (or crosstalk) between interconnect will become a catastrophic reliability problem in near-term integrated circuits unless noise mitigation strategies are used.

Various failure mechanisms are possible. Temporary signal transitions (or glitches) incorrectly representing the data signal may latch into bistable registers and store incorrect signals, creating short-term (since new data will overwrite the incorrect data) reliability problems. High speed digital signals may also couple into adjacent analog signals, distorting these sensitive analog signals. Low output impedance buffers driving high impedance nodes are particularly susceptible to noise induced crosstalk. Therefore, in those systems in which high current drivers may be physically close to the control logic and storage elements, coupling induced crosstalk has the potential to severely affect circuit reliability, performance, and yield.

The strategy being used to investigate these substrate current noise effects is composed of a number of sequential steps. The physical mechanisms inducing this noise coupling have been studied. Criteria to measure these potential problems have been devised, followed by the design and manufacture of a number of experimental test circuits to characterize this coupling behavior. Test data are currently being extracted from these manufactured test circuits, permitting the development of empirical design rules for both physical layout and for simulating these circuits so as to minimize any significant noise coupling. Finally, design techniques will be devised to reduce noise related crosstalk below some catastrophic threshold.

Circuit level techniques will be developed to minimize the effects of coupling crosstalk between signal lines. A wide variety of techniques are being considered to isolate the high current MOSFETs from the storage elements. Examples of possible approaches include: 1) substrate decoupling capacitors to supply current to the switching nodes, 2) multiple substrate connections to ground to isolate coupling current paths within the substrate, thereby decreasing any $L\ di/dt$ noise, 3) multiple power/ground pins and bond connections to minimize inductances, 4) tailored driver turn-on characteristics to minimize the magnitude of the coupling signals, and 5) differential circuit styles with feedback to compensate for transient noise. Issues such as area penalty and implementation technology are primary criteria in these noise mitigation strategies.

6. Summary and Conclusions

This paper is intended to summarize some of the research efforts currently under development within the High Performance VLSI/IC Design and Analysis Laboratory within the Department of Electrical Engineering at the University of Rochester. The topics reviewed within this paper focus primarily on synchronous digital and mixed-signal VLSI-based systems. Emphasis has been placed on developing practical design techniques and methodologies with direct application to both high speed and low power applications. Target application areas include high complexity, high performance microprocessors, ultra-low power portable wireless communication devices, and signal and image processors.

References

[1] E. G. Friedman, "Clock Distribution Design in VLSI Circuits - an Overview," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 1475-1478, May 1993.

[2] E. G. Friedman, *Clock Distribution Networks in VLSI Circuits and Systems*, Piscataway, New Jersey:IEEE Press, 1995.

[3] J. L. Neves and E. G. Friedman, "Topological Design of Clock Distribution Networks based on Non-Zero Clock Skew Specifications," *Proceedings of the IEEE 36th Midwest Symposium on Circuits and Systems*, pp. 468-471, August 1993.

[4] J. L. Neves and E. G. Friedman, "Synthesizing Distributed Buffer Clock Trees for High Performance ASICs," *Proceedings of the IEEE ASIC Conference*, pp. 126-129, September 1994.

[5] J. L. Neves and E. G. Friedman, "Circuit Synthesis of Clock Distribution Networks based on Non-Zero Clock Skew," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 4.175-4.179, May 1994.

[6] J. L. Neves and E. G. Friedman, "Design Methodology for Synthesizing Clock Distribution Networks Exploiting Non-Zero Clock Skew," *IEEE Transactions on VLSI Systems*, Vol. VLSI-4, No. 2, June 1996.

[7] J. L. Neves and E. G. Friedman, "Minimizing Power Dissipation in Non-Zero Skew-based Clock Distribution Networks," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1576-1579, May 1995.

[8] J. L. Neves and E. G. Friedman, "Buffered Clock Tree Synthesis with Optimal Clock Skew Scheduling for Reduced Sensitivity to Process Parameter Variations," *Proceedings of the TAU'95 ACM/SIGDA International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pp. 131-141, November 1995.

[9] J. L. Neves and E. G. Friedman, "Reduced Sensitivity of Clock Skew Scheduling to Technology Variations," *Proceedings of the Fifth ACM/SIGDA Physical Design Workshop*, pp. 241-248, April 1996.

[10] J. L. Neves and E. G. Friedman, "Optimal Clock Skew Scheduling Tolerant to Process Variations," *Proceedings of the ACM/IEEE Design Automation Conference*, June 1996.

[11] E. G. Friedman, "Latching Characteristics of a CMOS Bistable Register," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. CAS-40, No. 12, pp. 902-908, December 1993.

[12] J. P. Fishburn, "Clock Skew Optimization," *IEEE Transactions on Computers*, Vol. C-39, No. 7, pp. 945-951, July 1990.

[13] C. E. Leiserson and J. B. Saxe, "Retiming Synchronous Circuitry, *Algorithmica*, Vol. 6, pp. 5-35, January 1991.

[14] T. Soyata, E. G. Friedman, and J. H. Mulligan, Jr., "Integration of Clock Skew and Register Delays into a Retiming Algorithm," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 1483-1486, May 1993.

[15] T. Soyata and E. G. Friedman, "Retiming with Non-Zero Clock Skew, Variable Register, and Interconnect Delay," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 234-241, November 1994.

[16] T. Soyata, E. G. Friedman, and J. H. Mulligan, Jr., "Monotonicity Constraints on Path Delays for Efficient Retiming with Localized Clock Skew and Variable Register Delay," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1748-1751, May 1995.

[17] B. S. Cherkauer and E. G. Friedman, "Channel Width Tapering of Serially Connected MOSFETs with Emphasis on Power Dissipation," *IEEE Transactions on VLSI Systems*, Vol. VLSI-2, No. 1, pp. 100-114, March 1994.

[18] B. S. Cherkauer and E. G. Friedman, "A Unified Design Methodology for CMOS Tapered Buffers," *IEEE Transactions on VLSI Systems*, Vol. VLSI-3, No. 1, pp. 99-111, March 1995.

[19] B. S. Cherkauer and E. G. Friedman, "Design of Tapered Buffers with Local Interconnect Capacitance," *IEEE Journal of Solid-State Circuits*, Vol. SC-30, No. 2, pp. 151-155, February 1995.

[20] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Application to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid-State Circuits*, Vol. SC-25, No. 2, pp. 584-594, April 1990.

[21] N. C. Li, G. L. Haviland, and A. A. Tuszynski," *IEEE Journal of Solid-State Circuits*, Vol. SC-25, No. 4, pp. 1005-1008, August 1990.

[22] V. Adler and E. G. Friedman, "Delay and Power Expressions for a CMOS Inverter Driving a Resistive-Capacitive Load," *Proceedings of the IEEE Symposium on Circuits and Systems*, May 1996.

[23] V. Adler and E. G. Friedman, "Delay and Power Expressions for Short-Channel CMOS Inverters Driving Resistive Interconnect," *Proceedings of the Workshop on Academic Electronics in New York State*, June 1996.