The Application of Localized Clock Distribution Design to Improving the Performance of Retimed Sequential Circuits *

E G Friedman
Department of Electrical Engineering
University of Rochester, New York, USA

Clock distribution networks synchronize the flow of data signals between data paths and the design of these networks can significantly affect system performance. By exploiting specific lead/lag clock waveform relationships, the delay of the worst case paths can be minimized by shifting time from sequentially adjacent less critical paths. Thus, as retiming is performed on pipeline registers by shifting their temporal locations, local clock distribution networks are used to tune the effective path delays of each local data path. Thus, retiming can be considered a two-phase process, logical partitioning for gross temporal allocation followed by local clock distribution design. By applying localized clock distribution design to the critical path delays of a large system, both maximum and minimum delay paths can be adjusted such that these paths will be of approximately equal delay, improving both system performance and reliability and permitting the realization of higher performance retimed circuits.

1. INTRODUCTION

Retiming is a behavioral synthesis methodology for improving synchronous performance in VLSI-based digital systems by automating the process of pipelining for parallel concurrency such that logic elements are shifted across sequentially adjacent registers so as to minimize the delay of the critical worst case paths [1-3]. In this synthesis technique, the total latency is maintained constant; the objective is to choose an optimal temporal location for each of the pipeline registers so as to minimize the maximum clock period of any local data path. Thus, the system operating clock frequency is maximized by adjusting the relative temporal placement of the pipeline registers. The approach presented in this paper integrates localized clock distribution design with retiming in order to improve overall synchronous performance as well as provide greater temporal resolution of the retiming process.

In presenting these results, the paper is composed of five principal sections. A brief overview of a synchronous system and its component parts is presented in section 2. The nature of the clock distribution network directly affects the performance and reliable operation of a synchronous digital system. Minimum and maximum constraint relationships as well as design techniques to improve performance are described in Section 3. In Section 4, latency and clock frequency relationships are developed in terms of the circuit and timing characteristics of a data path. Practical design equations are provided for analyzing the performance of retimed circuits. An example of applying localized clock distribution design to a pipelined system is provided in section 5 and some conclusions are presented in section 6.

*This research was supported in part by NSF Grant MIP-9208165

2. HIGH PERFORMANCE SYNCHRONOUS DATA PATHS

A synchronous digital system is composed of data paths in which data are moved from a register through some logic functions and into a second register. The synchronization of the data flow between the initial and final registers is typically coordinated by a single control signal, commonly called the clock signal. Thus, a synchronous digital system, as depicted in Figure 1, is composed of three interrelated systems [4,5]:

1. the combinatorial network which performs the logical function of the digital system,
2. the registers which store the data signals awaiting the synchronizing clock pulse, and
3. the clock distribution network which generates the synchronizing clock pulse and defines when data can flow from one register to the next.

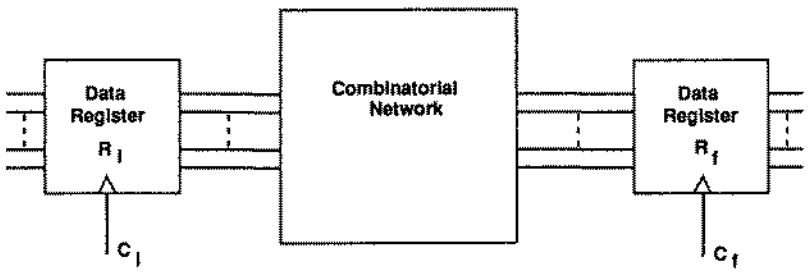

Figure1: Synchronous Data Path

The total delay of a data path is determined by the time required to leave the initial register once the clock signal C_i arrives, $T_{c\text{-}Q}$, the time necessary to propagate through the logic and interconnect, $T_{logic} + T_{int}$, and the time required to successfully propagate to and latch within the final register of the data path, $T_{set\text{-}up}$ [6].

$$T_{PD} = T_{c\text{-}Q} + T_{logic} + T_{int} + T_{set\text{-}up} \quad . \tag{1}$$

In (1), the individual delay components making up the total delay T_{PD} of a data path are summed. Those data paths whose total delay plus any clock skew are greatest represent the critical worst case timing requirements of a digital system, and the delay and clock skew of these paths must be minimized in order to maximize the performance of the entire digital system. Thus, in a high performance synchronous digital system, the critical paths constrain and define the maximum performance of the entire system. Therefore, the goal in designing a high performance system is to minimize each delay component in (1) as well as to utilize any possible advantages (and minimize any possible disadvantages) of the clock distribution circuitry which will increase the speed of operation of the critical data paths.

A general form of a data path is shown in Figure 2, where an initial register R_i begins the data path and is followed by N stages of logic and N+1 sections of interconnect, ending in a final register R_f. Each interconnect section is represented as a single pole time constant and designated as T_i, where i represents each logic and interconnect stage and N is the total number of logic stages. Thus, in a data path composed of N logic stages, there are N+1 interconnect sections.

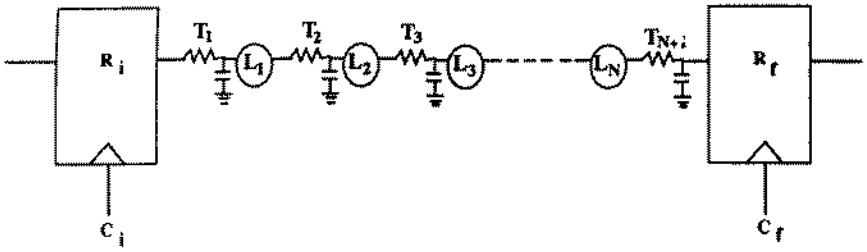

Figure 2: Synchronous Data Path with N Stages of Logic

Since logic paths are composed of only logic stages and interconnect sections, the total delay through a logic path can be modeled as the sum of the delay through the individual logic stages and interconnect sections. For convenience in representing the delay through the system, it is desirable to combine each individual RC interconnect section and logic stage as a single delay component of the logic path. This permits one to define the time required for the data signal to propagate through the i^{th} distributed RC interconnect section T_i and logic stage L_i as T_{fi}.

The minimum possible latency L_{min} of a data path, for a given function, occurs when it is completely unpipelined. For N logic stages traversed between the input and output of the system, L_{min} can be expressed as

$$L_{min} = \sum_{i=1}^{N} T_{fi} \quad . \tag{2}$$

When registers are inserted into the data path from system input to output (defined as the global data path), the minimum clock period can be decreased (providing a higher maximum clock frequency), albeit with an increase in latency. Each global data path is composed of individual cascaded register-to-register data paths and these are defined as local data paths. Each local data path is composed of an initial and final register and typically, *n* logic stages between them. Note that each register within a local data path performs double duty, serving as the initial (final) and final (initial) register of the current and previous (next) local data path, respectively.

In order to improve computational concurrency, the process of pipelining and retiming is applied to these systems. In the pipelining process, registers are inserted into the data path in order to provide temporal concurency of the data flow and increased system clock frequency. Retiming then follows the initial pipelining process by shifting the location of the pipeline registers in order to minimize the maximum clock period.

3. CLOCK DISRIBUTION NETWORKS

In a synchronous digital system, the global clock signal is used to define a relative time reference for all movement of data within that system. Because this function is vital to the operation of a synchronous system, much attention has been given to the characteristics of these clock signals and the networks used in their distribution [4,5,7-17]. Most synchronous digital systems consist of cascaded banks of sequential registers with combinatorial logic between each set of registers. The functional requirements of the digital system are satisfied by the logic stages, while the global performance and local timing requirements are satisfied by the careful insertion of pipeline registers into equally spaced time windows to satisfy critical worst case timing constraints [4,5,7-10,13] and the proper design of the clock distribution network to satisfy critical timing requirements as well as to ensure that no race conditions exist [7,9,10].

Each data signal typically is stored in a latched state within a bistable register awaiting the incoming clock signal to define when the data should leave the register. Once the enabling clock signal reaches the register, the data signal leaves the bistable register and propagates through the combinatorial network, and for a properly working system, enters the next register and is fully latched into that register before the next clock signal appears [6,8]. C_i and C_f represent the clock signals driving the initial register and the final register, respectively, and both originate from the same clock signal source. The clock delay of the initial clock signal T_{Ci} and the final clock signal T_{Cf} define the time reference when the data signals begin to leave their respective registers. These clock signals originate from a clock distribution network which is designed to generate a specific clock signal waveform which synchronizes each register [4-10]. The difference in delay between two sequentially adjacent clock paths, as shown in (3), is the clock skew T_{SKEW}. If the clock signals C_i and C_f are in complete synchronism (i.e., the clock signals arrive at their respective registers at exactly the same time), the clock skew is zero.

$$T_{SKEW} = T_{Ci} - T_{Cf} \quad . \tag{3}$$

The minimum allowable clock period T_{cpmin} between two registers in a sequentially adjacent data path is given by [4,5]

$$\frac{1}{f_{clkMAX}} = T_{cpmin} \geq T_{PD} + T_{SKEW} \quad , \tag{4}$$

where T_{PD} is defined in (1) and T_{SKEW} can be positive or negative depending on whether C_f leads or lags C_i, respectively. A timing diagram depicting each delay component in (1) in terms of the clock period is shown in Figure 3. These waveforms show the timing requirement of (4) being barely satisfied.

3.1 Maximum Data Path/Clock Skew Constraint Relationship

For a design to meet its specified timing requirements, the greatest collective propagation delay of any data path between a pair of data registers, R_i and R_f, being synchronized by a clock distribution network must be less than the inverse of the maximum clock frequency of the circuit as shown in (4). If the time of arrival of the clock signal at the final register of a data path T_{Cf} leads that of the time of arrival of the clock signal at the initial register of the same sequential data path T_{Ci} (see Figure 4A), the clock skew is referred to as positive clock skew and, under this condition, the maximum attainable operating frequency is decreased [4,5,7-10]. Positive clock skew is the

additional amount of time which must be added to the minimum clock period to reliably apply a new clock signal at the final register, where reliable operation implies that the system will function correctly at low as well as at high frequencies.

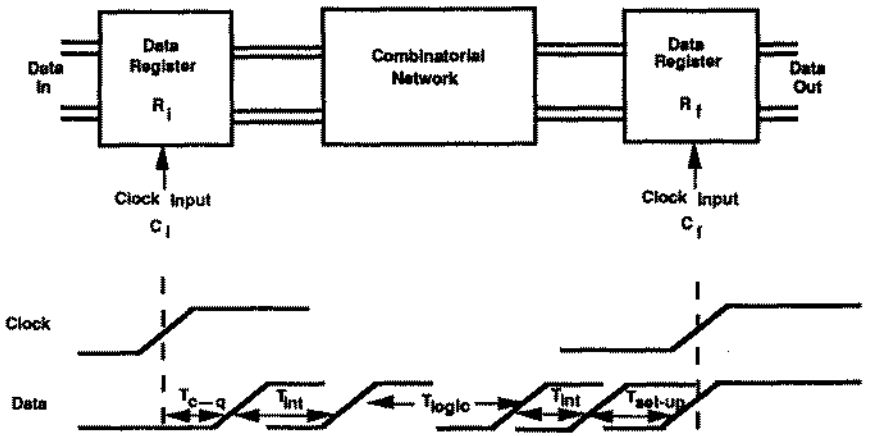

Figure 3: Timing Diagram of Clocked Data Path

In the positive clock skew case, the clock signal arrives at R_f before it reaches R_i. From (1) and (4), the maximum permissible positive clock skew can be expressed as

$$T_{SKEW} \leq T_{cp} - (T_{c\text{-}Q} + T_{logic} + T_{int} + T_{set\text{-}up}) \quad for\ T_{Ci} > T_{Cf} \quad . \qquad (5)$$

This situation is the typical critical path timing analysis requirement commonly seen in most high performance synchronous digital systems. In circuits where positive clock skew is significant and (5) is not satisfied, the clock and data signals should be run in the same direction, thereby forcing C_f to lag C_i and making the clock skew negative.

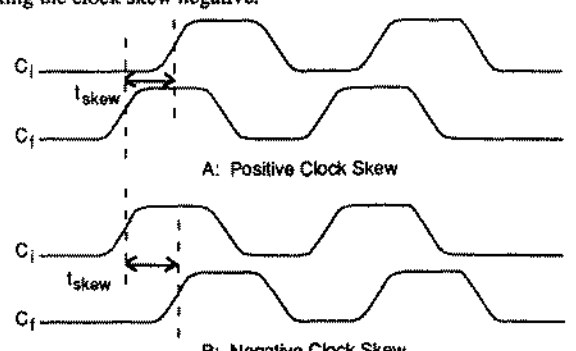

Figure 4: Clock Timing Diagrams

3.2 Minimum Data Path/Clock Skew Constraint Relationship

If the clock signal arrives at R_i before it reaches R_f (see Figure 4B), the clock skew is defined as being negative. Negative clock skew can be used to improve the maximum performance of a synchronous system by decreasing the delay of a critical path; however, a potential minimum constraint can occur [4,5,7,9,10], creating a race condition. In this case, the clock skew, when C_f lags C_i, must be less than the time required for the data to leave the initial register, propagate through the interconnect and combinatorial logic, and set-up in the final register (see Figure 3). If this condition is not met before the data stored in register R_f can be shifted out of R_f, it is overwritten by the data that had been stored in register R_i and had propagated through the combinatorial logic. Correct operation requires that R_f latches data which corresponds to the data R_i latched during the previous clock period. This constraint on clock skew is

$$|T_{SKEW}| \leq T_{PD} = T_{c\text{-}Q} + T_{logic} + T_{int} + T_{set\text{-}up} \quad for\ T_{Cf} > T_{Ci} \quad . \qquad (6)$$

An important example in which this minimum constraint can easily occur is in those designs which use cascaded registers, such as a serial shift register or a k-bit counter. As shown in Figure 5, T_{logic} is equal to zero and T_{int} approaches zero (since cascaded registers are typically designed, at the geometric level, to abut). If $T_{Cf} > T_{Ci}$ (i.e., negative clock skew), then the minimum constraint becomes

$$|T_{SKEW}| \leq T_{c\text{-}Q} + T_{set\text{-}up} \quad for\ T_{Cf} > T_{Ci} \quad . \qquad (7)$$

and all that is necessary for the system to malfunction is a poor relative placement of the flip flops or a highly resistive connection between C_i and C_f. In a circuit configuration such as a shift register or counter, where negative clock skew is a more serious problem than positive clock skew, provisions should be made to force C_f to lead C_i, as in the example circuit of Figure 5.

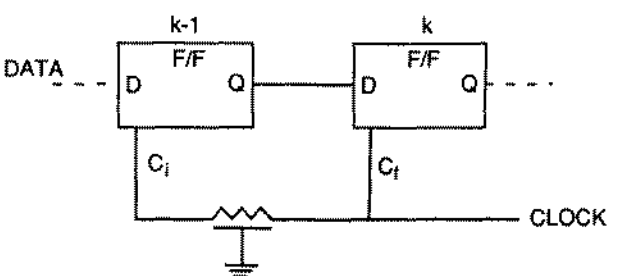

Figure 5: k-Bit Shift Register with Positive Clock Skew

As higher levels of integration are achieved, on-chip testability becomes necessary. Data registers, configured in the form of serial set/scan chains when operating in the test mode, are a common example of a built-in test design technique [18]. The placement of these circuits is typically

optimized around the functional flow of the data. When the system is reconfigured to use the registers in the role of the set/scan function, different path delays are possible. In particular, the clock skew of the local data path can be negative and greater in magnitude than the local register delays. Therefore, with increased negative clock skew, (7) may not be satisfied and the incorrect data will latch into the final register of the reconfigured local data path.

Finally, in ideal scaling of MOS devices, all linear dimensions and voltages are multiplied by the factor 1/S, where S > 1. Device dependent delays, such as $T_{c\text{-}Q}$, $T_{set\text{-}up}$, and T_{logic}, scale as 1/S while interconnect dominated delays such as T_{SKEW} remain constant to first order, and if fringing capacitance is considered, actually increase with decreasing dimensions [16,19]. Therefore, when examining dimensional scaling, (6) and (7) should be considered carefully.

3.3 Enhancing Synchronous Performance by Applying Negative Clock Skew

Negative clock skew, as previously mentioned and as shown in (4), can be used to improve synchronous performance by minimizing the delay of the critical worst case data paths [4,5,10,20]. By applying negative clock skew to the critical paths, excess time is shifted from the neighboring less critical local data paths to the critical local data paths by forcing C_i to lead C_f at each critical local data path. This negative clock skew represents the additional amount of time that the data signal at R_i has to propagate through the n stages of logic and n+1 sections of interconnect and into the final register. Negative clock skew subtracts from the logic path delay, thereby decreasing the minimum clock period. This, in effect, increases the total time that a given critical data path has to accomplish its functional requirements by giving the data signal released from R_i more time to propagate through the logic and interconnect stages and latch into R_f. Thus, the differences in delay between each local data path is minimized, thereby compensating for any inefficient partitioning of the global data path into local data paths common in most practical systems.

The maximum permissible negative clock skew of any data path, however, is dependent upon the clock period itself as well as the time delay of the previous data paths. This results from the structure of the serially cascaded local data paths making up the global data path. Since a particular clock signal synchronizes a register which functions in a dual role, as the initial register of the next local data path and as the final register of the previous data path, the earlier C_i is for a given data path, the earlier that same clock signal, now C_f, is for the previous data path. Thus, the use of negative clock skew in the i^{th} path results in a positive clock skew for the preceding path, which may then establish the new upper limit for the system clock frequency. It should be emphasized that in [7,9,10], the authors describe many of these characteristics of clock skew and its effects on the maximum clock frequency and, in [7,9], designate the lead/lag clock skew polarity (positive/negative clock skew) notation as the opposite of that used here.

4. PIPELINING AND RETIMING OF SYNCHRONOUS SEQUENTIAL SYSTEMS

In digital systems, the minimum latency, as described in (2), occurs when the data path consists entirely of logic stages; it is the time required for propagation of a data signal through these logic stages. The clock period for this system, which is also the latency, is equal to the time required to process one data sample. If new data appear at the input of a system at time intervals smaller than the latency for this simple configuration, registers can be inserted into the data path to increase the frequency at which new data signals are processed through the system and appear at the system output. This degrades the latency, however.

The total delay from the output of the initial register R_i to the output of the n^{th} logic stage is the sum of the individual T_{fi} terms along that local data path as shown below:

$$T_{logic} + T_{int} = \sum_{i=1}^{n} T_{fi} \quad . \tag{8}$$

Thus, for a local data path consisting of n logic stages, the time delay through the path T_{PD} can be expressed as

$$T_{PD} = T_{c\text{-}Q} + \sum_{i=1}^{n} T_{fi} + T_{set\text{-}up} \quad . \tag{9}$$

Equation (9) is composed of the delay required to get out of and into the initial and final registers, respectively, and the time required to propagate through n stages of logic and n + 1 sections of interconnect. If T_{REG} represents the total register related delay of both R_i and R_f, (9) can be written as

$$T_{PD} = T_{REG} + \sum_{i=1}^{n} T_{fi} \quad . \tag{10}$$

Thus, the total time to move a data signal through a data path is composed of the overhead requirements to get in and out of the final and initial registers as well as the time required to perform the logical operations.

The latency L is defined as the time required to move a data signal from the input of the system to its output. For the special case of no pipelining, the latency equals the minimum clock period. Each added register increases the latency by a register delay. Thus, the latency of a pipelined data path is the summation of the total delay through the global data path.

$$L = \sum_{i=1}^{N} T_{fi} + \sum_{k=1}^{M} T_{ek} \quad , \tag{11}$$

where N is the number of logic stages per global data path, M is the number of local data paths (and clock distribution networks) per global data path, and the total number of clock periods (and registers) required to move a particular data signal from the input of the system to its output is M+1. The maximum permissible negative clock skew T_{ek} in (11) is given by (12). In (12), T_{ek} is the aggregate delay of the k^{th} local data path due to the initial and final registers (T_{REGk}) and the

$$T_{ek} = T_{REGk} + T_{SKEWk} \quad , \tag{12}$$

clock distribution network (T_{SKEWk}). T_{ek} can be used to represent the acceptable tolerance of negative clock skew for the k^{th} local data path and can be described as the total effective delay of the

registers and clock distribution network per local data path. Note that T_{sk} is typically positive for most circuit configurations.

In a retimed system, the latency L is constant. The objective is to shift the temporal location of each of the pipeline registers so as to minimize the maximum clock period. Thus, for a given system,

$$\Delta L = 0 \quad . \tag{13}$$

Since, from (13), M is held constant and, in (11), N and T_{fi} are assumed constant, the only two remaining opportunities to improve system wide performance are to temporally shift the pipeline registers keeping M constant (i. e., classical retiming), or to vary T_{sk}. In (12), T_{REG_k} is assumed constant for a specific circuit implementation, thereby leaving T_{SKEW_k} as the sole parameter to vary in (4) and (11) in order to improve system performance.

The clock period T_{cp} can be expressed as

$$T_{cp} \geq Max \left\{ T_{REG_k} + \sum_{i=1}^{n} T_{fi} + T_{SKEW_k} \right\} \tag{14}$$

$$T_{cp} = \sum_{i=1}^{N} T_{fi} \qquad \text{for M = 0} \tag{15}$$

$$T_{cp} = T_{sk} + \sum_{i=1}^{n} T_{fi} \qquad \text{for M} \geq 1 \tag{16}$$

where Max is used to designate the critical worst case path. For a pipelined global data path with registers placed at both its input and output, the total latency L can also be described by the relation

$$L = \frac{M+1}{f_{clk}} \quad . \tag{17}$$

The primary objective of applying localized clock distribution design to retimed circuits is to partition a global data path into local data paths of approximately equal delay. By applying the concept of local negative and positive clock skew, one can, in effect, even out the delays of each pipelined data path, making each local data path of approximately equal delay.

Thus, this approach can be used to increase both the clock frequency and the temporal resolution of retimed circuits. Typically, a pipeline register is repositioned within a Boolean network in order to minimize overall clock period, thereby providing a temporal resolution defined by the minimum number of logic elements shifted within the Boolean network [1-3,21,22]. By using the approach of positive and negative clock skew, in addition to increasing the maximum system clock frequency, one can shift time between local data paths at a resolution much smaller than is possible by standard retiming algorithms [10]. Therefore, the maximum resolution of a retimed circuit is defined by the precision of the clock distribution network. An order of magnitude improvement in resolution is possible by applying the aforementioned local clock distribution design methods instead of relying solely on adjusting the relative temporal placement of the logic elements. For a 1 µm CMOS technology, precision of the clock distribution network on the order of 200 to 300 picoseconds [8] are possible instead of 2 to 5 nanoseconds typical of the logical transformations of many retiming algorithms [21,22].

5. AN EXAMPLE OF APPLYING LOCAL CLOCK SKEW TO RETIMED CIRCUIT

Consider the non-recursive retimed system shown in Figure 6 where the horizontal oval represents a logic delay and the vertical oval a clock delay. Since the local data path from R_2 to R_3 represents the worst case path (assuming the register delays are equal), by delaying C_3 with respect to C_2, negative clock skew is added to the R_2 - R_3 local data path. If C_1 is synchronized with C_3, then the R_1 - R_2 local data path receives some positive clock skew. Thus, if the register delays are 2 ns., C_2 would be designed to lead C_3 by 1.5 ns., forcing both paths to be the same total $T_{PD} + T_{SKEW} = 7.5$ ns. Thus, the critical path of the retimed circuit is further temporally refined to the precision of the clock distribution network and the entire system (for this simple example) could operate at a clock frequency of 133.3 MHz. instead of 111.1 MHz. Note that $|T_{SKEW}| < T_{PD}$ (|-1.5 ns.| < 9 ns.) for the R_2-R_3 local data path; therefore, the correct data signal is successfully latched into R_3 and no minimum data path/clock skew constraint relationship exists. This design technique of applying localized clock skew is particularly affective in sequentially adjacent temporally irregular local data paths; however, it is applicable to any type of synchronous sequential system and for certain architectures, significant improvement in performance and reliability is both possible and likely.

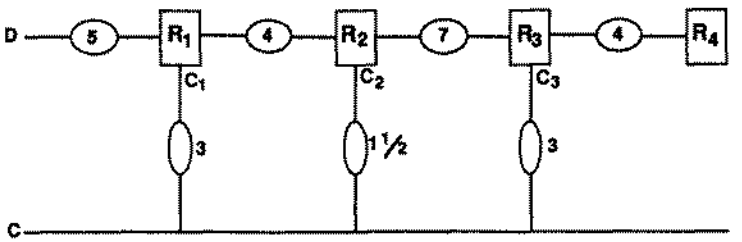

Figure 6: Example of Localized Clock Distribution Design Technique Applied to Retimed Circuit

6. CONCLUSIONS

Clock distribution networks have been described in terms of their data path timing requirements. Local clock skew can be used to improve performance by locally varying its magnitude and lead/lag relationship with its sequentially adjacent local data path. Data path/clock skew constraint relationships were developed for both the positive clock skew case and the negative clock skew case. From these specific constraint relationships, recommended design procedures were offered to eliminate the deleterious effects of clock skew on both the maximum performance and the reliable operation of a synchronous digital system. Techniques were described for improving synchronous performance by applying negative clock skew for equalizing path delays between local data

paths, thereby minimizing the delay of the critical worst case paths. Thus, local clock distribution networks can be used to tune the effective path delays of each local data path. This provides logically retimed circuits with retiming precision consistent with the resolution of the clock distribution network. Thus, retiming becomes a two-phase process: 1) logical retiming for gross temporal partitioning and 2) local clock distribution design for fine readjustment of the clock period. An order of magnitude improvement in resolution is attainable in retimed circuits by applying local clock distribution design methods instead of relying solely on moving logic elements across sequentially adjacent data paths.

7. REFERENCES

[1] Leiserson, C. E. and Saxe, J. B., "Optimizing Synchronous Systems," *Proceedings of 22nd Annual Symposium on Foundations of Computer Science*, October 1981, pp. 23-26.

[2] Leiserson, C. E. and Saxe, J. B., "Optimizing Synchronous Circuitry by Retiming," *Proceedings of the 3rd Caltech Conference on Very Large Scale Integration*, 1983, pp. 87-116.

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

[4] Friedman, E. G. and Mulligan, Jr., J. H., "Clock Frequency and Latency in Synchronous Digital Systems," *IEEE Transactions on Signal Processing*, Vol. SP-39, No. 4, April 1991, pp. 930-934.

[5] Friedman, E. G. and Mulligan, Jr., J. H., "Pipelining of High Performance Synchronous Digital Systems," *International Journal of Electronics*, Volume 70, Number 5, May 1991, pp. 917-935.

[6] Friedman, E. G., "The Limiting Performance of a CMOS Bistable Register based on Waveform Considerations," *International Journal of Electronics* (in press).

[7] Hatamian, M. and Cash, G. L., "Parallel Bit-Level Pipelined VLSI Designs for High Speed Signal Processing," *Proceedings of the IEEE*, Vol. 75, No. 9, September 1987, pp. 1192-1202.

[8] Friedman, E. G. and Powell, S., "Design and Analysis of a Hierarchical Clock Distribution System for Synchronous Standard Cell/Macrocell VLSI," *IEEE Journal of Solid-State Circuits*, Vol. SC-21, No. 2, April 1986, pp. 240-246.

[9] Hatamian, M., "Understanding Clock Skew in Synchronous Systems," Chapter 6 of *Concurrent Computations (Algorithms, Architecture, and Technology)*, S. K. Tewksbury, B. W. Dickinson, and S. C. Schwartz (Eds.), New York, New York: Plenum Publishing, 1988, ch. 6.

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

[11] Anceau, F., "A Synchronous Approach for Clocking VLSI Systems," *IEEE Journal of Solid-State Circuits*, Vol. SC-17, No. 1, February 1982, pp. 51-56.

[12] Wann, D. and Franklin, M., "Asynchronous and Clocked Control Structures for VLSI Based Interconnection Networks," *IEEE Transactions on Computers*, Vol. C-32, No. 3, March 1983, pp. 284-293.

[13] Alves Marques, J. and Cunha, A., "Clocking of VLSI Circuits," *VLSI Architecture*, Randell and Treleaven (Eds.), Englewood Cliffs, New Jersey:Prentice-Hall, 1983, pp. 165-178.

[14] Unger, S. H. and Tan, C-J., "Clocking Schemes for High-Speed Digital Systems," *IEEE Transactions on Computers*, Vol. C-35, No. 10, October 1986, pp. 880-895.

[15] Bakoglu, H. B., Walker, J. T., and Meindl, J. D., "A Symmetric Clock-Distribution Tree and Optimized High-Speed Interconnections for Reduced Clock Skew in ULSI and WSI Circuits," *Proceedings of IEEE International Conference on Computer Design*, October 1986, pp. 118-122.

[16] Gura, C. V., "Analysis of Clock Skew in Distributed Resistive-Capacitive Interconnects," University of Illinois, Urbana, Illinois, SRC Technical Report No. T87053, June 1987.

[17] Wagner, K. D., "Clock System Design," *IEEE Design & Test of Computers*, October 1987, pp. 9-27.

[18] Williams, T. and Parker, K., "Design for Testability-A Survey," *Proceedings of the IEEE*, Vol. 71, No. 1, January 1983, pp. 98-112.

[19] Bakoglu, H. B. and Meindl, J. D., "Optimal Interconnection Circuits for VLSI," *IEEE Transactions on Electron Devices*, Vol. ED-32, No. 5, May 1985, pp. 903-909.

[20] E. G. Friedman, *Performance Limitations in Synchronous Digital Systems*, Ph.D. Dissertation, University of California, Irvine, June 1989.

[21] De Micheli, G., "Synchronous Logic Synthesis: Algorithms for Cycle-Time Minimization," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-10, No. 1, January 1991, pp. 63-73.

[22] Malik, S., Sentovich, E. M., Brayton, R. K., and Sangiovanni-Vincentelli, A., "Retiming and Resynthesis: Optimizing Sequential Networks with Combinational Techniques," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-10, No. 1, January 1991, pp. 74-84.