Clock Frequency and Latency in Synchronous Digital Systems

Eby G. Friedman, *Senior Member, IEEE*, and J. H. Mulligan, Jr., *Fellow, IEEE*

Abstract—**This paper describes the tradeoff in the design of synchronous digital systems between clock frequency and latency in terms of the circuit characteristics of a pipelined data path. A design paradigm relating latency and clock frequency as a function of the level of pipelining is developed for studying the performance of a synchronous system. This perspective permits the development of design equations for constrained and unconstrained design problems which describe these performance parameters in terms of the delays of the logic, interconnect, and registers, clock skew, and the number of logic stages.**

These results provide a new approach to the design of those synchronous digital systems in which latency and clock frequency are of primary importance. From the behavioral specifications for the proposed system, the designer can use these results to select the best logic architecture and the best available device technology to determine if the performance specifications can be satisfied, and if so, what design options are available for optimization of other system attributes, such as area. Furthermore, the results provide a systematic procedure for the design of the synchronous system once the logic architecture and technology have been selected by the designer.

I. Introduction

In a synchronous digital system, the latency is defined as the total time required to process a signal by moving a particular data signal from the input of a system to its output. The minimum latency occurs when the data path is composed entirely of logic stages; it is the time required to propagate a data signal through these logic stages. The clock period for this system is equal to the time required to process one data sample; namely, the latency. If the system requirement for the time interval at which data is sampled at the input (i.e., the clock period) is less 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 and appear at the output of the system, thereby degrading the latency. This process is spoken of as pipelining.

Different applications of synchronous digital systems suggest different criteria for use in the optimization of their performance. For example, for a broad class of systems, optimization is done on the basis of a speed/area product. On the other hand, there are applications which are particularly sensitive to the latency of the system implementation. The results discussed in this paper are primarily intended for feedforward nonrecursive systems and describe a design approach for choosing the appropriate level of pipelining, thereby defining the system clock frequency and latency based on application specific performance requirements and architectural and technological limitations.

Systems can be designed which minimize the latency, maximize the clock frequency, or achieve tradeoffs between minimum latency and maximum clock frequency. In Section II of this paper, relations between latency and clock frequency are developed in terms of the delay components and circuit characteristics of a data path. In Section III, a graphical interpretation of the performance tradeoffs of a pipelined system is presented, illustrating the constraints and limitations of the design space.

Most synchronous digital systems are designed to satisfy specific performance requirements such as minimum clock frequency or maximum latency. Thus, in these systems the design problem becomes either one of maximizing the clock frequency while not exceeding a maximum latency or minimizing the latency while meeting a specified clock frequency. In certain systems, neither the latency nor the clock frequency ultimately constrains the design problem. In these unconstrained design problems, the level of pipelining can be chosen to tradeoff the latency with the clock frequency. These constrained and unconstrained systems are investigated in Section IV. Finally, some conclusions are presented in Section V.

II. Relation Between Latency and Clock Frequency

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 define the time required for the data signal to propagate through the ith distributed RC interconnect section T_i and logic stage L_i (see Fig. 1) as T_{fi} and the average delay of all the logic and interconnect stages per data path as T_{fN}. Thus, an unpipelined data path provides the minimum latency $L_{\min}$ of a data path and, for N logic stages traversed between the input and output of the system, $L_{\min}$ can be expressed as

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

Once registers are inserted into the data path representing the complete system, the minimum clock period is decreased (higher maximum clock frequency), albeit with an increase in latency. If the original system data path is defined to be the global data path, then each individual data path between inserted registers within a global data path can be described as a local data path. Each local data path is composed of an initial and final register and typically, n logic stages between them. Note that each register within each 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.

For each pipeline register, additional register related delay

Manuscript received July 28, 1989; revised April 4, 1990. This work was supported in part by a Hughes Aircraft Doctoral Fellowship.

E. G. Friedman was with the Department of Electrical and Computer Engineering, University of California, Irvine, CA 92717. He is now with the Hughes Aircraft Company, Carlsbad, California 92009.

J. H. Mulligan, Jr., is with the Department of Electrical and Computer Engineering, University of California, Irvine, CA 92717.

IEEE Log Number 9042254.

1053-587X/91/0400-0930$01.00 © 1991 IEEE

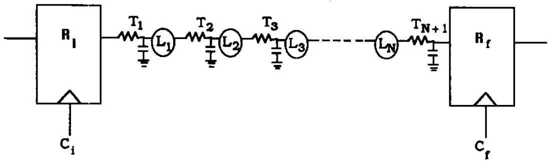


Fig. 1. Synchronous data path with N stages of logic.

components are added to the logic and interconnect delays. These register related delay components, as observed in Fig. 1, originate in R_i and R_f. T_{c-Q} is the time interval between the arrival of the clock signal at R_i and the appearance of the data signal at the register output. The time required for the signal at the output of the final logic stage to propagate through the $N + 1$st interconnect section and lâtch into the final register R_f is the set-up time $T_{\text{set-up}}$.

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

$$T_{\text{PD}} = T_{c-Q} + \sum_{i=1}^{n} T_{fi} + T_{\text{set-up}}. \tag{2}$$

If T_{REG} represents the total register related delay, then

$$T_{\text{REG}} = T_{c-Q} + T_{\text{set-up}} \tag{3}$$

and (2) can be written as

$$T_{\text{PD}} = T_{\text{REG}} + \sum_{i=1}^{n} T_{fi}. \tag{4}$$

As shown in Fig. 1, the times of arrival of the initial clock signal C_i and the final clock signal C_f 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 typically designed to generate a specific clock signal waveform synchronizing each register [1]–[3]. The difference in delay between two sequentially adjacent clock paths is described as 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. If the time of arrival of the clock signal at the final register of a data path (C_f) leads that of the clock signal at the initial register of the same sequential data path (C_i), then the clock skew is defined as positive; this condition degrades the maximum attainable operating frequency. This positive clock skew represents the additional amount of time which must be added to the minumum clock period to reliably apply a new clock signal at the final register. If C_f lags C_i, the clock skew is defined to be negative; this can be used to improve the maximum performance of a synchronous system. This negative clock skew represents additional amount of time for the data signal at R_i to propagate through the n stages of logic and $n + 1$ sections of interconnect into the final register. This clock skew is subtracted from the logic path delay, thereby decreasing the minimum clock period. 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 occurs because the use of negative clock skew in the ith path results in a positive clock skew for the preceding path, which may establish the upper limit for the system clock frequency, as discussed below. It should be noted that in [1], [3], Hatamian and Cash describe these characteristics of clock skew and its effects on the maximum clock frequency and designate the lead/lag clock skew polarity (positive/negative clock skew) opposite to that described herein.

The maximum clock frequency at which a synchronous digital system can move data is defined in (5) below:

$$f_{\text{clk}} = \frac{1}{T_{\text{cp}}} \leq \frac{1}{T_{\text{PD}} + T_{\text{SKEW}}} \tag{5}$$

where T_{cp} is the clock period, T_{PD} is defined in (4), and the local data path with the greatest $T_{\text{PD}} + T_{\text{SKEW}}$ represents the critical path of the system, i.e., establishes the maximum clock frequency. Note that for positive clock skew, the maximum clock frequency decreases while for negative clock skew, the maximum clock frequency increases.

If a single register is inserted into the data path, registers external to the system would be required to synchronize the external data flow. Two registers, one at the input and the other at the output of the global data path, represents a self-contained synchronous system (as shown in Fig. 1). With each additional register, the latency increases. For a pipelined data path, the latency is the summation of the total delay through the global data path as shown below in (6) and (7):

$$L = \sum_{i=1}^{N} T_{fi} + \sum_{k=1}^{M} T_{ek} \tag{6}$$

$$L = NT_{fN} + MT_{eM} \tag{7}$$

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 $M + 1$ is the number of clock periods (and registers) required to move a particular data signal from the input of the system to its output. T_{ek} can be represented by (8) where T_{ek} is the aggregate delay of the kth local data path due to the initial and final registers (T_{REG}) and the clock distribution network (T_{SKEW}):

$$T_{ek} = T_{\text{REG}} + T_{\text{SKEW}}. \tag{8}$$

The average of the individual T_{ek} values over all the M serially connected cascaded data paths is defined as T_{eM}. Since T_{fN} is the average delay of all of the logic and interconnect stages (between the input and the output of the system), for convenience and improved interpretation, (7) represents the latency in terms of average delays instead of individual summations.

The average number of logic stages per local data path n is given by (9) below:

$$n = \frac{N}{M}. \tag{9}$$

The clock period T_{cp} can be expressed as

$$T_{cp} \geq T_{\text{REG}} + nT_{fN} + T_{\text{SKEW}} \tag{10}$$

$$T_{cp} = \begin{cases} NT_{fN} & \text{for } M = 0 \quad (11) \\ T_{eM} + \dfrac{NT_{fN}}{M} & \text{for } M \geq 1. \quad (12) \end{cases}$$

III. Design Paradigm for Pipelined Synchronous Systems

Registers are inserted into global data paths in order to increase the clock frequency of a digital system with, albeit, an increase in the latency. This tradeoff between clock frequency and latency is graphically described in Fig. 2. In this figure both the latency and the clock period are shown as a function of the number of pipeline registers M inserted into a global data path.

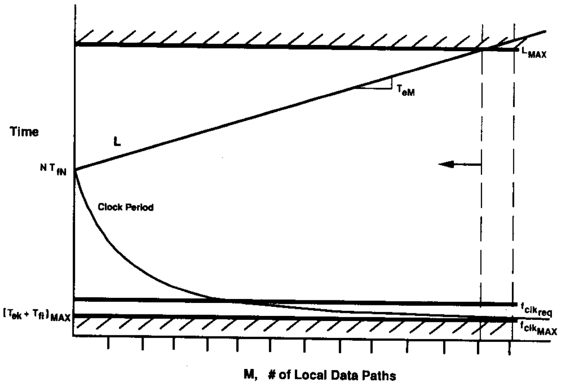


Fig. 2. Design paradigm for pipelined synchronous systems.

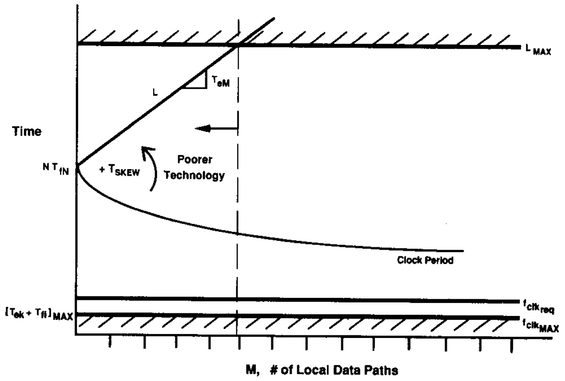


Fig. 3. Effect of positive clock skew and technology on design paradigm.

If no registers are inserted into the data path, the minimum latency $L_{\min}$ is the summation of the individual logic delays, NT_{fN}, as shown by (1) or by (7) with $M = 0$. As each register is inserted into the global data path, L increases by T_{eM}. Thus L increases linearly with M; this is depicted in Fig. 2.

As seen from (12), the expression for the clock period contains a term which varies inversely with M; this behavior is shown in Fig. 2. From (9) and (12), it is seen that the minimum clock period occurs when n equals one. The local data path having the largest value of $T_{ek} + T_{fi}$, defined as the critical data path, establishes the maximum clock frequency f_{clkMAX} for the system. This assumes that logical operations are being performed (i.e., the function is not a simple shift register). The MAX subscript in Fig. 2 is used to emphasize that the critical local data path constrains the minimum clock period (maximum clock frequency) of the total global data path.

Most design requirements must satisfy some specified maximum time for latency while satisfying or surpassing a required clock frequency. The design constraints due to $L_{\max}$ and f_{clkMAX} are shown in Fig. 2 by the vertical dashed lines. Thus, for a given $L_{\max}$, the recommended maximum clock frequency and level of pipelining is defined by the intersection of the L curve and the $L_{\max}$ line. If $L_{\max}$ is not specified and the desire is to make the clock frequency as high as possible, then the recommended f_{clk} is defined by the intersection of the clock period curve and the f_{clkMAX} line. Thus, for a specified L and f_{clk}, the extent of the possible design space is indicated by the horizontal arrow. If L and f_{clk} are both of importance and no $L_{\max}$ or f_{clkMAX} is specified, then some optimal level of pipelining is required to provide a "reasonably high" frequency while maintaining a "reasonable" latency. This design choice is represented by a particular value of M, defining an application specific f_{clk} and L.

The effects of clock skew, technology, and logic architecture on latency and clock period are graphically demonstrated in Figs. 3 and 4. If the clock skew is positive or if a poorer technology (i.e., slower) is used, as shown in Fig. 3, then T_{eM} increases and L reaches $L_{\max}$ at a smaller value of M than previously. In addition, the minimum clock period increases (decreasing the maximum clock frequency) which, for large positive clock skew or a very poor technology, eliminates any possibility of satisfying a specified clock frequency f_{clkreq} and decreases the entire design space as defined by the intersection of L and $L_{\max}$. Also, for a poorer technology or logic architecture, the intersection between either the latency or the clock period curve and the ordinate shifts upwards since T_{fN} increases due to the slower technology and N increases for the less optimal architecture.

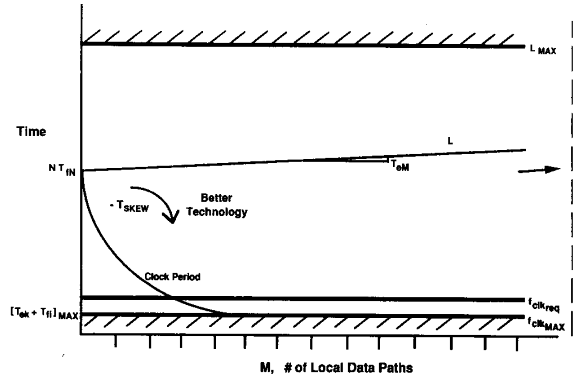


Fig. 4. Effect of negative clock skew and technology on design paradigm.

If the clock skew is negative or a better technology (i.e., faster) is used, as shown in Fig. 4, T_{eM} decreases and the latency is less dependent on M. In addition, the minimum clock period decreases, satisfying f_{clkreq} and f_{clkMAX} with less pipelining. The optional design space, represented by the intersection of L and $L_{\max}$, is much larger, permitting higher levels of pipelining if very high clock rates are desired. Also, for a better technology or logic architecture, the intersection between either the latency or the clock period curve and the ordinate shifts downward since T_{fN} decreases due to the faster technology and N decreases for a more optimal architecture. Thus, Figs. 3 and 4 graphically describe how clock skew, technology, and logic architecture affect both the latency and the maximum clock frequency of a pipelined synchronous digital system.

IV. Formulation of Design Equations

A. Design Objectives

Three types of design problems are considered using this approach: 1) the maximum latency constrains the design problem, 2) the required clock frequency constrains the design problem, or 3) the problem is unconstrained and a tradeoff between L and f_{clk} must be made.

1) Maximum Latency: In applications where the maximum latency of a system is specified and $L_{\max}$ constrains the design space, the degree of pipelining can be determined from (13), where T_{eM} is taken as the estimate of an average $T_{\text{REG}} + T_{\text{SKEW}}$:

$$M \leq \frac{L_{\max} - NT_{fN}}{T_{eM}}. \tag{13}$$

A range of possible values of clock frequency is defined in (14), assuming a value of M from (13), where the lower bound on clock frequency is due to the constraint on maximum latency and the upper bound ensures correct system operation:

$$\frac{M}{L_{\max}} \leq f_{\text{clk}} \leq \frac{M}{MT_{eM} + NT_{fN}}. \tag{14}$$

Thus, as shown in Fig. 2, for a given maximum latency and knowledge of the average logic, register, and clock delay characteristics of a global data path, the degree of pipelining and range of clock frequency can be directly determined.

2) Required Clock Frequency: In applications where the maximum clock frequency is specified and f_{clkMAX} constrains the design space, the number of registers and the latency can be determined from (15) and (7), respectively,

$$M = \frac{NT_{fN}}{T_{\text{CP}} - T_{eM}}. \tag{15}$$

Thus, as shown in Fig. 2, for a given maximum or required clock frequency and knowledge of the average logic, register, and clock delay characteristics of global data path, the minimum latency and the required level of pipelining can be directly determined.

3) Unconstrained Design Requirement: Each additional register increases L by T_{eM} and decreases the maximum clock period by the decreased logic delay [4]-[11]. There exists a level of pipelining where the increase in latency costs the system more than the increase in clock frequency benefits the system. In order to quantify this, an arbitrary performance criterion (the pipelining efficiency, P_e) is defined to describe the performance cost of latency. P_e is a measure of the relative performance penalty incurred by the insertion of a single additional pipeline register to an existing global data path. It is a normalized function which is the ratio of the total local logic delay to the total local data path delay, after the register has been inserted. It defines what percentage of the local data path delay is logic related and what percentage is register related. As n increases, the ratio of the total local logic delay to the total local data path delay increases toward unity, reaching it when n is infinite (or practically, when the total local logic delay is much greater than the register delay).

The benefit of inserting a register into a data path is decreased clock period as described by (12). A measure of the cost/benefit of inserting registers into an N stage global data path is the function $P_e f_{\text{clk}}$, where P_e increases for increasing n and f_{clk} decreases for increasing n. A different function could be applied if the effects of increased area, for example, were also of significant importance [7]-[9], [12]. Cappello *et al.* [8], [9] describe an AP product, where A is the chip area and P is the clock period, and use this figure of merit to optimize the speed/area performance of a pipeline system. However, the results described in this paper emphasize optimal latency and clock frequency over area/speed optimization. These results provide an approach to the design of those systems in which both latency and clock frequency are of primary importance.

If one assumes that P_e and f_{clk} are assigned equal importance and there are no constraints placed on $L_{\max}$ or f_{clkMAX} in achieving the design of the system, then an optimal value of the number of logic stages between pipeline registers N_{opt} can be obtained by determining where the product $P_e f_{\text{clk}}$ is maximized or where

$$\frac{d(P_e f_{\text{clk}})}{dn} = 0. \tag{16}$$

By the use of (16), N_{opt}, the optimal number of logic stages per local data path, is obtained as

$$N_{\text{opt}} = \frac{1}{T_{fN}} \sqrt{T_{\text{REG}}(T_{\text{REG}} + T_{\text{SKEW}})}. \tag{17}$$

Under the condition of an ideal clock distribution network with zero clock skew, (17) simplifies to (18):

$$N_{\text{opt}} = \frac{T_{\text{REG}}}{T_{fN}}. \tag{18}$$

N_{opt}, in (18), is the ratio of the register delay overhead to the average stage delay of the data path. If $T_{\text{REG}} \ll T_{fN}$, which occurs when the stage performs a large high level function, then the cost of inserting registers is small and N_{opt} should be as small as feasible (since N_{opt} must be an integer, its smallest realizable value is one) or one should pipeline as often as the system permits. If $T_{\text{REG}} \gg T_{fN}$, which often exists when operating at the level of individual logic stages, then the cost of inserting registers is high and N_{opt} is some large number specified by (18). Another interpretation of (18) is that the optimal number of logic stages between registers occurs when the total logic path delay NT_{fn} equals the total register delay T_{REG}, thereby maximizing $P_e f_{\text{clk}}$.

T_{SKEW} in (17) can be zero, negative, or positive with the constraint that if T_{SKEW} is negative, its magnitude must be less than T_{REG}. It is interesting to note that the effect of clock skew on N_{opt} is relative to T_{REG} and T_{fN}. Thus, if T_{REG} is large with respect to T_{SKEW}, the relation essentially reduces to (18). Also, positive clock skew adds directly to T_{REG} and increases the cost of pipelining, thereby increasing the recommended number of logic stages between registers and quantifying how the clock distribution network affects the optimal design of a high speed data path.

V. Conclusions

Latency and clock frequency are convenient parameters on which to base the design of high speed synchronous digital systems. The results of this paper deal directly with the systematic design of those systems based upon these two performance attributes.

In system design, global data paths are often partitioned into local pipelined data paths, thereby decreasing the delay of the critical paths and increasing the clock frequency, albeit with an increase in latency. The results presented deal specifically with three types of design options; namely, that in which 1) $L_{\max}$ constrains the design space, 2) f_{clkMAX} constrains the design space, and 3) the design space is unconstrained and a tradeoff must be made between L and f_{clk}. The solution suggested for the unconstrained design problem is the use of an algorithm which considers the effects of increased latency and increased clock frequency for increasing levels of pipelining.

There is an important class of practical systems which require both high clock frequency and minimal latency (e.g., radar, sonar, high speed computing) for which there is a need for developing a design approach which satisfies their application specific performance objectives. The results presented in this paper permit the implementation of a systematic strategy for designing high performance synchronous digital systems in which both clock frequency and latency are of primary interest.

Acknowledgment

The authors thank the reviewers for comments which were helpful in the revision of the manuscript.

References

[1] M. Hatamian and G. L. Cash, "Parallel bit-level pipelined VLSI designs for high speed signal processing," *Proc. IEEE*, vol. 75, no. 9, pp. 1192-1202, Sept. 1987.

[2] E. G. Friedman and S. Powell, "Design and analysis of a hierarchical clock distribution system for synchronous standard cell/macrocell VLSI," *IEEE J. Solid-State Circuits*, vol. SC-21, no. 2, pp. 240-246, Apr. 1986.

[3] M. Hatamian, "Understanding clock skew in synchronous systems," in *Concurrent Computations (Algorithms, Architecture, and Technology)*, S. K. Tewksbury, B. W. Dickinson, and S. C. Schwartz, Eds. New York: Plenum, 1988, ch. 6.

[4] L. W. Cotton, "Circuit implementation of high-speed pipeline systems," in *Proc. Fall Joint Comput. Conf.*, 1965, pp. 489-504.

[5] P. R. Cappello and K. Steiglitz, "Bit-level fixed-flow architectures for signal processing," in *Proc. IEEE Int. Conf. Circuits Comput.*, Sept. 1982, pp. 570-573.

[6] P. R. Cappello and K. Steiglitz, "Completely-pipelined architectures for digital signal processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, no. 4, pp. 1016-1023, Aug. 1983.

[7] C. E. Leiserson and J. B. Saxe, "Optimizing synchronous systems," in *Proc. 22nd Annu. Symp. Foundations Comput. Sci.*, Oct. 1981, pp. 23-26.

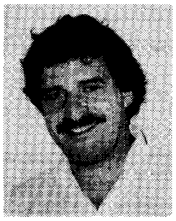
[8] P. R. Cappello, A. LePaugh, and K. Steiglitz, "Optimal choice of intermediate latching to maximize throughput in VLSI circuits," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1983, pp. 935-938.

[9] P. R. Cappello A. LaPaugh, and K. Steiglitz, "Optimal choice of intermediate latching to maximize throughput in VLSI circuits," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 1, pp. 28-33, Feb. 1984.

[10] J. R. Jump and S. R. Ahuja, "Effective pipelining of digital systems," *IEEE Trans. Comput.*, vol. C-27, no. 9, pp. 855-865, Sept. 1978.

[11] M. Hatamian, L. A. Hornak, T. E. Little, S. K. Tewksbury, and P. Franzon, "Fundamental interconnection issues," *AT&T Tech. J.*, vol. 66, no. 4, pp. 13-30, July/Aug. 1987.

[12] K. O. Siomalas and B. A. Bowen, "Synthesis of efficient pipelined architectures for implementing DSP operations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 6, pp. 1499-1508, Dec. 1985.



Eby G. Friedman (S'78-M'79-SM'90) was born in Jersey City, NJ, in 1957. He received the B.S. degree in electrical engineering from Lafayette College, Easton, PA, in 1979 and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Irvine, in 1981 and 1989, respectively.

He was previously employed by Philips Gloeilampen Fabrieken, Eindhoven, The Netherlands, in 1978 where he worked on the design of bipolar differential amplifiers. From 1979 to 1983, he was employed by Hughes Aircraft Company, Newport Beach, CA, working in the areas of custom IC design, software compatible gate array design, one- and two-dimensional device modeling, circuit modeling, and double-level metal process development. He is currently manager of the Signal Processing Design and Test Department, Technology Center of Hughes Aircraft Company, Carlsbad, CA, responsible for the design and test of high performance CMOS and BIMOS analog and digital IC's, the development of complementary design and test methodologies and CAD tools, functional and parametric test, and the development of high performance and high resolution DSP and oversampled systems. He is the author of several papers and presentations in the fields of VLSI design, CMOS design techniques and CAD tools, silicon compilation, and the theory and application of synchronous clock distribution networks to high performance digital systems.

J. H. Mulligan, Jr., (S'41-M'44-SM'54-F'59-LF'86) received the B.E.E. and E.E. degrees from the Cooper Union School of Engineering in 1943 and 1947, the M.S. degree from Stevens Institute of Technology in 1945, and the Ph.D. degree from Columbia University in 1948, both in electrical engineering.

His career includes engineering responsibilities in industrial, government, and academic organizations. He has been employed by Bell Laboratories, the Naval Research Laboratory, and the Allen B. DuMont Laboratories. From 1949 to 1968 he was a member of the faculty of the Department of Electrical Engineering at New York University, serving as Chairman of the Department from 1952 to 1968. From 1968 to 1974 he was Secretary and Executive Officer of the National Academy of Engineering. He served as Dean of the School of Engineering at the University of California, Irvine, from 1974 to 1977, following which he returned to full-time teaching and research as Professor of Electrical Engineering at that institution.