Variable Input Delay CMOS Logic for Low Power Design

Tezaswi Raja, *Member, IEEE*, Vishwani D. Agrawal, *Fellow, IEEE*, and Michael L. Bushnell, *Fellow, IEEE*

***Abstract*—We propose a new complementary metal-oxide semiconductor (CMOS) gate design that has different delays along various input to output paths within the gate. The delays are accomplished by inserting selectively sized "permanently on" series transistors at the inputs of a logic gate. We demonstrate the use of the variable input delay CMOS gates for a totally glitch-free minimum dynamic power implementations of digital circuits. Applying a linear programming method to the c7552 benchmark circuit and using the gates described in this paper, we obtained a power saving of 58% over an unoptimized design. This power consumption was 18% lower than that for an alternative low power design using conventional CMOS gates. The optimized circuits had the same critical path delays as their original unoptimized versions. Since the overall delay was not allowed to increase, the glitch elimination with conventional gates required insertion of delay buffers on noncritical paths. The use of the variable input delay gates drastically reduced the required number of delay buffers.**

***Index Terms*—CMOS delay devices, CMOS logic gate design, design automation, digital integrated circuits, dynamic power, linear programming, low power design.**

I. Introduction

MODERN digital circuits consist of logic gates implemented in the complementary metal oxide semiconductor (CMOS) technology. The power consumption of these circuits has two components [14]. The *dynamic power* is consumed only when the circuit performs a function and signals change. *Leakage or static power* is consumed all the time, i.e., even when the circuit is idle. It is unnecessary and one would like to eliminate it. But there are practical difficulties. Because the advanced CMOS technologies have higher leakage, this component has received much attention. The ways to reduce leakage work at the transistor design and manufacturing process levels [10]. On the other hand, we realize that the dynamic power cannot be eliminated completely because it is caused by the computing activity. It can, however, be reduced by circuit design techniques.

Whenever a logic gate changes state, power is consumed. The state change can be due to the essential logic value changes as well as due to glitches. The latter are not necessary and their elimination is the subject of this paper. The methods described here are based on the authors' research discussed in several papers [36]–[40] and theses [34], [35].

II. Background

The reasons for the glitch activity in a digital circuit are the hardware delays. In this section, we give the background on gate delays and the mechanisms through which glitches are produced or can be eliminated.

A. Delay of a CMOS Gate

There are many ways of combining transistors to perform the logic functions such as NOT, NAND, NOR, etc. We will describe the CMOS design style which is most prominent in current day technologies. A CMOS gate is constructed by a combination of MOSFETs to realize a logic function. But a MOSFET is not an ideal switch. When *open* it provides a large but finite resistance between its source and drain terminals. When *closed* it provides a small nonzero resistance. For a CMOS gate, the output signal change follows the input change with a certain delay. First, the closing and opening of MOSFETs in the gate depends upon the slope of input signals. Then, the output signal change requires charging or discharging of the output capacitance through a low resistance path provided by the "ON" MOSFETs. We define the delay of a gate as follows.

Definition: Gate delay is the time taken for a signal at the output of a gate to reach 50% of $V_{\rm dd}$ (logic 1 level) after the signal at the input of the gate has reached 50% of $V_{\rm dd}$.

Gate delay is a function of the amount of resistance and capacitance in the current path. A MOSFET when closed (ON state) offers a finite resistance $R_{\rm on}$ that is a function of the width and length of the device. Since gate delay is given by $R_{\rm on} \times C_L$, where C_L is the load capacitance, it can be varied by changing the width and length of the transistor [17], [32], [47]. For example, a NAND gate output rises due to current flow in its *p*FETs. Hence, the delay of the NAND gate for a rising output transition, can be altered by changing the sizes of *p*FETs. To increase the delay, we increase the resistance of a transistor by increasing its length. Similarly the output delay for a falling transition can be varied by changing the lengths of the *n*FETs. The delay can also be reduced by increasing the widths of transistors. The delay is effectively changed by manipulating the width and length of the transistors in the gate. Note that it is possible only to manipulate the overall delay of a gate but not the individual delays along different paths through it. For instance, the delay of a gate when one input transitions cannot be independently controlled without altering the delay when the other input transitions. These delays are inter-related.

Manuscript received November 28, 2004; revised February 05, 2006. First published August 18, 2009; current version published September 23, 2009. This work was supported in part by the National Science Foundation Grant CCR-9988239.

T. Raja is with the nVIDIA Corporation, Santa Clara, CA 95050 USA.

V. D. Agrawal is with the Department of Electrical and Computer Engineering (ECE), Auburn University, Auburn, AL 36849 USA (e-mail: vagrawal@eng.auburn.edu).

M. L. Bushnell is with the Department of Electrical and Computer Engineering (ECE), Rutgers University, Piscataway, NJ 08854 USA.

Digital Object Identifier 10.1109/TVLSI.2009.2027567

1063-8210/$26.00 © 2009 IEEE

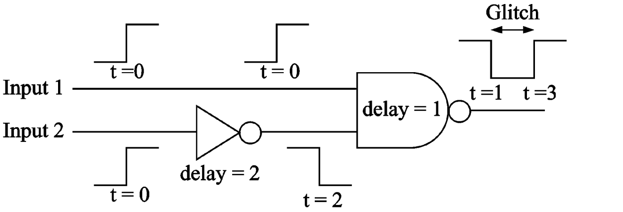


Fig. 1. Circuit showing the formation of glitches. The inverter has a delay of 2 units and the NAND gate has a 1 unit delay. Due to differing arrival times at the inputs of the NAND gate, the output produces a glitch consisting of two transitions.

B. Glitches and Dynamic Power

Every signal transition consumes a finite amount of energy. For the correct functioning of a logic circuit, every signal net needs to transition at most one time in one clock cycle. But in reality, the gate outputs transition more than once and these unnecessary transitions are called *glitches*. These transitions consume energy and are quite unnecessary for the correct functioning of the circuit. Glitch power consumption can be as much as 40% [3, p. 493] or higher [33, p. 45] as compared to the overall power consumption and it is advantageous to eliminate glitches from circuits as power consumption is critical in today's chips. Glitches arise due to the differences in the arrival times of signal transitions at the inputs of a gate.

Definition: Differential delay is the maximum difference in the signal arrival times at different inputs of a multi-input gate.

Consider the circuit shown in Fig. 1. The signal arrival at the lower input of the NAND gate is always 2 time units later than the signal arrival at the upper input due to the inverter in its path. Thus the differential delay at the NAND gate is 2. This differential delay makes the output of NAND gate transition twice when in reality, it should have no logic transition at all. These extra transitions form a glitch and waste energy.

There have been many techniques proposed to eliminate the glitches. In *delay balancing*, the inputs are made to arrive at the same time by inserting extra delay buffers on selected paths [13]–[15], [33], [41]. In *hazard filtering*, the gate delay is made greater than the differential delay at the inputs of the gate to filter the glitch [1]. However, both these approaches lead to greedy algorithms and do not yield globally optimum results. In *gate sizing*, every gate is assumed to be an equivalent inverter with the gate size as a design variable [4]–[9]. The goal of this technique is to solve for the equivalent inverter sizes such that the glitches are eliminated. *Transistor sizing* is similar to gate sizing but treats every transistor's size as a variable and tries to solve for the optimal sizes for all transistors in the design for glitch-free operation [16], [18], [42], [43], [48], [49]. Gate sizing and transistor sizing try to balance delays for eliminating glitches by sizing the devices accordingly but gate delay varies nonlinearly with device sizes [34]. This leads to nonlinear convergence problems for both these techniques.

One way to get around the nonlinearity is to solve the problem in two stages. First stage would be to use *linear programming* where the gate delays are first treated as independent variables and the optimum delays are found by solving a *linear program* (LP) [2], [34], [36]. The second stage does sizing of each gate to realize the delay of the LP solution [38], [40]. Thus, the nonlinearity is localized to each gate and convergence is not a major issue. However, a problem with this technique is that it inserts delay buffers in the circuit. These extra inserted elements also consume power themselves and hence reduce the achievable power savings. In all of the above techniques, the problem of buffer insertion arises due to the conventional gate design. The conventional CMOS gates have a single delay, no matter which input of the gate causes the transition. Although, some modern libraries might contain gates with differing input-output delays, these delays cannot be controlled independently. Raja *et al.* proposed a *minimum dynamic power* LP technique using a new gate delay model, where the input delay of the gate can also be varied [37]. This makes the gate delay different for different input-output paths through the gate. The advantage of this gate model is that glitches can be completely eliminated from the circuit without the insertion of any delay buffers, thus achieving more power saving.

C. Present Contribution

A novel implementation of the last technique is the focus of this paper. We describe the formulation of an LP using variable input-output gate delay model with an overall delay constraint. This is a variation of the *minimum dynamic power* LP described in previous publications [35], [37]. In speed-critical designs, the critical path delay should not exceed some given value we will refer to as *maxdelay*. If the delay requirement is stringent, then the circuit designed using the *minimum dynamic power* LP may not meet the specification. In such cases, we provide the designer with another version of the LP in which, the design is optimized by adding buffers at selected nets in addition to the input delay elements. We call this the *delay specification LP*.

We describe three possible ways of implementing the variable-input delay gate and its application to low power design. The technique for the sizing of the new gates and delay elements is also discussed in recent publications [38]–[40]. It is to be noted that this technique is different from *input reordering* techniques, because the amount of *delay difference between two paths through a single gate* (a parameter we define in Section III-G. as u_b) that can be achieved by *input reordering* is much smaller compared to that achieved in the designs described here.

In general, dynamic power consumption in circuits can be described as the product of *number of transitions* (N_t) and *average power per transition* (P_t). Our technique reduces glitches in the circuit thereby reducing N_t. It is true that the quantities N_t and P_t may not be completely independent and reducing N_t alone can potentially increase P_t. However, we would like to point out that the final goal is total power reduction and that a large reduction in N_t can offset the slight increase in P_t resulting in overall power savings. As an evidence in support of this statement, we provide the power consumption results from c7552 circuit designed using our technique in Section V. The power for this circuit was estimated using a circuit-level (Spice type) simulator, which takes both N_t and P_t into account. Even if we assume that P_t could have increased in the design, there is substantial reduction in the total power.

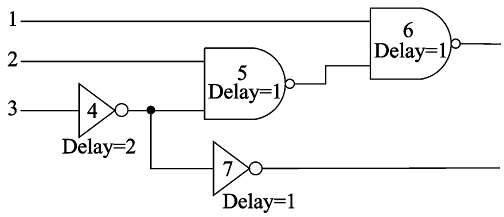


Fig. 2. Example circuit.

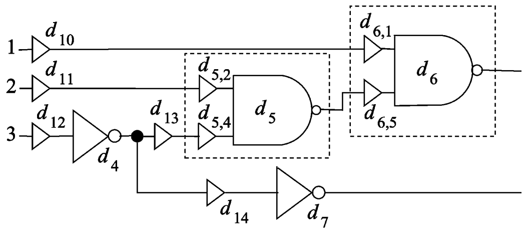


Fig. 3. Delay model for the circuit of Fig. 2.

Other power reduction techniques such as *dual-*$V_{\rm dd}$ and *dual-*$V_{\rm th}$ [32], [47] can be used in conjunction with the present technique to improve the design quality. Note that these techniques reduce P_t and do not change the power saving from our technique which primarily targets N_t. Recent papers [23]–[26], [31] combine the glitch elimination and leakage reduction through mixed integer linear programs.

III. Delay Specification LP

We use the example of a simple circuit to explain the new delay specification LP formulation. Consider the combinational circuit of Fig. 2. Traditionally, for the purpose of gate sizing the circuit is generally viewed with each gate having a single *inertial delay* and all input/output (I/O) paths through the gate are assumed to have the same delay. We redefine the gate delay. A gate can now be viewed as having one *basic inertial delay* and a set of input delays for the I/O paths running through the gate. This is illustrated in Fig. 3.

Each gate can be assumed to have a single inertial delay variable with additional input delays modeled by delay elements at the inputs of the gate. The inertial delay and input delays of a CMOS gate are not independent. In the LP we treat them as independent variables, which are bounded by a feasibility constraint defined in Section III-G so that the delays can be realized in practice. There are *buffer variables* inserted at *primary inputs* (PIs) and fan-out stems. These buffer variables will be used by the LP for adjusting the differential delay when the constraint cannot be met by input delay elements alone. We assert that *these input delay elements and extra buffer elements are only for analysis purposes and are not actual extra components in the circuit.* The LP consists of variables that define various parameters in a circuit, constraints on the variables and an objective function that needs to be minimized. This LP produces the *minimum dynamic power* consuming circuit for given feasibility constraint and *maxdelay*. We describe the LP for the circuit shown in Fig. 3.

A. Variables

The variables for a circuit are as follows [34], [36].

- *Basic Inertial Delays of Gates:* d_4, d_5, d_6, d_7.
- *Gate Input Delays:* $d_{i,j}$ is the extra delay on the path from the fanin gate j to gate i. For instance, $d_{5,4}$ is the extra delay of the path through gate 5 while arriving from gate 4. This models the differences in delays of various I/O paths through the gate. Its minimum value is 0.
- *Buffer Delays:* d_{10}, d_{11}, d_{12}, d_{13}, d_{14} are the buffer delay variables. These buffers are used by the LP to balance delays that cannot be balanced by just using the input delay variables. Their minimum value is 0.
- T_i is the latest time of signal change at the output of gate i. The reference for T_i is the time instant when primary inputs change. Thus, for primary inputs, $T_i = 0$. However, any time skews or ambiguities at primary inputs can be modeled by assigning nonzero values to T_i.
- t_i is the earliest time of signal change at the output of gate i. The reference for t_i is the same as for T_i and for primary inputs, $t_i = 0$. Also, any skews or ambiguities at primary inputs can be modeled by assigning nonzero values to t_i.
- The interval (t_i, T_i) is defined as a *timing window* during which the signal i (primary input or output of a gate) can change.

B. Constraints on Delays

The following constraints set the lower and upper bounds on the variables:

- lower bounds on gate inertial delays are set to 1; The actual value of this time unit will depend on the specific technology used;
- lower bounds on gate input delays are set to 0;
- lower bounds on buffer delays are set to 0;
- we also set an upper bound u_b on the gate input delays (see Section III-G).

C. Buffer Delay Constraints

These constraints are for buffer variables and gates with a single fan-in. They ensure that the minimum and maximum arrival times at the input of a gate are propagated to the next stage. For example, at the output of the buffer d_{10} at primary input 1 in Fig. 3, we have

$$t_{10} \leq t_1 + d_{10} \qquad T_{10} \geq T_1 + d_{10}.$$

D. Glitch Suppression Constraints

These constraints ensure that the timing window for signal transitions at every gate output does not exceed the inertial delay [1], [34], [36]. Consider gate 6 in Fig. 3. The constraints for it are given as

$$\begin{aligned} t_6 &\leq t_{10} + d_{6,1} + d_6 \\ t_6 &\leq t_5 + d_{6,5} + d_6 \\ T_6 &\geq T_{10} + d_{6,1} + d_6 \\ T_6 &\geq T_5 + d_{6,5} + d_6 \\ d_6 &> T_6 - t_6. \end{aligned}$$

E. Maxdelay Constraints

These are the constraints that specify the speed of the circuit. Here *maxdelay* is a parameter given by the designer to specify the maximum path delay in the circuit. The delay of the slowest path is bounded by the latest time of arrival of a signal transition at a PO. Hence, for every PO of the circuit in Fig. 3 we have

$$T_6 \leq maxdelay \qquad T_7 \leq maxdelay$$

F. Objective Function

The following objective function reduces the insertions of buffers into the circuit, while achieving the required overall delay:

$$\text{Minimize} \sum_{j \in buffers} d_j.$$

Since the total extra power of the circuit is proportional to the number of buffers added, the exact objective function should minimize the number of buffers added by the LP. However, this becomes a nonlinear objective function and an LP solver cannot handle a nonlinear objective [2]. Hence, we use the linear version of the objective function as shown.

G. Feasibility Constraints

The main issue here is the upper bound on the delay of gate I/O elements. The idea behind this formulation is to design a gate that can have different delays along different I/O paths through it. If unconstrained, the linear program may give gate input delays that are too large to be realized. Every technology has a limit on the amount of flexibility that designer are allowed. This limit is based on the feasibility of designing the gate input delays for the technology used at the transistor and layout levels. We call this the*feasibility condition*. For example, the feasibility constraints for gate 6 in Fig. 3 are

$$d_{6,1} \leq u_b \qquad d_{6,5} \leq u_b.$$

These allow the gate input delay to be varied up to a value of u_b by the program. This value is a design parameter and is specific to the design technology in which the circuit is being designed. Given a feasibility value, we can use the delay specification LP to design the *lowest power consuming circuit at the given delay specification*. The LP guarantees the minimum dynamic power condition and the delay specification. This LP inserts fewer buffers than the prior techniques and hence, achieves more power saving. Hence, the dynamic power consumed may not be *absolute minimum* because a finite number of buffers could have been added to meet the delay specification.

H. Solution Curves

Let us consider the solution space explored by the proposed LP and the previous LP techniques [34], [36]. Consider the illustration of Fig. 4. The solid curve shows the power-delay tradeoff in previous methods with conventional gate design [39]. The circuit designs given by the points on this curve are the fastest minimum power circuits for $u_b = 0$. If we increase u_b and use the proposed delay specification LP, the power delay curve shifts to one of the dotted curves shown in the figure. For a given *maxdelay* the point on a dotted curve is the solution with minimum dynamic power for the corresponding u_b. The maximum u_b allowed for the technology is given by $\text{Max } u_b$. The region of feasible implementations is shown as the shaded region in the figure. Every point in the shaded region is a possible minimum dynamic power solution for a given u_b and a given *maxdelay* requirement. The lower boundary of this region represents zero-buffer designs. If the gate input to output delay variability (u_b) is allowed to increase indefinitely, all glitch suppression conditions would be satisfied without increasing the critical path delays. The point $u_b = \infty$ in Fig. 4 has the same speed as the original circuit prior to power minimization. Any increase in dynamic power over this minimum is caused by the delay buffers inserted to satisfy the technology limit on the allowable u_b [37]. The delay specification LP proposed in this paper effectively exploits the power-delay solution space. Note that for any given *maxdelay*, the delay specification LP solution consumes less power than the $u_b = 0$ solution [34], [36].

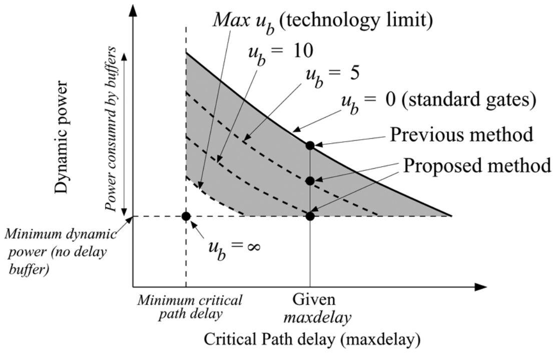
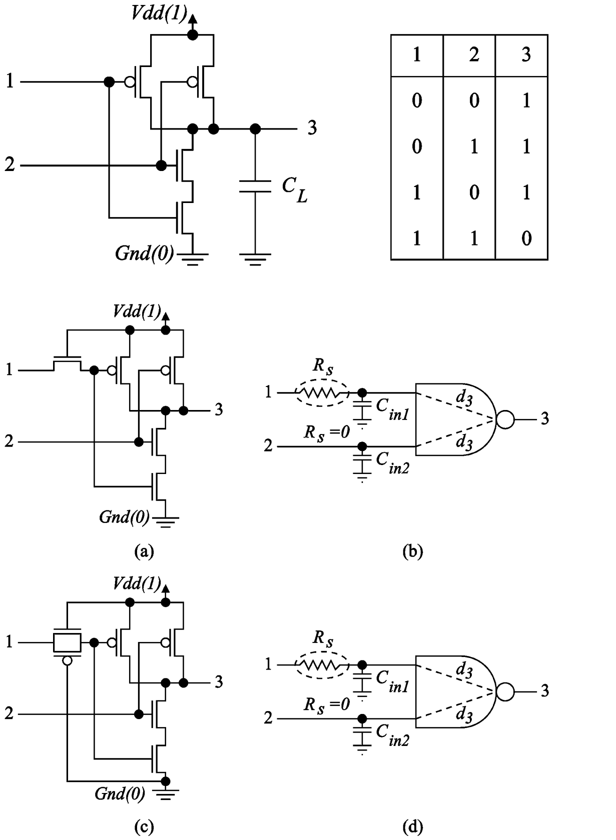


Fig. 4. Power-delay curves for a previous method [34], [36] and for the delay specification LP proposed here.

In the proposed *delay specification LP* of Section III, if we use the conventional logic gates, i.e., $u_b = 0$, we get the same design as obtained by previous methods [39]. This design will be buffer-less only if some large *maxdelay* was specified. If we use nonzero u_b, the power delay tradeoff curve shifts down to one of the dotted curves shown in Fig. 4. This is because the design with delay specification LP requires fewer buffers than the previous technique (shown by the solid curve) and hence consumes less power. Thus, by increasing u_b for a design, we reduce the power consumption of any given solution from the solid curve to one of the dotted curves for the same *maxdelay*. As the delay variability upper bound u_b is increased to a higher value, the design consumes less power for the same *maxdelay* as shown by the dotted-line curves. However, every technology has some maximum u_b and it is difficult to construct gates with any higher delay variability. The limiting power delay tradeoff curve is labeled with $\text{Max}\, u_b$. Thus, by choosing any u_b between 0 and $\text{Max}\, u_b$ and using the proposed delay specification LP, the entire design space can be explored.

IV. Proposed Gate Design

Several delay elements have been discussed in the literature [21], [22], [27], [30], [45], [46]. Our focus here is on indepen-



1	2	3
0	0	1
0	1	1
1	0	1
1	1	0

Fig. 5. Schematic of the proposed variable input delay gate: a conventional 2-input CMOS NAND gate characterized by a single output delay (top), and two ways of varying input delays by always-on *n*MOS pass transistor (center) and by always-on CMOS transmission gate (bottom).

dently changing the delays of interconnects leading to a gate inputs without inserting any gates or buffers, which might consist of their own pull-up and pull-down paths and node capacitances.

As described above, it is advantageous to design a gate with differing delays along different input-output paths through the gate. In other words, it is advantageous to design gates with high u_b, such that fewer buffers will be needed to meet the delay specification. We call such a gate *variable input-delay gate*. In this section, we propose a transistor level implementation of the gate and its characteristics [35]. Consider a two-input NAND gate shown in Fig. 5. Suppose, the delay of path 1 to 3 is $d_{1\to3}$ and that of 2 to 3 is $d_{2\to3}$. Then

$$d_{1\to3} = R_{\text{on}} \times C_{\text{in1}} + d_3 \qquad d_{2\to3} = R_{\text{on}} \times C_{\text{in2}} + d_3$$

where C_{in1} and C_{in2} are the input capacitances seen at the inputs of the gate and R_{on} is the series resistance of the ON transistors in the previous stage. The intrinsic inertial delay d_3 of node 3 is independent of the input causing the output to change. It depends upon the total capacitance C_L of node 3 and the resistance of ON transistors within the gate. A conventional CMOS gate (top figure) is characterized by a single delay normally assigned to the output. We examine the following four different implementations.

- ***Input capacitance manipulation*** is the technique by which C_{in1} is increased without altering C_{in2}. This is achieved by increasing the sizes of the transistors connected to input 1 such that $C_{\text{in1}} > C_{\text{in2}}$. Now the delays are

$$\begin{aligned} d_{1\to3} &= R_{\text{on}} \times C_{\text{in1}} + d_3 \\ d_{2\to3} &= R_{\text{on}} \times C_{\text{in2}} + d_3 \\ d_{1\to3} &> d_{2\to3}. \end{aligned}$$

The problem with this implementation is that the resistances of ON transistors in the series path are interrelated and hence the output delay d_3 is also altered. The formulation becomes nonlinear.

- ***Resistance with a single nMOS pass transistor*** can be added in series to the path in which extra delay is desired. This scheme is shown in Fig. 5(a) and (b). This *n*MOS transistor is always ON and hence adds a series resistance R_s to the path 1 → 3. The delays are

$$\begin{aligned} d_{1\to3} &= (R_{\text{on}} + R_s) \times C_{\text{in1}} + d_3 \\ d_{2\to3} &= R_{\text{on}} \times C_{\text{in2}} + d_3 \\ d_{1\to3} &> d_{2\to3}. \end{aligned}$$

The resistance R_s can be controlled by changing the size of the *n*MOS transistor. The delay along the path 1 → 3 can be controlled independent of the delay along path 2 → 3. Hence the gate has different delays along different I/O paths. A disadvantage of this design is that the *n*MOS pass transistor degrades the signal when it passes a logic 1. This causes the transistors in the next stage to have a higher source-drain leakage current. This disadvantage can be alleviated by using a CMOS pass transistor as described below.

- ***Resistance with a CMOS pass transistor (transmission gate)*** can be added to introduce the extra resistance in the path as shown in Fig. 5(c) and (d) The inserted CMOS pass transistor contains both *n*MOS and *p*MOS transistors that are always ON. This does not degrade the signal but has the disadvantage that it adds an additional transistor. Moreover, gate leakages of the two devices of the CMOS transmission gate can provide power to ground paths. Device-level simulations in Section V do not show this to be a concern for the 0.25-μm CMOS technology. But, for finer devices, technology solutions to this problem will be required [10].

We should point out that the delays obtained either by single *n*MOS transistor or by the CMOS transmission gate are static. Because the control inputs of these devices are permanently tied to fixed supply or ground voltages. This is quite different from the gate triggering method of glitch suppression [28], [29], where pass transistors are controlled by dynamic clock-like signals.

- ***Resistance with a feedthrough resistive cell*** is a technique of adding the resistance using a *polysilicon* serpentine resistor overlaid with *silicide blocking*. This is the standard way of creating a resistance in an analog circuit. The advantage of these cells is continuous controllability of resistance rather than the discrete control provided by transistors [45], [46]. We did not use this method because of its large area overhead.

A. Design Issues

We selected the single *n*MOS pass transistor for our design. Some of the underlying reasons are low area overhead, simplicity of integrating it in the a cell library, and the ease of incorporating it in circuit-level simulation. A previous paper has discussed the CMOS transmission gate delay element [30] and reported that it provides good delay controllability but has poor signal integrity. Since we did not require a standalone delay element, the delay producing transistors were integrated with logic gates whose input delays and output signal integrity are locally analyzed and adjusted.

There are several design issues regarding this variable input delay gate design. The delay along a path is varied by changing the series resistance, which is a function of the length of the transistor/transmission gate and hence the delay along the line can be altered by changing the length of the extra transistor/transmission gate.

- This transistor cannot be infinitely long because that would increase the voltage drop across it and cause signal integrity issues at the output of the gate. Hence there is a realistic limit to the length of the transistor added and this determines the maximum differential delay that can be added. Raja *et al.* describe this as the *gate differential delay upper bound* u_b in their low power design [37]. This parameter u_b is related to the technology the gate is implemented in and is called the *feasibility condition*. Our design analysis determined a u_b of 10 delay units for the 0.25-μm fabrication process we used [35], [38], [40]. The *delay unit* is the minimum gate delay for the technology and is the delay of an inverter feeding into another inverter of the same size.
- The logic 1 state is degraded by the threshold voltage of *n*MOS transistor. If the signal does not drive the transistors in the logic gate into cutoff, there will be increased leakage. This problem can be alleviated by using a CMOS transmission gate instead of a single transistor. The effect of increased leakage is shown in the results section.
- The placement of the series transistor with respect to the routing capacitance also needs to be examined. If the routing capacitance is small it does not matter where the transistor is placed in the path. But if the routing capacitance is large, then the delay at the input of the gate changes as the transistor is moved along the path as it sees a different capacitance at every stage [17]. We have inserted the transistor at the end of the routing path in our designs [35].

More details on these design issues may be found in recent papers [38]–[40].

V. Results

In this section, we present an application of the new gate design in implementing custom circuits for minimum dynamic power.

- ***Unoptimized Example Circuit.*** Consider the simple example circuit of Fig. 2. Assume that the delays of all gates are the minimum allowed by the technology through sizing of transistors in gates appropriately. We observe that the differential delay, at gates 5 and 6, exceeds the inertial delay and we expect potential glitches. The circuit was simulated for rising signals at all three inputs using *Spectre* analog simulator from *Cadence* [12]. The results are shown in Fig. 8. As expected, gates 5 and 6 transition 2 and 3 times, respectively.

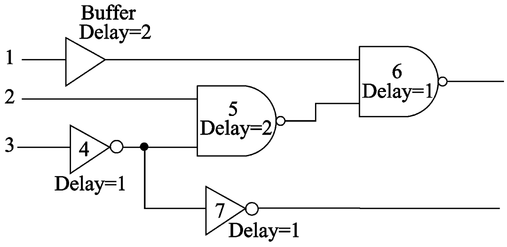


Fig. 6. Design with two-inverter delay buffer.

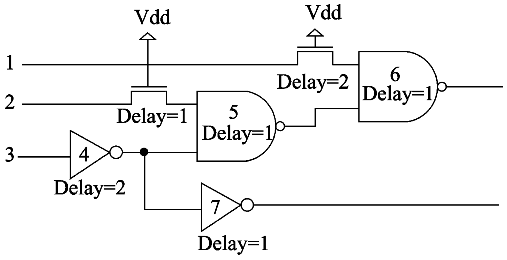


Fig. 7. Design with proposed input delays.

- ***Buffer Optimized Circuit.*** The buffer optimization using conventional gates requires the use of one buffer for the circuit to operate at the same speed [2], [36]. The optimized circuit with the buffer is shown in Fig. 6. It is implemented using two CMOS inverters and has an overall delay of 2 delay units. The buffer optimized circuit was simulated for the same vector-pair as the unoptimized circuit. As expected, the optimization eliminated all glitches as shown in Fig. 8. However, the buffer optimization requires that the transition of input 1 should pass through the buffer. This increases the total number of transitions in this circuit.
- ***Low-Power Design with Proposed Gate.*** When variable input-delay gates are used, the optimized circuit is shown in Fig. 7. We have used the single *n*MOS transistor implementation here but any of the proposed designs could have been used. Circuit-level simulation for the same vector pair is shown in Fig. 8. The glitches at the outputs of gates 5 and 6 are eliminated in this optimized design as well.

A. Energy Consumption

During the simulation for the three circuits described above, we measured the supply current for the given input vectors and computed the energy. The results are shown in Table I. The simulations were done with *Spectre* analog simulator from *Cadence* [12]. As recorded in the table, the unoptimized circuit consumes 800 fJ, the buffer optimized circuit consumes 550 fJ and the new variable-input delay gate circuit consumes 300 fJ. Thus, the energy saving of the new design is 62.5% with respect to the unoptimized circuit. The new gate design achieves substantially higher power saving than the buffer optimized design. The total power obtained from the simulator includes the

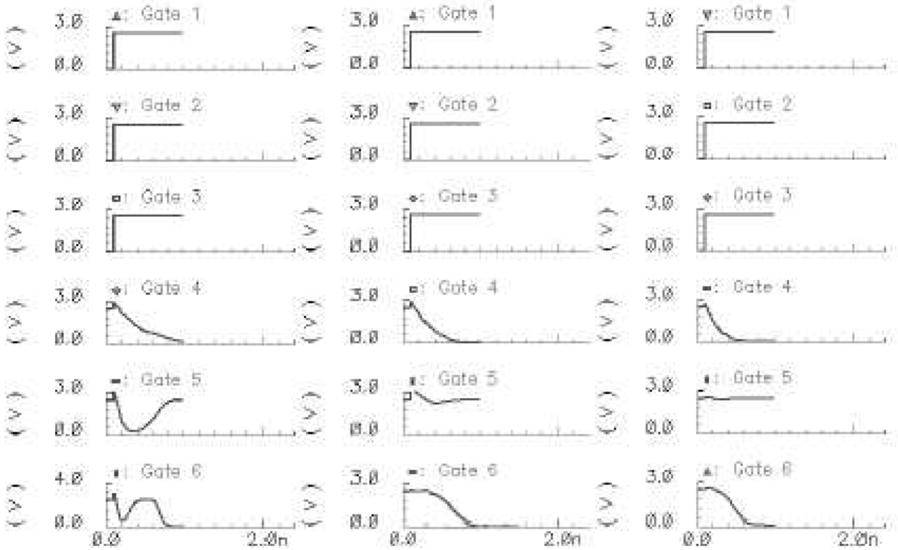


Fig. 8. Circuit simulation of vectors $000 \rightarrow 111$ for (left to right) circuits of Figs. 2, 6, and 7.

TABLE I
SIMULATION OF THE THREE DESIGNS OF THE EXAMPLE CIRCUIT FOR INPUT $000 \rightarrow 111$

Circuit	Logic activity		Energy consumed		Leakage I_{DDQ}	
	Gate transitions	Reduction	Total	Reduction	Vector 000	Vector 111
Figure 2	8	0.0%	800fJ	0.0%	38.1pA	60.6pA
Figure 6	5	37.5%	550fJ	31.3%	–	–
Figure 7	3	62.5%	300fJ	62.5%	38.1pA	60.9pA
Circuit of Figure 7 with CMOS Transmission gates					38.1pA	60.7pA

short circuit and leakage components as well. However, for the 0.25-μm CMOS technology used, dynamic power dominates as discussed in Section V-B. Table I also shows a good correlation between the reduction in the number of transitions and power saving. The power savings of the optimized circuit are due to the following two reasons:

- glitches have been removed, which makes the proposed example circuit save power over the unoptimized circuit;
- no buffers have been added, which does not add extra transitions in the circuit, thereby making the new design consume less power than the buffer-optimized circuit.

B. Leakage Current

The introduction of an *n*MOS pass transistor degrades the signal at the gates of the transistors. This increases the leakage current of the circuit and may even drive the transistors out of cutoff. The current flowing in the steady state is called the *quiescent current* (I_{DDQ}) and is due to the leakage through OFF transistors. The quiescent current is a function of the input vectors at the primary inputs (PIs) of the circuit. To analyze the relative effect, we simulated circuits with a pair of input vectors allowing the transients to completely settle down following each vector. The three circuits simulated for leakage were the unoptimized circuit of Fig. 2, the optimized circuit of Fig. 7, and another optimized circuit obtained by replacing the *n*MOS pass transistors in Fig. 7 with CMOS transmission gates. The leakage currents for the vector 000 showed no change for the three circuits as in this state the *n*MOS transistors are passing logic 0, which is not degraded (see Table I). For vector 111, however, there was an increase of 0.45% in leakage due to the *n*MOS pass transistors. The circuit with the CMOS transmission gates had an increase of only 0.2%. This increase is not due to the degradation of the signal but is due to the leakage path added from V_{dd} to Gnd through the sidewall capacitance. This is a very minor increase for the 0.25-μm fabrication technology but further analysis needs to be done for more recent technologies.

C. Benchmark Circuits

We optimized several ISCAS'85 benchmark circuits for dynamic power. The results in Table II compare the designs done with the new variable-input delay gates to original versions of circuits and those optimized using conventional gates [34], [36]. For each method, two optimized designs were created, one where no increase in the overall delay (*maxdelay*) was permitted and the other where the overall delay was allowed to increase to twice that of the original design. The "original designs" were optimized not for power but for speed in the given 0.25-μm CMOS technology. Each design had the smallest possible delays along critical paths as well as all other paths in the given technology. We analyzed the circuit at the logic level assuming that it is possible to design such unit or constant delay gates, i.e., gates sized to feed the fan-outs.

For each circuit, first an original version (not optimized for glitch removal) was created as a reference. This version used the fastest gates available in our 0.25-μm CMOS technology. These gates have larger transistors and typically consume more power. This design functions somewhat similar to a unit-delay logic circuit that is generally known to consume more power [41].

TABLE II
Dynamic Power Consumption of Custom Designs of ISCAS'85 Circuits Estimated by Logic Simulation. the Original Designs Are the Highest Speed Designs in the 0.25 μ CMOS Technology Used. The "Conv. CMOS Gate" Design Is Done Using an LP Formulation That Uses a Single Delay Variable for a Gate and Does Glitch Elimination by Inserting Buffers as Delay Elements [34], [36]

Circuit	*maxdelay*	Orig. design	Variable input-delay gate			Conv. CMOS gate [34,36]		
		Norm. power=1	Added	Norm. power		Added	Norm. power	
		No. of Trans.	Trans.	Av.	Peak	Trans.	Av.	Peak
c432	1.0	784	291	0.69	0.66	380	0.72	0.67
	2.0	784	98	0.65	0.55	264	0.62	0.60
c499	1.0	1,364	105	0.86	0.84	192	0.91	0.87
	2.0	1,364	86	0.71	0.65	0	0.70	0.66
c880	1.0	1,802	174	0.58	0.45	248	0.68	0.54
	2.0	1,802	154	0.56	0.45	136	0.68	0.52
c1355	1.0	2,196	550	0.48	0.42	896	0.58	0.48
	2.0	2,196	410	0.44	0.39	768	0.57	0.48
c1908	1.0	3,878	206	0.56	0.46	876	0.69	0.59
	2.0	3,878	192	0.55	0.45	280	0.59	0.44
c2670	1.0	5,684	436	0.70	0.56	628	0.79	0.65
	2.0	5,684	380	0.69	0.57	140	0.71	0.58
c3540	1.0	7,822	677	0.57	0.46	956	0.64	0.44
	2.0	7,822	642	0.54	0.43	560	0.58	0.46
c5315	1.0	11,308	1,310	0.57	0.48	1,120	0.63	0.52
	2.0	11,308	1,361	0.55	0.46	684	0.60	0.45
c6288	1.0	10,112	2,854	0.91	0.87	1,176	0.40	0.36
	2.0	10,112	1,815	0.21	0.16	480	0.36	0.34
c7552	1.0	15,512	1,439	0.28	0.24	1,464	0.38	0.34
	2.0	15,512	1,406	0.27	0.24	444	0.36	0.32

Dynamic power consumption was estimated by an event-driven simulator, which assumed that each gate has the same delay and that the power consumed per signal transition is proportional to the number of fan-outs. The simulator uses a glitch-filtering procedure [11]. Thus, whenever a new event is scheduled such that a previously scheduled event on the same signal is still pending, then both events are canceled. Estimates of peak and average power were obtained for a set of vectors. These vectors were generated for a complete or almost complete stuck-at fault coverage. It is assumed that such vectors provide appreciable logic activity and hence a reasonable estimation of power. In Table II, the power of original circuits is normalized to unity and transistor counts for all circuits are given. Power estimation for all other designs (discussed below) was similar but used the delays obtained from the LP.

Next, we redesigned the circuits with variable-input delay gates described in previous sections. An LP determined the input and output delays for all gates under an input differential delay constraint of $u_b = 10$ (see Section IV-A). Each circuit was designed for two overall delays, $maxdelay = 1$ and 2, respectively, normalized with respect to the corresponding reference design. Columns 4–6 of Table II show the number of transistors added (see next paragraph) and the power consumption normalized with respect to the corresponding original design. To meet the *maxdelay* constraint, some circuits used delay buffers. But in most cases no buffers were required. In the linear program optimization, an upper bound (u_b) is used on the input differential delay that can be achieved. This upper bound is a technology parameter and is determined through actual design and simulation of gates. When the circuit topology requires very large differential delays, delay buffers must be used to satisfy the glitch removal conditions. The linear program, however, keeps the number of such buffers to a minimum. The circuit c6288 is a typical case where a large number of buffers were essential. Since each delay buffer has two inverters, which provide additional node capacitances to be charged and discharged during operation, extra power is consumed. The $maxdelay = 2$ design of c6288 did not require buffers and all glitch removal conditions were satisfied by the gate input delays.

The added transistors are mostly for the *n*MOS transmission gates inserted at gate inputs to realize the input delays given by the LP. Gate inputs were treated as symmetric although, in general, it is possible to take advantage of the input asymmetries in implementing the differential delays. As explained above some circuits needed delay buffers. Each buffer was implemented with four transistors (two inverters) included in the counts given in column 4 of Table II. Inverters in buffers have drain to source leakage. In addition, all transistors have gate leakage. Roughly, the leakage of the circuit may increase in proportion to the transistor overhead, which, on an average, is 13.3% for $maxdelay = 1$ designs and 10.8% for $maxdelay = 2$ designs. However, it should also be noted that all transistors in the circuit are not sized equally and the leakage depends upon their sizes as well. If the designs were done with CMOS transmission gates, the added transistor counts will double only for transmission gates and will remain unchanged for buffers.

The last three columns of Table II provide a comparison with an alternative method in which "conventional" CMOS gates were used. Here, each gate was designed for its own almost equal input-to-output delays and is characterized by a single delay. The circuit design is obtained by another linear program [34], [36] that inserts delay buffers in most cases to satisfy the glitch elimination conditions under the single gate-delay restriction. With the exception of a few, most circuits consumed more power than the variable input delay gate design. The numbers of added transistors in column 7 are due to the delay buffers, each requiring 4 transistors. Thus, c7552, required 366 buffers implemented with 1464 transistors for the design with $maxdelay = 1$.

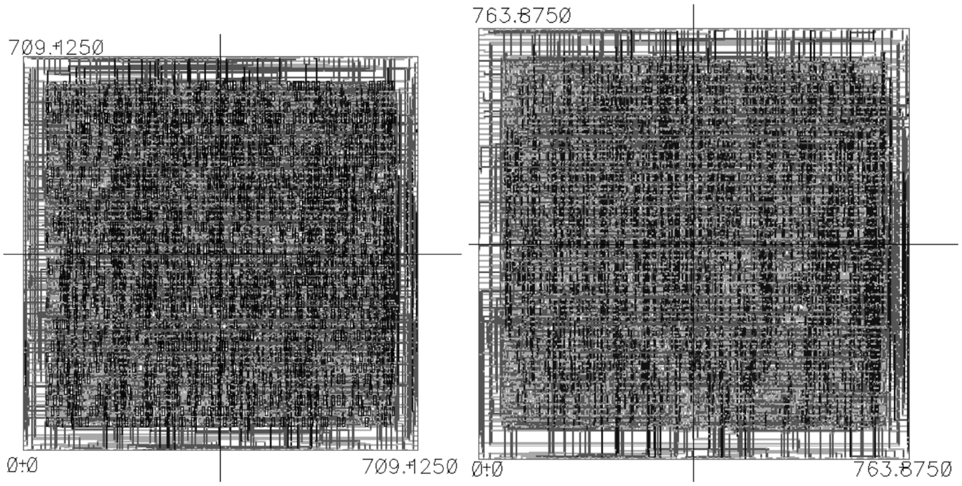


Fig. 9. 0.25-μm custom CMOS layouts of unoptimized (left) and optimized c7552 circuits (right).

TABLE III
DYNAMIC POWER CONSUMPTION OF C7552 CHIPS ESTIMATED BY LOGIC SIMULATION

Circuit	*maxdelay*	Unopt. design	Variable input-delay gate			Conv. CMOS gate [34, 36]		
		Norm. power=1	Added	Norm. power		Added	Norm. power	
		No. of Trans.	Trans.	Av.	Peak	Trans.	Av.	Peak
c7552	1.0	15,512	1,439	0.42	0.34	1,464	0.49	0.35

The circuit c499 with $maxdelay = 2$ is an interesting case in Table II. The conventional gate design does not require any delay buffers because all glitch elimination conditions could be satisfied by the original single-delay gates. The overall power consumption is lower (though marginally) than that for the variable-input delay gate design. This shows that the two types of gates offer different design alternatives and in some cases, like this one, conventional gate may give a lower power design.

D. Chip Design and Total Power

We did the physical design of the ISCAS'85 benchmark circuit c7552. First, an "unoptimized" design was created. This circuit contained 3827 gates and was implemented with 15 512 transistors. We used gates with smallest size transistors as compared to the fastest gates used in the "original" design of the previous subsection. The unoptimized circuit, therefore, is slower but consumes less power. Its physical layout was done by the *Cadence layout editor*. We redesigned the circuit using the proposed variable input delay gates and that design contained 1435 *n*MOS transmission gates and one delay buffer, requiring 1439 extra transistors. This design is the $maxdelay = 1$ version of c7552, shown in Table II (columns 4 to 6). A third design using the conventional CMOS gates (last three columns in Table II) was also implemented. It required 366 delay buffers or 1464 extra transistors added to the unoptimized version. All three designs were implemented in 0.25-μm CMOS technology and worked at the same speed [35]. Two layouts shown in Fig. 9 are for the unoptimized design and the variable-input delay gate design. The areas of these chips are 710 μm× 710 μm and 760 μm×760 μm, respectively.

Power consumption was evaluated in two ways. First, the logic simulation method of the previous subsection was used with few differences. For the unoptimized circuit, gate delays were assumed to be proportional to fan-outs instead of being the same, and the signal activity was weighted by the node capacitance extracted from the chip layout. The circuits were simulated for a set of 156 fault coverage test vectors. As shown in Table III, the variable-input delay gate design saves 58% average and 66% peak power. In comparison with the conventional CMOS gate design using 366 delay buffers, the variable-input delay gate design consumed about 17% less average power.

These power savings, though appreciable, are lower than those estimated in Table II. The reason for the discrepancy is that our "unoptimized" design uses the smallest gates and consumes less power as compared to the "original" design, which used the fastest gates. Indeed, the "original" design is faster than the "unoptimized" design.

A second evaluation of power was done with a circuit-level simulator. The results of instantaneous and average power measurements are shown in Figs. 10 and 11. These results were obtained by a circuit-level simulator, *Spectre* from *Cadence* [12]. The estimated power here includes all components, namely, dynamic, short-circuit, and leakage. For simulation, node capacitances were extracted from the layouts. The circuits were simulated for the same set of 156 vectors. These plots show a peak power saving of 68% and average power saving of 58%, which are very close to those obtained by logic simulation (see Table III). The circuit-level simulation thus confirms that the short-circuit and leakage power components were indeed negligible, as is expected for the 0.25-μm CMOS technology, for both optimized and unoptimized circuits.

VI. CONCLUSION

The novelty of our work is in: 1) finding the best combination of path balancing and hazard filtering by linear programming to minimize the hardware and timing penalties; 2) formulating the

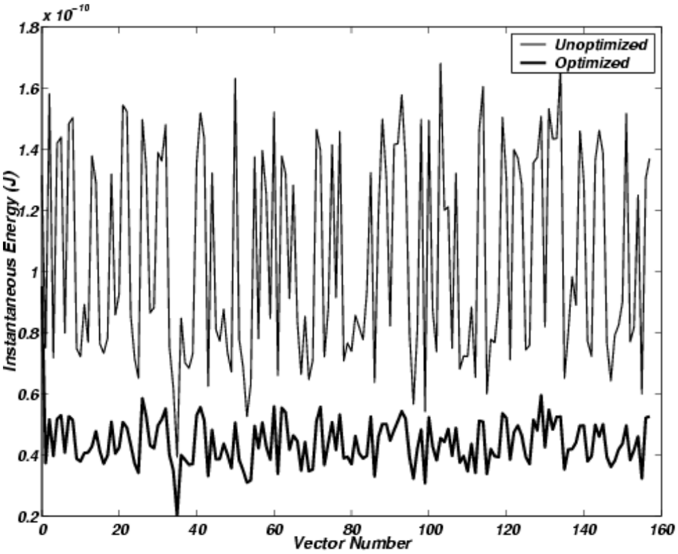


Fig. 10. Instantaneous total energy consumption in benchmark circuit c7552 for 156 vectors obtained by circuit-level simulation. A peak power saving of 68% over the unoptimized circuit is realized.

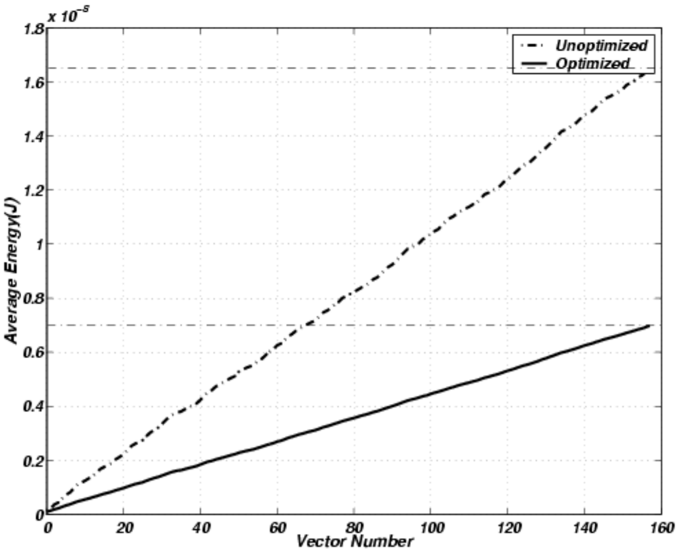


Fig. 11. Average total energy consumption of benchmark circuit c7552 for 156 vectors obtained by circuit-level simulation. The curves show cumulative power versus input vector and the slope of a curve gives the average power. This graph indicates an average saving of 58% for the optimized design.

linear program with reduced-complexity (linear in circuit size) constraint set so large circuits can be designed; and 3) using a two-step solution where gate delays are first found by the linear program and then implemented at the transistor-level so that the size-delay nonlinearity is locally (not globally) dealt with. The physical design of a circuit and its circuit-level simulation in Section V-D demonstrates the effectiveness of this technique.

We have used a new variable input delay gate, which has different delays along different I/O paths through the gate [35], [38]–[40]. Although we show significant power savings, it is well known that the delays of gates tend to change due to process and environmental variations such as temperature, fabrication impurities, etc. These variations make delays vary over a range rather than being static numbers. This can be accounted for in our technique during the LP stage, where the constraints can be modified to incorporate the maximum gate delay value in the latest time of arrival constraints and the minimum gate delay values in the earliest time of arrival constraints. One such analysis has been recently reported [19].

The procedure discussed here is independent of any operational conditions and hence the optimization is valid for all input vectors sequences. It is possible to reduce the hardware overhead of delay buffers by customizing the optimization to a subset of highly probable or worst-case vectors [20].

A possible area for future investigation is the use of the CMOS transmission gate for realizing delays in 90, 65, 45 nm, and even finer CMOS technologies. Though intended to add resistance in the charging path, the transmission gate also adds capacitance causing extra power consumption. Gate leakage of the *always-on* CMOS transmission gate can be another problem.

Higher leakage or static power is a known problem of the finer technologies where dual-supply or dual-threshold devices have been used [44]. The use of multi-threshold transistors for reduced leakage has been incorporated into integer linear programming techniques for simultaneous glitch elimination and leakage reduction [23]–[26], [31]. Recent advances in process technology also provide encouraging results on reduced leakage [10].

The application discussed in this paper is custom VLSI design. The problem of glitch-free standard cell based design of application-specific integrated circuits is also relevant and its solution requires a reformulation of the LP presented here [45], [46].

ACKNOWLEDGMENT

The authors gratefully acknowledge the diligent and painstaking work by editors and reviewers that helped them improve this paper.

REFERENCES

[1] V. D. Agrawal, "Low power design by hazard filtering," in *Proc. 10th Int. Conf. VLSI Des.*, Jan. 1997, pp. 193–197.

[2] V. D. Agrawal, M. L. Bushnell, G. Parthasarathy, and R. Ramadoss, "Digital circuit design for minimum transient energy and a linear programming method," in *Proc. 12th Int. Conf. VLSI Des.*, Jan. 1999, pp. 434–439.

[3] A. Bellaouar and M. I. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*. Boston, MA: Kluwer, 1995.

[4] M. Berkelaar, "Statistical delay calculation," in *Proc. Workshop Notes Int. Workshop Logic Synth.*, Lake Tahoe, CA, May 1997, pp. 2.1.1–2.1.4.

[5] M. Berkelaar, "Statistical delay calculation: A linear time method," in *Proc. IEEE Int. Workshop Timing Issues Specif. Synth. Digit. Syst.*, Dec. 1997, pp. 15–24.

[6] M. Berkelaar, P. Buurman, and J. Jess, "Computing entire area/power consumption versus delay trade-off curve for gate sizing using a piecewise linear simulator," *IEEE Trans. Circuits Syst.*, vol. 15, no. 11, pp. 1424–1434, Nov. 1996.

[7] M. Berkelaar and E. Jacobs, "Using gate sizing to reduce glitch power," in *Proc. ProRISC Workshop Circuits, Syst., Signal Process.*, Mierlo, The Netherlands, Nov. 1996, pp. 183–188.

[8] M. Berkelaar and E. T. A. F. Jacobs, "Gate sizing using a statistical delay model," in *Proc. Des. Autom. Test Eur. Conf.*, Paris, France, Mar. 2000, pp. 283–290.

[9] M. Berkelaar and J. A. G. Jess, "Transistor sizing in MOS digital circuits with linear programming," in *Proc. Eur. Des. Autom. Conf.*, Mierlo, The Netherlands, Mar. 1990, pp. 217–221.

[10] M. T. Bohr, R. S. Chau, T. Ghani, and K. Mistry, "The high-k solution," *IEEE Spectrum*, vol. 44, no. 10, pp. 29–35, Oct. 2007.

[11] M. L. Bushnell and V. D. Agrawal*, Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Boston, MA: Springer, 2000.

[12] Cadence Design Systems, San Jose, CA, "Affirma analog environment reference," 2003. [Online]. Available: http://sourcelink.cadence.com,

[13] *Low-Power CMOS Design*, A. Chandrakasan and R. Brodersen, Eds. Piscataway, NJ: IEEE Press, 1998.

[14] A. P. Chandrakasan and R. W. Brodersen*, Low Power Digital CMOS Design*. Boston, MA: Springer, 1995.

[15] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[16] S. Datta, S. Nag, and K. Roy, "ASAP: A transistor sizing tool for area, delay and power optimization of CMOS circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1994, pp. 61–64.

[17] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *J. Appl. Phys.*, vol. 19, no. 1, pp. 55–63, Jan. 1948.

[18] J. P. Fishburn and A. E. Dunlop, "TILOS: A posynomial approach to transistor sizing," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 1985, pp. 326–328.

[19] F. Hu, "Process-variation-resistant dynamic power optimization for VLSI circuits," Ph.D. dissertation, Dept. ECE, Auburn Univ., Auburn, AL, 2006.

[20] F. Hu and V. D. Agrawal, "Input-specific dynamic power optimization for VLSI circuits," in *Proc. Int. Symp. Low Power Electron. Des. (ISLPED)*, Oct. 2006, pp. 232–237.

[21] C. J. Ihrig, G. J. Dhanabalan, and A. K. Jones, "A low-power CMOS thyristor based delay element with programmability extensions," in *Proc. 19th Great Lakes Symp. VLSI*, May 2009, pp. 297–302.

[22] S. Krishnamohan and N. R. Mahapatra, "An analysis of the robustness of CMOS delay elements," in *Proc. 15th IEEE Great Lakes Symp. VLSI*, Apr. 2005, pp. 412–415.

[23] Y. Lu, "Power and performance optimization of static CMOS circuits with process variation," Ph.D. dissertation, Dept. ECE, Auburn Univ., Auburn, AL., 2007.

[24] Y. Lu and V. D. Agrawal, "Leakage and dynamic glitch power minimization using integer linear programming for V_{th} assignment and path balancing," in *Proc. 15th Int. Workshop Power Timing Model., Opt. Simulation (PATMOS)*, Sep. 2005, pp. 217–226.

[25] Y. Lu and V. D. Agrawal, "CMOS leakage and glitch minimization for power-performance tradeoff," *J. Low Power Electron.*, vol. 2, no. 3, pp. 378–387, Dec. 2006.

[26] Y. Lu and V. D. Agrawal, "Total power minimization in glitch-free CMOS circuits considering process variation," in *Proc. 21st Int. Conf. VLSI Des.*, Jan. 2008, pp. 531–536.

[27] N. R. Mahapatra, S. V. Garimella, and A. Tareen, "An empirical and analytical comparison of delay elements and a new delay element design," in *Proc. IEEE Comput. Soc. Annu. Workshop VLSI*, Apr. 2000, pp. 81–86.

[28] N. R. Mahapatra, S. V. Garimella, and A. Tareen, "Efficient techniques based on gate triggering for designing static CMOS ICs with very low glitch power dissipation," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2000, vol. II, pp. 537–540.

[29] N. R. Mahapatra and R. Janakiraman, "Gate triggering: A new framework for minimizing glitch power dissipation in static CMOS ICs and its ILP-based optimization," in *Proc. 3rd IEEE Int. Caracas Conf. Devices, Circuits, Syst.*, Cancun, Mexico, Mar. 2000, pp. C 109:1–C 109:7.

[30] N. R. Mahapatra, A. Tareen, and S. V. Garimella, "Comparison and analysis of delay elements," in *Proc. 45th IEEE Midw. Symp. Circuits Syst.*, Aug. 2002, vol. 2, pp. 473–476.

[31] D. Nguyen, A. Davare, M. Orshansky, D. Chinney, B. Thompson, and K. Keutzer, "Minimization of dynamic and static power through joint assignment of threshold voltages and sizing optimization," in *Proc. Int. Symp. Low Power Electron. Des. (ISLPED)*, 2003, pp. 158–163.

[32] J. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*. Upper Saddle River, NJ: Prentice-Hall, 2003.

[33] J. M. Rabaey and M. Pedram*, Low Power Design Methodologies*. Boston, MA: Kluwer, 1996.

[34] T. Raja, "A reduced constraint set linear program for low-power design of digital circuits," M.S. thesis, Dept. ECE, Rutgers Univ., Piscataway, NJ, 2002.

[35] T. Raja, "Minimum dynamic power CMOS design with variable input delay logic," Ph.D. dissertation, Dept. ECE, Rutgers Univ., Piscataway, NJ, 2004.

[36] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Minimum dynamic power CMOS circuit design by a reduced constraint set linear program," in *Proc. 16th Int. Conf. VLSI Des.*, Jan. 2003, pp. 527–532.

[37] T. Raja, V. D. Agrawal, and M. L. Bushnell, "CMOS circuit design for minimum dynamic power and highest speed," in *Proc. 17th Int. Conf. VLSI Des.*, Jan. 2004, pp. 1035–1040.

[38] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Variable input delay CMOS logic design for low dynamic power circuits," in *Proc. 15th Int. Workshop Power Tim. Model., Opt. Simulation (PATMOS)*, Sep. 2005, pp. 436–445.

[39] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Variable input delay CMOS logic for low power design," in *Proc. 18th Int. Conf. VLSI Des.*, Jan. 2005, pp. 596–604.

[40] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Transistor sizing of logic gates to maximize input delay variability," *J. Low Power Electron.*, vol. 2, no. 1, pp. 121–128, Apr. 2006.

[41] K. Roy and S. C. Prasad*, Low-Power CMOS VLSI Circuit Design*. New York: Wiley, 2000.

[42] C. V. Schimpfle, A. Wroblewski, and J. A. Nassek, "Transistor sizing for switching activity reduction in digital circuits," in *Proc. Eur. Conf. Circuit Theory Des.*, Aug. 1999, vol. 1, pp. 114–117.

[43] V. Sundararajan, S. Sapatnekar, and K. Parhi, "Fast and exact transistor sizing based on iterative relaxation," *IEEE Trans. Comput.-Aided Des. Circuits Syst.*, vol. 21, no. 5, pp. 568–581, May 2002.

[44] Y.-F. Tsai, D. Durate, N. Vijaykrishna, and M. J. Irwin, "Implications of technology scaling on leakage reduction," in *Proc. Des. Autom. Conf.*, 2003, pp. 187–190.

[45] S. Uppalapati, "Low power design of standard cell digital VLSI circuits," M.S. thesis, Dept. ECE, Rutgers Univ., Piscataway, NJ, 2004.

[46] S. Uppalapati, M. L. Bushnell, and V. D. Agrawal, "Glitch-free design of low power ASICs using customized resistive feedtrough cells," in *Proc. 9th VLSI Des. Test Symp.*, Aug. 2005, pp. 41–48.

[47] N. H. E. Weste and D. Harris*, CMOS VLSI Design: A Circuits and Systems Perspective*, third ed. Boston, MA: Pearson Addison-Wesley, 2005.

[48] A. Wroblewski, C. V. Schimpfle, and J. A. Nassek, "Automated transistor sizing algorithm for minimizing spurious switching activities in CMOS circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2000, pp. 291–294.

[49] A. Wroblewski, O. Schumacher, C. V. Schimpfle, and J. A. Nassek, "Minimizing gate capacitances with transistor sizing," in *Proc. IEEE Int Symp. Circuits Syst.*, May 2001, pp. 186–189.

Tezaswi Raja (S'02–M'04) received the B.E. degree in electrical engineering from Andhra University, Visakhapatnam, India, in 2000 and the M.S. and Ph.D. degrees in computer engineering from Rutgers University, New Brunswick, NJ, in 2002 and 2004, respectively.

He is currently with nVIDIA Corporation, Santa Clara, CA, prior to which he worked at LSI Corporation, Milpitas, CA and at Transmeta Corporation, Santa Clara, CA. His research interests include low-power design, circuit testing, asynchronous circuit design, ultra low voltage design, process variations and emerging nanotechnology devices. He has published papers in the areas of low power VLSI design and distributed computing. He has two U.S. patents pending.

Dr. Raja has served on the Program Committees of International Conference on VLSI Design in 2006 and 2009. He has been a reviewer for *Journal of Electronic Testing: Theory and Applications* (JETTA), IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, *Design Automation Conference (DAC)*, *International Conference on Computer Aided Design* (ICCAD), and others.

Vishwani D. Agrawal (S'68–M'70–SM'80–F'86) received the B.E. degree from Indian Institute of Technology, Roorkee, India, in 1964, the M.E. degree from the Indian Institute of Science, Bangalore, India, in 1966, and the Ph.D. degree in electrical engineering from the University of Illinois, Urbana-Champaign, in 1971.

He is the James J. Danaher Professor of Electrical and Computer Engineering at Auburn University, Auburn, AL. His research interests include VLSI testing, low-power design and microwave antennas. He has published over 300 papers, has coauthored five books, and holds 13 U.S. patents. He coauthored textbook the textbook *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits* (Springer, 2000) with M. Bushnell.

Dr. Agrawal is recipient of eight Best Paper Awards, two Honorable Mention Paper Awards, the Life-Time Achievement Award of the VLSI Society of India, Harry H. Goode Memorial Award of the IEEE Computer Society, and the Distinguished Alumnus Award of the University of Illinois at Urbana-Champaign. He is the cofounder and Editor-in-Chief (since 1990) of the *Journal of Electronic Testing: Theory and Applications* and a past Editor-in-Chief (1985–1987) of the *IEEE Design and Test of Computers Magazine*. During 2003–2008, he served on the Editorial board of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He is a cofounder of the International Conference on VLSI Design and the VLSI Design and Test Symposium, both held annually in India. He is a Fellow of the ACM and IETE-India.

Michael L. Bushnell (M'74–SM'01–F'08) received the B.S. degree from the Massachusetts Institute of Technology, Cambridge, and the M.S. and Ph.D. degrees from Carnegie Mellon University, Pittsburgh, PA.

He is a Professor I and a Board of Trustees Research Fellow with the Electrical and Computer Engineering Department, Rutgers University, Piscataway, NJ. He has written 110 papers, coauthored 4 books, and holds 7 U.S. Patents. He has graduated 9 Ph.D. students and 39 M.S. students. His research interests include spectral analysis of digital and analog built-in self-testing, fault tolerance, wireless circuits, and nanotechnology.

Dr. Bushnell was a recipient of four Best Paper Awards. He is an Editor of the *Journal of Electronic Testing: Theory and Applications* and has served as the program chair for the International Conference on VLSI Design.