Performance Improvement in VLSI Circuit Design using Constant Delay Logic Style

Vallisree. S, G.Renganakayi
Department of Electronics and Communication Engineering
K.C.G College of Technology, Chennai, India
vallisrees@gmail.com

***Abstract* - A new logic style named Constant Delay (CD) logic has been introduced for improving the speed in VLSI circuits. The Constant Delay feature of this logic style makes it appropriate for implementing complicated logic expressions. In this logic style, the output is pre-evaluated before the input is fed to the Pull Down Network. This pre-evaluated characteristic is used for speed improvement over the existing static and dynamic domino logic styles. It has a timing window block for generating the required clock skew in this logic style. Data Output mode of operation is preferred to replace the critical path in the circuits. NAND gate has been implemented using Domino logic style and Constant Delay Logic style using 180nm CMOS technology in CADENCE. Performance Improvement was observed in Constant Delay Logic Style when compared to Domino Logic Implementation.**

Keywords—Logic style, Speed Improvement, VLSI Circuit Design

I. INTRODUCTION

Choosing an efficient logic style is considered to be one of the most important factors in the design of VLSI circuits. The logic style used in logic gates influences the basic parameters like speed, area and power dissipation of a circuit. The circuit delay is determined by the propagation delay of each gate, the number of transistors in series, transistor sizes. Circuit area depends on the number of transistors and their sizes. Power dissipation is determined by the switching activity and the node capacitances like gate, diffusion, and wire capacitances. All these characteristics may vary considerably from one logic style to another and thus make the proper choice of logic style is very much crucial for circuit performance.

The invention of the dynamic domino logic allowed designers to implement high performance circuit blocks like arithmetic logic units, at an operating frequency that cannot be achieved by traditional static and pass transistor CMOS logic styles. However, the performance improvement comes with several costs, including a reduced noise margin, a problem of charge-sharing and higher power dissipation due to a higher data activity. Compound domino logic (CDL), which uses dynamic and static gates alternating between each other has become the most popular logic style in high performance circuit blocks like 64 bit adder in modern CPUs. This implementation however comes at the expense of increased power consumption due to the possible direct path current during the pre-charge period. There is another logic named Source-coupled logic which has shown superior performances that are difficult to achieve by any other logic styles. However, it suffers from high power dissipation and it requires 2 complementary signals. Pseudo-nMOS logic, which uses a single pull-up pMOS transistor, provides both high speed and low transistor count at the expense of high static power consumption as well as reduced output voltage swing. While numerous high-speed logic styles have been proposed, dynamic and Compound domino logic still remain the most attractive choices when performance is the main objective. Let us discuss few of the most commonly used logic styles in VLSI Circuit Design.

A. Dynamic Logic Style

Dynamic logic is often used in CMOS circuits to reduce the transistor count, to increase speed and to avoid static power dissipation. The nMOS pull down network implements the logic function. Dynamic gates are clocked based on the sequence of two phases: Pre-charge and Evaluation phase.
PRECHARGE: When clock is low, the output node is pre-charged to V_{dd} by pMOS transistor. The nMOS transistor whose gate is connected to clock is cut-off and therefore no DC current flows regardless of the values of the DC signals

EVALUATION: When clock is high, pMOS transistor is off, while the nMOS transistor present below the PDN is turned ON. Depending on the value of inputs, a conditional path between Out and Ground is created, discharging the output node causing a low output signal. If such a path does not exist, the output node remains pre-charged causing a high output value.

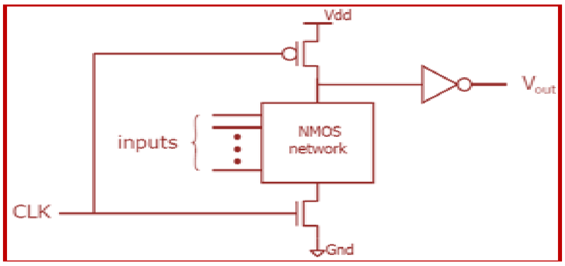


Figure 1: Dynamic Domino Logic

As opposed to static gates, dynamic gates are clocked and work in two phases, a pre-charge and an evaluation phase. The logic function is realized in a single nMOS pull down or pMOS pull up network, resulting in small input capacitances and fast evaluation times. This makes dynamic logic attractive for high-speed applications. However, the large clock loads and the high signal transition activities due to the pre-charging mechanism result in excessive high power dissipation. Also, the usage of dynamic gates is not as straightforward and universal as it is for static gates, and robustness is considerably degraded. With the exception of some very special circuit applications, dynamic logic is no viable candidate for low-power circuit design. Dynamic logic styles are often a good choice for high-speed, but not for low-power circuit implementations due to the high node activity and large clock loads

B. Compound Domino Logic:

According to the compound domino logic, plural nMOS input transistors are split into multiple sections, each section providing a separate preliminary output node. Thus, each cascaded section processes respective input logic signals to produce independent preliminary outputs. The preliminary output node of each section is connected to static logic gates which logically combine the signals of the preliminary outputs to produce multiple circuit outputs according to the logic function desired.

Figure 2 shows a compound domino logic implementation of a function. It is composed of a dynamic gate followed by a static gate such as 2-input NOR or NAND instead of an inverter. When the clock signal is low, all dynamic nodes are pre-charged to logic "1" and all static nodes fall to logic "0".

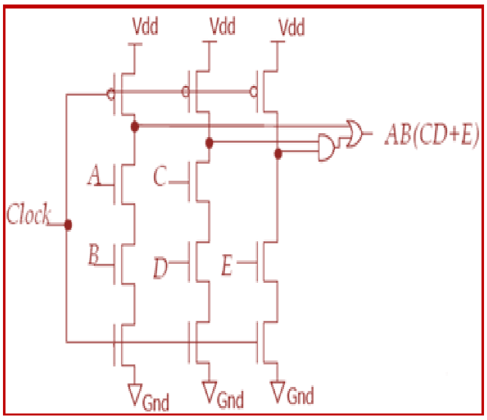


Figure 2: Compound Domino Logic

C. Pseudo Nmos Logic

The Pull-up Network in Conventional static CMOS is replaced by pMOS transistor with its gate connected to Ground; thereby the number of transistors is reduced. Pseudo-nMOS are ratioed circuits which rely on the correct pMOS to nMOS strength ratio to perform correct logic operations. pMOS transistor width is often selected to be about one-fourth the strength of the nMOS PDN as a compromise between noise margin and speed in pseudo-nMOS.

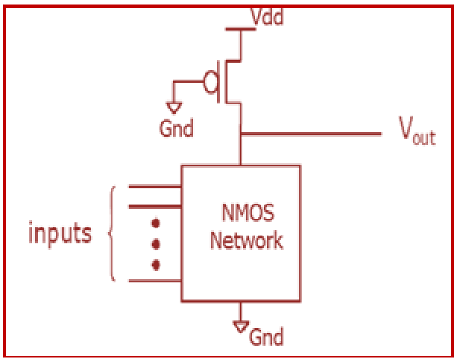


Figure 3: Pseudo nMOS logic

II. CONSTANT DELAY LOGIC STYLE

Figure 5: Buffer using CD logic

CONTENTION MODE: The contention mode happens when CLK is low while IN remains at logic "1." In this case, X is at a nonzero voltage level which causes 'Out' to experience a temporary glitch. The duration of this glitch is determined by the local window width, which is determined by the delay between CLK and CLK_d. When CLK_d becomes high, and if X remains low, then Y rises to logic "1," and turns off M1. Thus the contention period is over, and the temporary glitch at Out is eliminated.

C-Q DELAY MODE: C–Q delay mode takes places when IN makes a transition from high to low before CLK becomes low. When CLK becomes low, X rises to logic "1" and Y remains at logic "0" for the entire evaluation cycle. The delay is measured by the falling edge of both CLK and Out: hence the name C–Q delay.

D-Q DELAY MODE: D–Q delay mode utilizes the pre-evaluated characteristic of CD logic to enable high-performance operations. In this mode, CLK falls from high to low before IN transit, hence X initially rises to a nonzero voltage level. As soon as IN become logic "0," while Y is still low, then X quickly rises to logic "1." A race condition exists in this case between X and Y.

If CLK_d rises much earlier than X and Y will go to logic "1," turn off M1, and result in a false logic evaluation. If CLK_d rises slightly slower than X, then Y will initially rise (thus slightly turns off M1) but eventually settle back to logic "0."

CD logic can still perform the correct logic operation in this case, however, its performance is degraded because of M1‟s reduced current drivability. Therefore, it is important to maintain a sufficient window width under process-voltage-temperature (PVT) variations.

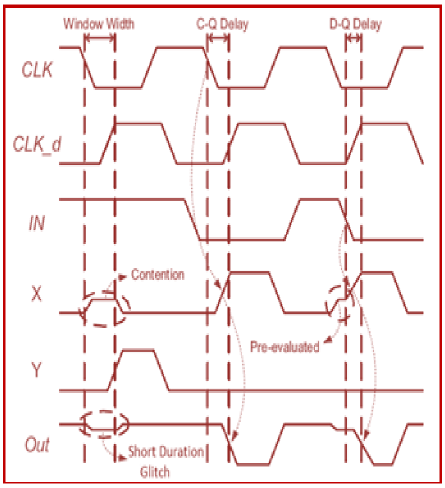


Figure 6: Timing Diagram for the CD logic

Compared to feed through logic, where the contention lasts for the entire evaluation period, Timing Block effectively reduces Constant Delay logic's power consumption during the contention mode. The local window technique in the proposed Constant Delay gate allows designers to customize the window width for different logic expressions to achieve minimal power dissipation while not sacrificing the performance. Another advantage of Constant Delay logic is that the internal node (X) is always connected to either VDD or GND, thus making the robustness of Constant Delay logic comparable to static logic, except during the contention mode.

B. Design Considerations

The sizing of INV1-3 and M3–M6 in Figure 5 is close to the minimum size so that they do not create a huge area burden. The length of INV1-3 can be altered to provide the required timing window duration based on designer's choices.

C. *Output Glitch*

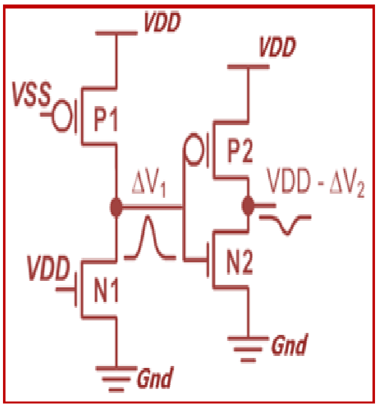


Figure 7: Contention Mode

Figure 7 depicts a simplified schematic of CD logic during the contention mode, where both transistors P1 and N1 are on simultaneously and induce a glitch voltage ΔV1, which in turn generates another smaller glitch ΔV2. By design, ΔV1 should be small [i.e., less than the threshold voltage (V_t)]. Hence, P1 operates in the saturation region while N1 is in the linear region. The current equation is given as

$$\frac{1}{2}\mu_p C_{ox}\frac{Wp1}{Lp1}(Vgs1 - Vtp)^2 = \mu_n C_{ox}\frac{Wn1}{Ln1}[(Vgsn1 - Vtn)Vds1 - \frac{V}{} \quad (1)$$

where μ_p and μ_n are the hole and electron mobility of pMOS and nMOS transistors, respectively, Cox is the oxide capacitance, W and L are the transistor width and length respectively, and Vgs and Vds are the transistor gate-to-source and drain-to-source voltages, respectively. Assuming same length devices are used, and $\mu_p \approx 0.5\ \mu_n$, rearranging (1) gives

$$\frac{W_{p1}}{4W_{n1}}(V_{DD} - V_{tp})^2 = (V_{DD} - V_{tn})\Delta V_1 - \frac{(\Delta V_1)^2}{2}. \quad (2)$$

ΔV1 can be found by solving the quadratic equation

$$\Delta V_1 = V_{DD} - V_{tn} - \sqrt{(V_{DD} - V_{tn})^2 - \frac{W_{p1}}{2W_{n1}}(V_{DD} - V_{tp})^2}. \quad (3)$$

Assuming Vtp ≈ Vtn, (3) can be approximated as

$$(4)\ \Delta V_1 = \frac{W_{P1}(V_{DD} - V_{tp})}{4W_{n1}}$$

For a given ΔV1, designers can quickly estimate the required Wp1 to Wn1 ratio. Moreover, ΔV1 is linearly proportional to the shift of V_t and transistor width in the presence of process variations.

D. Power Consumption

Data activity measures how frequent signals toggle and is defined as

$$\text{data activity} = \frac{\#\text{ of signal transitions}}{\#\text{ of signals} \times \#\text{ of clock cycles}}.$$

Static logic has an empirical α of 0.1–0.2 and dynamic domino logic has an activity factor of 0.5. While Constant Delay logic's α is also 0.5, it always consumes power when it enters the evaluation period. During the evaluation period, Constant Delay logic always dissipates power via either dynamic power dissipation (X goes to VDD and Out is discharged to GND) or direct path current (contention mode). While CD logic consumes more power, we believe that Constant Delay logic is still an attractive choice in a high-performance full-custom design because:

1) Constant Delay logic is only intended to replace the critical path

2) Power management techniques such as clock gating, where the clock connection to idle module is turned off (gated), will significantly reduce Constant Delay logic's dynamic power consumption.

III.CD LOGIC CHARACTERISTIC

All simulation runs in this paper are done in schematic level in the Cadence design environment using 180-nm CMOS technology. All the CD logic gates are designed such that the worst case glitch level is less than 300 mV at 110 °C. The window duration (width) is defined as the 50% point of the falling edge of CLK to the 50% point of the rising edge of node *Y*. The delay is measured at the 50% switching point of either the CLK or data to the 50% switching point of the latest output.

A. Noise Margin

Noise margin is defined as the dc noise level at the input generating a false logic evaluation at the output of the same gate and can be computed based on the following formula:

Noise Margin = |Voriginal – Vnoise|

where Voriginal is the expected voltage level without any input noise interference, and Vnoise is the input dc noise voltage that causes the false logic evaluation. For CD logic, two types of noise margin are defined: logic “1” and “0” noise margins. Logic “1” noise margin refers to the input dc noise level that causes the CD logic to fail to remain in the contention mode. If IN which is supposed to be at full V_{DD}, is now degraded due to noise, then the glitch level at X may be too high such that Out is falsely discharged. In this case, the noise margin can be calculated as 1V −Vin, where Vin is Vg of M7. Similarly, logic “0” noise margin refers to the input dc noise level that causes the CD logic to fail evaluating. In this case, if an input which is supposed to be at GND is now much higher due to noise, the contention between M1 and M7 will cause X to settle at an intermediate voltage instead of V_{DD}. When CLK_d rises to V_{DD} (window closes), Y will also be charged up through M3 and M4, since M3 is on and M4 is partially on because X is not at V_{DD}. If the voltage level at X is too low, then Y will be charged to V_{DD} through positive feedback and X will be discharged to GND through M7, which is driven by the noise source.

B. Performance

Constant Delay logic demonstrates superior performance, especially for complicated logic expressions, such as Y = AB + CD (AOI22), in the D-Q mode due to the pre-evaluated characteristic. Constant Delay logic is approximately two times faster than dynamic domino logic. This is contributed by:

1) Pre-evaluated characteristic;

2) Less number of transistors in the critical path (3N1P for dynamic, while only 2P1N for CD logic).

On the other hand, Constant Delay logic’s performance is only approximately the same as or even worse than that of dynamic domino logic during the C–Q mode.Therefore, it is advantageous to implement CD logic in a single-cycle multistage data path because then the pre-evaluated feature (D–Q delay) of CD logic can be fully utilized. This suggests that Constant Delay logic should be used only to replace the critical path in any circuit block, since it is not energy efficient to implement any system with CD logic only.

IV.Circuit Schematics And Simulation Result

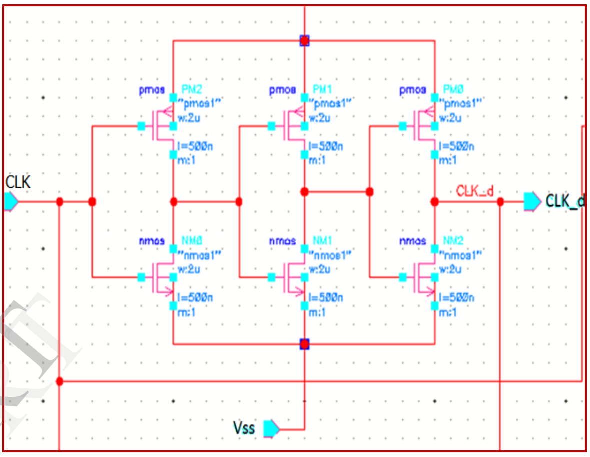
A. Simulation of CD Inverter

Constant Delay Logic design has been implemented using 180nm technology in CADENCE. Circuit design has been done in VIRTUOSO and simulated using SPECTRE.

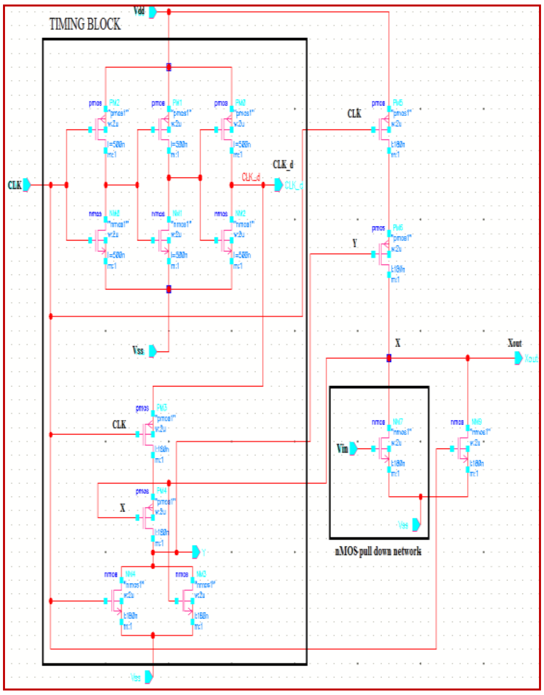
Table 1. Design Specifications

ACTUAL DESIGN SPECIFICATIONS	
Technology	180 nm technology
V_{DD}	1.8V
nMOS width Wn	2 um
pMOS width Wp	2.4 um
Clock Frequency	0.5 GHz
Data Frequency	0.5 GHz
Window Width	345ns (approx)

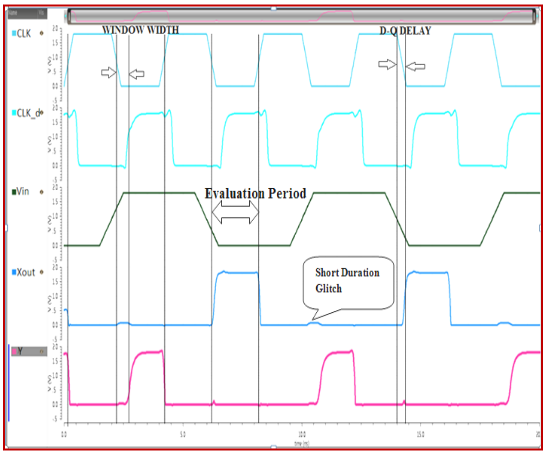
Timing Window Adjustment: Gate Length of Inverters 1-3 need to be altered to provide the appropriate delay between CLK and CLK_d signal .



1. Circuit Schematic for CD Inverter



2. Simulated Waveform for CD Inverte



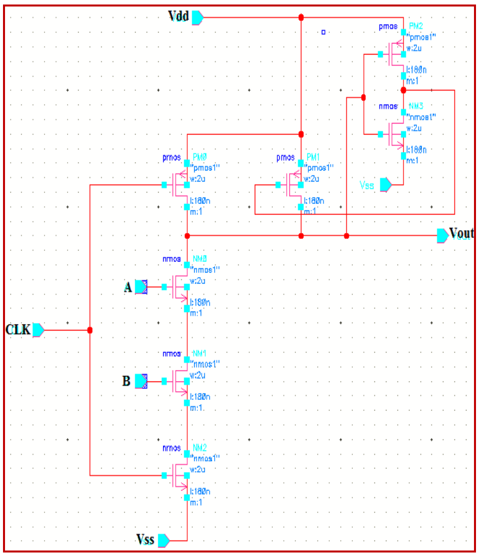
CLK High: Pre-discharge phase
CLK low: Evaluation phase

Glitch or Contention occurs only during a very short duration in this logic style instead of entire evaluation period in other dynamic logic styles. The duration of the glitch is decided by the Window width as shown in above figure.

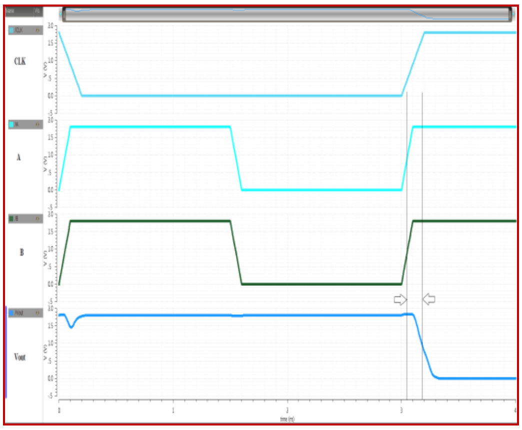
B. Delay Comparison of CD with Dynamic Logic

NAND gate has been implemented using both dynamic logic and Constant Delay logic and the performance has been compared in the following sections.

1.Circuit Schematic of NAND Gate Using Dynamic Logic



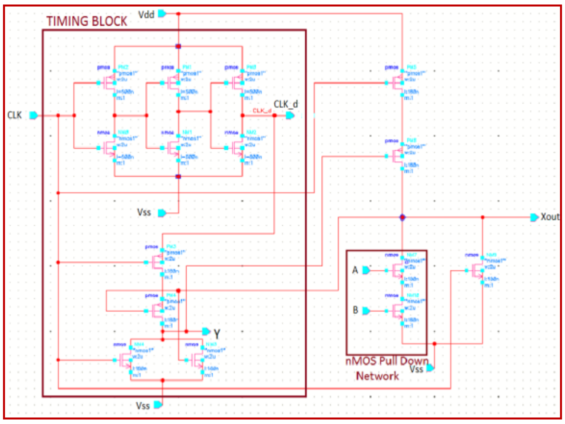
2.Simulation of NAND Gate Using Dynamic Logic



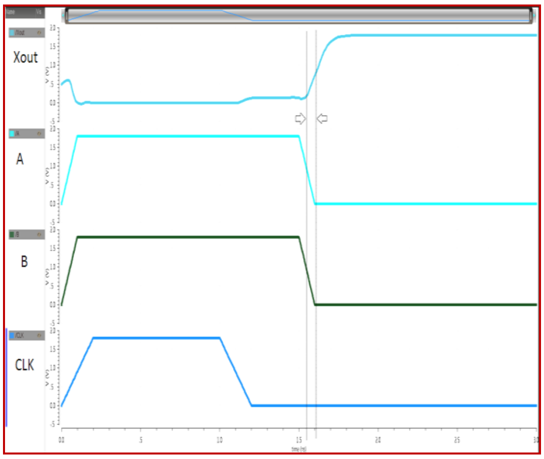
Propagation Delay:

Circuit schematic has been entered for Dynamic domino logic gate in CADENCE virtuoso and simulation with CLK, A and B input waveforms. Propagation delay is measured from 50% rise of input signals to 50% fall of Output signal slope as shown in the above result.
Measured Delay: 137ps

3.Circuit Schematic for CD NAND Gate



4.Simulation Result for CD NAND Gate



Measured Delay: 69ps

Table 2: Response Comparison

S.No	Comparison between Domino and CD Nand Gate	Response
1	Technology	180nm
2	Dynamic Nand Gate Delay	137ps
3	CD Nand Gate Delay	69ps

V.ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g." Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

1. S. Kiaei, S.-H. Chee, and D. Allstot, (May 1990), "CMOS source-coupled logic for mixed-mode VLSI," in Proc. IEEE Int. Circuits Syst. Symp., New Orleans, LA, pp. 1608–1611.

2. R. Zimmermann and W. Fichtner, (Jul. 1997), "Low-power logic styles: CMOS versus pass-transistor logic," IEEE J. Solid-State Circuits, vol. 32, no. 7, pp. 1079–1090.

3. N. Goncalves and H. De Man, (Jun. 1983), "NORA: A racefree dynamic CMOS technique for pipelined logic structures," IEEE J. Solid-State Circuits,vol. 18, no. 3, pp. 261–266.

4. C. Lee and E. Szeto, (May 1986), "Zipper CMOS," IEEE Circuits Syst. Mag., vol. 2, no. 3, pp. 10–16.

5. K. Bernstein, (Aug. 1998), High Speed CMOS Design Styles, 1st ed. New York:
Springer-Verlag.

6. L. McMurchie, S. Kio, G. Yee, T. Thorp, and C. Sechen, (2000), "Output prediction logic: A high-performance CMOS design technique," in Proc. Comput. Des. Int. Conf., Austin, TX, 2000, pp. 247–254.

7. V. Navarro-Botello, J. A. Montiel-Nelson, and S. Nooshabadi, (Aug. 2006), "Low power arithmetic circuit in feedthrough dynamic CMOS logic," in Proc. IEEE Int. 49th Midw. Symp. Circuits Syst., pp. 709–712.

8. V. Navarro-Botello, J. A. Montiel-Nelson, and S. Nooshabadi, (Jun. 2007), "Analysis of high-performance fast feedthrough logic families in CMOS," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 6, pp. 489–493.

9. P. Chuang, D. Li, and M. Sachdev, (May 2009), "Design of a 64-bit low-energy high performance adder using dynamic feedthrough logic," in Proc. IEEE Int. Circuits Syst. Symp., pp. 3038–3041.