

A Generalized Minimum Dynamic Power and High-Speed Design Method for CMOS Circuits

September 11, 2003

Abstract

We formulate a linear program (LP) to simultaneously minimize the dynamic power and overall delay of a CMOS circuit. To eliminate all glitches either without or with minimal number of delay buffers, a CMOS gate is assumed to have adjustable input to output delays for each input. Since these delays are not independent, a transistor sizing problem would require very complex non-linear optimization. We solve the problem in three steps. First, CMOS gates are analyzed to determine the realizable maximum differential input delay, u_b , for the device technology being used. Second, an LP assumes the gate input and output delays as independent variables and determines them for all gates. This LP satisfies (1) glitch elimination conditions and the realizability constraint (u_b) for all gates, and (2) the specified overall delay for the circuit. The total number of constraints in our LP is a linear function of the circuit size. Third, all gates are designed with the delays determined by the LP. As a sample result, using $u_b = 10$ when we designed the c1355 benchmark circuit specifying a large overall delay, a zero buffer design was obtained. It consumed 33% power and had three times the overall delay as compared to an unoptimized design. When the overall delay was constrained not to increase, the low-power design required 64 delay buffers and consumed 37% power.

1. Introduction

The power dissipated in a CMOS circuit can be divided into *dynamic power*, *leakage power* and *short-circuit power* components. The topic of this paper is the reduction of *dynamic power*. When an input vector is applied to the primary inputs (PI), the minimum power requirement for each gate output is to produce either 0 or 1 transition. However, in reality there may be many more transitions due to *glitches* or *hazards*, which are caused by the *differential delays* of paths leading to the inputs of the gates.

Dynamic power of a circuit is reduced by eliminating some or all glitches. The principal idea of a glitch reduction technique is to find delay assignments for all gates in the circuit so as to reduce the differential path delays at gate inputs with respect to the inertial delays. Published techniques are *balanced delay method* [7, 11, 17, 18, 22], *hazard filtering method* [1, 26], *transistor sizing* [5, 6, 10, 12, 23, 24, 27], *gate sizing* [3, 4, 25], and *linear programming (LP) Techniques* [2, 19–21]. We will not elaborate on these techniques due to space limitation of this paper and the reader is directed to the cited references.

This paper falls under the category of LP techniques with the goal of eliminating all glitches. Earlier techniques [2, 19] provide a control over the speed of the circuit with buffers inserted. The next technique [21] provides minimum power possible but the speed is compromised. This paper provides a general solution with control over speed and minimum power. The contributions of this paper are 1) the control over the speed of the circuit while designing the circuit with different input-output gate delays 2) formulating an LP that incorporates this information 3) a consequence of the first two, which is, achieving a general solution to the problem of controlling the delay and power of the circuit, giving the designer flexibility in both the dimensions.

We outline the relevant prior work on LP techniques in Section 2. In Section 3 we propose the new generalized LP formulation. The results are tabulated in Section 4. The conclusion lists issues to be addressed by future research.

2. Prior Work

Among the linear programming techniques, three are most relevant to the present work. First is the basic path enumeration framework. Second is an alternative, but equivalent, formulation that reduces the constraint set size to be linear. The third is the expanded gate input to output delay design.

2.1. Path Enumeration Method

Agrawal *et al.* [2] show that for a correct operation with *minimum transient energy* (MTE) consumption, every CMOS gate in the circuit must produce no more than one event (signal change) at its output during a transition interval. The *transition interval* is defined as the interval after the primary inputs (PIs) change and during which all signals attain their steady state. They prove that if the new logic output is different from the old value then only a single transition can achieve the correct result. Assuming a single delay variable per gate, Agrawal *et al.* [2] eliminate all glitches by making the gate delay exceed the differential path delay at the gate inputs. They find that,

1. If the overall delay of the circuit is allowed to increase then an MTE design is always possible by adjusting the output delays of the gates. This MTE design does not require the addition of any delay buffers and hence is the lowest dynamic power design.
2. If the overall delay is bounded then the MTE design is not guaranteed without the insertion of delay buffers.

They describe an LP model to generate constraints for hazard filtering, keeping the overall delay within the specified limits.

Consider a gate with two inputs 1 and 2. The *minimum transient energy* (MTE) condition for this gate ensures that the delay difference between path P1 and path P2, arriving at inputs 1 and 2, respectively, is not greater than the inertial delay (d) of the gate as shown in Eq. 1.

$$\left| \sum_{P1\ path} gate\ delays - \sum_{P2\ path} gate\ delays \right| \leq d \quad (1)$$

$$\sum_{PI \rightarrow PO\ path} gate\ delays \leq maxdelay \quad (2)$$

Such a condition must be satisfied for all pairs of paths terminating at the inputs of all gates. Overall delay is also constrained for each path by constraints such as Eq. 2 where *maxdelay* is a given design parameter. Since the number of paths terminating at gates increases exponentially, the number of constraints also increase exponentially with the size of the circuit. This high complexity prevents the model from optimizing large circuits. For example the circuit c880 needs 6.9 million path constraints, which cannot be tackled by many linear programming tools.

2.2. Linear Constraint Set Method

Raja *et al.* [19,20] have described a way of reducing the complexity of the constraint set from exponential to linear in circuit size. In addition to the inertial delay, they introduce two new variables per gate in the LP, *viz.*, earliest time of arrival of the signal at a gate and the latest signal arrival time. This approach is similar to the timing verification algorithm described by Hitchcock [14,15]. These two variables define the *timing window* in which the signal can change at the output of the gate. The LP constraint set then forces the inertial delay to be greater than the timing window at the output of the gate. The authors prove that their formulation is equivalent to the path enumeration model though it reduces the constraint set to be linear in the size of the circuit. For example, the circuit c880, which earlier needed 6.9 million constraints, requires only 3,611 constraints by this method.

Chuang *et al.* use a similar set of delay variables to simultaneously optimize the area and timing of a standard-cell design [8,9].

2.3. Differential Delay Upper Bound Method

Raja *et al.* [21] have described a technique of designing gates with different input-output delay along different IO paths through the gate. Thus, their gate consists of an inertial delay for the output and a set of delays for the inputs.

Consider the transistor schematic of a NAND gate with inputs 1 and 2 and output 3. The gate has two IO paths (1,3) and (2,3). The IO path delay from gate 1 to gate 3 can be varied by changing the transistors connected to input 1. This will result in changing the input capacitance associated with input 1 without much affecting the path delay through input 2. Still these delays are not independent of each other.

A gate can be designed with different IO delays along different IO paths without adding new components. However, there is an upper bound to the amount of delay difference that can be created within two IO paths of the gate using this method. This limit is dictated by the technology in which the gate is designed and they specify it as a *feasibility parameter*, u_b .

Definition: *Gate input differential delay upper bound u_b :* The gate input delay upper bound is a measure of the maximum difference in delay of any two IO paths through a gate, that can be designed in a particular technology at the transistor and layout levels.

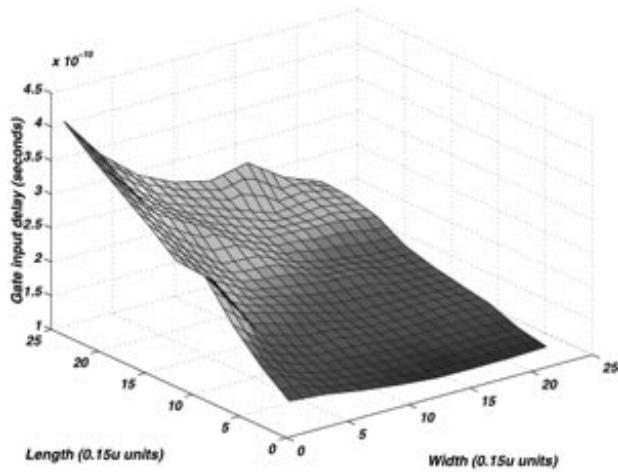


Figure 1: Delay plot of a NAND gate, by varying the transistor pair of input 1 and keeping the transistor pair in input 2 constant.

They designed a NAND gate in $0.25\mu\text{m}$ CMOS technology using the Cadence system. The width and length of a pair of transistors in the gate was varied and the difference in delay for the two IO paths through the gate was determined using the circuit-level simulator, Spectre. The result is given in Figure 1. This shows a differential delay of up to 400 ps. In this technology the fastest gate has a delay of about 50 ps. Hence, for this technology the feasibility parameter for a NAND gate is $u_b = 8$.

Realizing that the IO delays through a gate are *not independent*, Raja *et al.* [21] use a three step approach. In the first step, CMOS gates are analyzed at the transistor level to determine a suitable value of u_b for the technology used in the design. Then, the IO delays of a gate are assumed to be independent variables of the LP in the step 2. In step 3, using the delay assignment provided by the LP, gates are redesigned at the transistor level to meet the delay specification.

The use of variable input delays reduces the number of delay buffers needed to satisfy the glitch elimination conditions. Thus, the power otherwise consumed by the buffers is saved. However, a full-circuit transistor sizing solution will be quite complex due to the inter-dependences between the input delays, and the three-step solution is significantly efficient.

2.4. Shortcomings of Previous Methods

Consider the hypothetical power-delay graph shown in Figure 2. Techniques of Subsections 2.1 and 2.2 correspond to $u_b = 0$. The minimum power design (dot on the $u_b = 0$ curve) is obtained when the

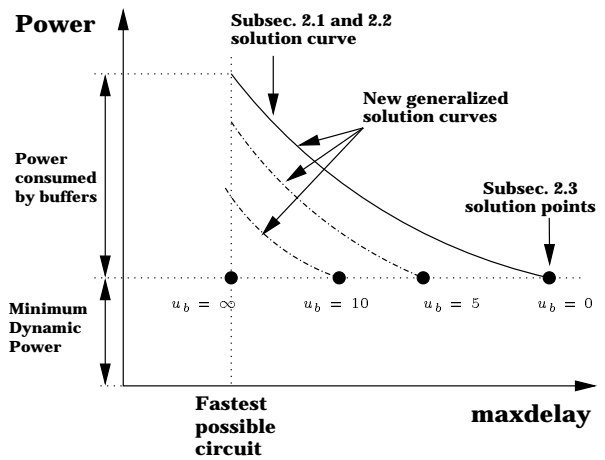


Figure 2: Power-delay curves and the solutions obtained by various methods.

overall circuit delay (*maxdelay*) is not constrained. Here, glitch elimination is accomplished at the cost of increased circuit delay and without inserting any delay buffers. As *maxdelay* is constrained, design inserts buffers on non-critical paths and the power increases due to the consumption in those buffers. These solutions lie on the solid curve in Figure 2.

The technique of Subsection 2.3 corresponds to $u_b > 0$. Differential input delays accomplish the same function that is served by the delay buffers in the $u_b = 0$ design. In the previously published work [21], power minimization was the only goal, which was obtained at the cost of the increased delay (dots on $u_b = 0, 5$ and 10 curves).

In this paper, we generalize the previous techniques. Thus, depending on the design parameters (u_b and *maxdelay*), the best power-delay solution is obtained along the curve that corresponds to the given u_b . When the given *maxdelay* is large, a no-buffer solution is obtained (one of the dots in Figure 2). For small *maxdelay* buffers may be inserted on non-critical paths to eliminate glitches.

3. A New Generalized Formulation

Consider the simple combinational circuit shown in Figure 3. Traditionally, for the purpose of gate sizing the circuit is viewed with each gate having a single *inertial delay* and all input-output (IO) paths through the gate are assumed to have the same delay. We redefine the gate delay. A gate can be viewed as having one *basic inertial delay* and another transport delay component for every IO path running through the gate. This is illustrated in Figure 4.

The delay buffers are inserted at each PI and at

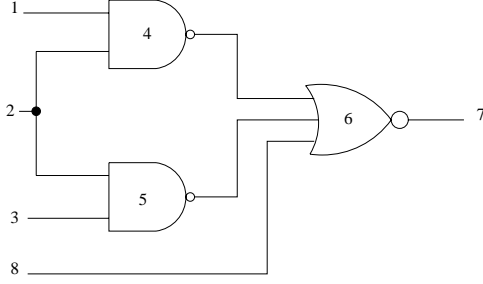


Figure 3: A combinational circuit.

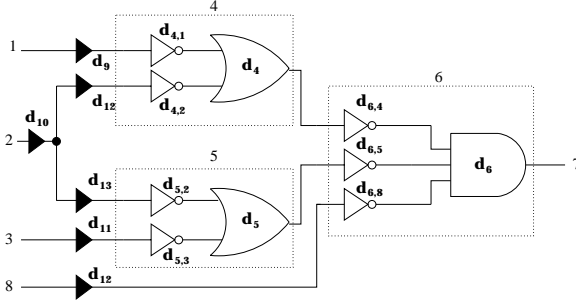


Figure 4: Delay model for the circuit of Figure 3.

each fanout stem as shown in Figure 4. We emphasize that these delay elements (inserted at primary inputs and fanout branches) are only for analysis purposes and are not physical components in the circuit. The buffers represent additional delay variables in the linear program (LP). The LP assigns a non-zero value to a buffer only when *necessary* to satisfy the glitch elimination or *maxdelay* conditions, in which case the buffer becomes a physical element.

3.1. Linear Program

Consider the delay model of Figure 4. Now the linear program can be written as follows.

3.1.1 Variables

- Inertial delays of gates: d_4, d_5, d_6 .
- Inertial delays of buffers: $d_9 \dots d_{13}$ with a minimum value of 0.
- Gate input delays: $d_{i,j}$ is the extra delay on the path from the fanin gate j to gate i . For instance $d_{4,1}$ is the extra delay of the path through gate 4 while arriving from gate 1. This models the difference in delay of various IO paths through the gate. Its minimum value is 0.
- T_i is the latest time of signal change at the output of gate or buffer i .

- t_i is the earliest time of signal change at the output of gate or buffer i .
- $T_{i,j}$ is the latest time of signal change at the output of delay element whose delay is $d_{i,j}$.
- $t_{i,j}$ is the earliest time of signal change at the output of delay element whose delay is $d_{i,j}$.

3.1.2 Constraints on Delays

These are the lower and upper bounds on variables:

- The lower bound on gate inertial delays are set to 1 and the lower bound on buffer inertial delays are set to 0.
- The lower bound on the gate input delays are set to 0 and upper bound is set to u_b known as *feasibility constraints* (see Subsection 2.3). For example, the feasibility constraints for gate 6 would become

$$d_{6,4} \leq u_b ; d_{6,5} \leq u_b ; d_{6,8} \leq u_b$$

where u_b is the maximum differential delay of that technology.

3.1.3 Glitch Suppression Constraints

These constraints ensure that the timing window for signal transitions at every gate output does not exceed the inertial delay [19, 20].

Consider gate 6 in Figure 4. The constraints for it are given as:

$$\begin{aligned} t_6 &\leq t_{6,4} + d_6 ; T_6 \geq T_{6,4} + d_6 \\ t_6 &\leq t_{6,5} + d_6 ; T_6 \geq T_{6,5} + d_6 \\ t_6 &\leq t_{6,8} + d_6 ; T_6 \geq T_{6,8} + d_6 \\ d_6 &\geq T_6 - t_6 \end{aligned}$$

and the constraints for an IO delay element $d_{6,4}$ are

$$t_{6,4} \leq t_4 + d_{6,4} ; T_{6,4} \geq T_4 + d_{6,4}$$

Consider buffer 12 in Figure 4. The constraints for it are given as:

$$t_{12} \leq t_{10} + d_{12} ; T_{12} \geq T_{10} + d_{12}$$

3.1.4 Maxdelay Constraints

For every PO, we have

$$T_7 \leq \text{maxdelay};$$

Here *maxdelay* is a parameter that is specified by the designer. It is the sum of delays along the longest path of the circuit. This parameter determines how fast the circuit is required to function.

3.1.5 Generalized Objective Function

To achieve both highest speed and lowest power we use an objective function:

$$\text{Minimize } a \times \sum_{j \in \text{buffers}} d_j + b \times \text{maxdelay} \quad (3)$$

where a and b are the relative weights for power and delay, respectively. Two cases of interest are:

- Case 1: maxdelay is a fixed parameter, $a = 1$ and $b = 0$. The buffer delay is minimized to meet the overall circuit delay requirement. This type of solution is illustrated in the next section.
- Case 2: maxdelay is optimized, $a = 0$ and $b = 1$. Buffer delays are fixed to zero. Thus, a bufferless design is obtained. For the least power dissipation (prime objective), this design has the highest speed [21].

4. Results

A linear program was written for the ISCAS'85 benchmark circuits and solved using AMPL [13]. The resulting delay assignments are used in the delay simulator for the power estimation analysis as described by Hsiao *et al.* [16]. Compacted fault coverage vectors from an ATPG program were used for simulation and power estimation. The results are shown in Table 1.

For comparison, an unoptimized version of each circuit was also analyzed for power dissipation. All gates in this circuit were assumed to have one unit of delay, which is the smallest realizable delay for a gate. All delays, including the values of u_b and maxdelay , are expressed in this delay unit. The power dissipation of the unoptimized circuit was normalized to 1.0.

We make two observations. The power of a circuit can be decreased by increasing u_b , the input delay flexibility allowed by the technology. Power increases as the circuit is made faster. Although the circuit still has no glitches, it contains buffers on non-critical paths, causing extra power dissipation.

Each circuit was designed for the minimum, twice the minimum, and thrice the minimum delay. For example, c432 has 15 gates on the longest path and the three designs have delays of 15, 30 and 45 units. Correspondingly, the unoptimized design has a delay of 15 units. The generalized design method allows a better power-delay trade-off.

5. Conclusion

A given CMOS circuit is optimized for minimum dynamic power when its operation contains no

glitches. Earlier techniques of glitch reduction do it either by inserting buffers or by increasing the delay both of which are undesirable. The generalized solution takes advantage of both methods and the design can be made faster and yet consume less power. On an average we found an extra 20% reduction in the dynamic power without any speed reduction.

References

- [1] V. D. Agrawal, "Low Power Design by Hazard Filtering," in *Proc. of the International Conference on VLSI Design*, 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 Linear Programming Method," in *Proc. of the International Conference on VLSI Design*, Jan. 1999, pp. 434–439.
- [3] 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 Transactions on Circuits and Systems*, vol. 15, no. 11, pp. 1424–1434, Nov. 1996.
- [4] M. Berkelaar and E. Jacobs, "Using Gate Sizing to Reduce Glitch Power," in *Proc. of the ProRISC Workshop on Circuits, Systems and Signal Processing*, (Mierlo, The Netherlands), Nov. 1996, pp. 183–188.
- [5] M. Berkelaar and J. A. G. Jess, "Transistor Sizing in MOS Digital Circuits with Linear Programming," in *Proc. of the European Design Automation Conference*, (Mierlo, The Netherlands), Mar. 1990, pp. 217–221.
- [6] M. Borah, M. J. Irwin, and R. M. Owens, "Minimizing Power Consumption of Static CMOS Circuits by Transistor Sizing and Input Reordering," in *Proc. of the International Conference on VLSI Design*, Jan. 1995, pp. 294–298.
- [7] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*. Boston: Kluwer Academic Publishers, 1995.
- [8] W. Chuang, S. S. Sapatnekar, and I. N. Hajj, "A Unified Algorithm for Gate Sizing and Clock Skew Optimization to Minimize Sequential Circuit Area," in *Proc. of the International Conference on Computer-Aided Design*, Nov. 1993, pp. 220–223.
- [9] W. Chuang, S. S. Sapatnekar, and I. N. Hajj, "Timing and Area Optimization for Standard Cell VLSI Circuit Design," *IEEE Transactions on Computer-Aided Design*, vol. 14, no. 3, pp. 308–320, Mar. 1995.
- [10] S. Datta, S. Nag, and K. Roy, "ASAP: A Transistor Sizing Tool for Area, Delay and Power Optimization of CMOS Circuits," in *Proc. of the IEEE International Symposium on Circuits and Systems*, May 1994, pp. 61–64.

Table 1: Dynamic power dissipation of ISCAS'85 benchmark circuits obtained from generalized low-power design method.

Circuit	Unopt. circuit Power	Vectors	Optimized circuits						$maxdelay$ (delay units)
			$u_b = 0$		$u_b = 5$		$u_b = 10$		
			Buffers	Power	Buffers	Power	Buffers	Power	
c432	1.0	56	73	0.62	63	0.57	63	0.56	15
		56	39	0.61	9	0.52	0	0.47	30
		56	36	0.60	0	0.47	0	0.47	45
c499	1.0	54	80	0.92	32	0.88	0	0.75	11
		54	48	0.91	0	0.73	0	0.73	22
		54	0	0.69	0	0.73	0	0.73	33
c880	1.0	78	62	0.68	0	0.48	0	0.45	24
		78	34	0.68	0	0.45	0	0.42	48
		78	29	0.62	0	0.44	0	0.42	72
c1355	1.0	87	224	0.42	64	0.39	64	0.37	24
		87	192	0.42	64	0.37	32	0.35	48
		87	160	0.40	32	0.36	0	0.33	72

- [11] M. S. Elrabaa, I. S. Abu-Khater, and M. I. Elmasry, *Advanced Low-Power Digital Circuit Techniques*. Boston: Kluwer Academic Publishers, 1997.
- [12] J. P. Fishburn and A. E. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor Sizing," in *Proc. IEEE International Conf. Computer-Aided Design*, Nov. 1985, pp. 326–328.
- [13] R. Fourer, D. M. Gay, and B. M. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. South San Francisco, California: The Scientific Press, 1993.
- [14] R. B. Hitchcock Sr., "Timing Verification and the Timing Analysis Program," in *Proc. of the 19th Design Automation Conf.*, June 1982, pp. 594–604.
- [15] R. B. Hitchcock Sr., G. L. Smith, and D. C. Cheng, "Timing Analysis of Computer Hardware," *IBM Journal of Research & Development*, vol. 26, no. 1, pp. 100–105, Jan. 1982.
- [16] M. Hsiao, E. M. Rudnick, and J. H. Patel, "Effects of Delay Model in Peak Power Estimation of VLSI Circuits," in *Proc. of the International Conference on Computer-Aided Design*, Nov. 1997, pp. 45–51.
- [17] J. Monteiro and S. Devadas, *Computer-Aided Design Techniques for Low Power Sequential Logic Circuits*. Boston: Kluwer Academic Publishers, 1997.
- [18] J. M. Rabaey and M. Pedram, *Low Power Design Methodologies*. Boston: Kluwer Academic Publishers, 1995.
- [19] T. Raja, "A Reduced Constraint Set Linear Program for Low Power Design of Digital Circuits," Master's thesis, Rutgers University, Dept. of ECE, Piscataway, New Jersey, May 2002.
- [20] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Minimum Dynamic Power CMOS Circuit Design by a Reduced Constraint Set Linear Program," in *Proc. of the International Conference on VLSI Design*, Jan. 2003, pp. 527–532.
- [21] T. Raja, V. D. Agrawal, and M. L. Bushnell, "CMOS Circuit design for Minimum Dynamic Power and Highest Speed," in *Proc. of the International Conference on VLSI Design*, Jan. 2004.
- [22] K. Roy and S. C. Prasad, *Low-Power CMOS VLSI Circuit Design*. New York: Wiley Interscience Publication, 2000.
- [23] C. V. Schimpfle, A. Wroblewski, and J. A. Nassek, "Transistor Sizing for Switching Activity Reduction in Digital Circuits," in *Proc. of the European Conference on Theory and Design*, Aug. 1999.
- [24] J. M. Shyu, A. L. Sangiovanni-Vincentelli, J. P. Fishburn, and A. E. Dunlop, "Optimization-based Transistor Sizing," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 400–409, Apr. 1988.
- [25] V. Sundararajan, S. Sapatnekar, and K. Parhi, "Fast and Exact Transistor Sizing Based on Iterative Relaxation," *IEEE Transactions on Computer Aided Design of Circuits and Systems*, vol. 21, 2002.
- [26] S. H. Unger, *Asynchronous Sequential Switching Circuits*. New York: Wiley-Interscience, 1969.
- [27] A. Wroblewski, C. V. Schimpfle, and J. A. Nassek, "Automated Transistor Sizing Algorithm for Minimizing Spurious Switching Activities in CMOS Circuits," in *Proc. of the IEEE International Symposium on Circuits and Systems*, May 2000.