Minimum Dynamic Power CMOS Circuit Design by a Reduced Constraint Set Linear Program

Tezaswi Raja
Rutgers University, Dept. of ECE
Piscataway, NJ 08854, USA
tezaswir@caip.rutgers.edu

Vishwani D. Agrawal*
Agere Systems
Murray Hill, NJ 07974, USA
vishwani02@yahoo.com

Michael L. Bushnell
Rutgers University, Dept. of ECE
Piscataway, NJ 08854, USA
bushnell@caip.rutgers.edu

Abstract

In the previous work, the problem of finding gate delays to eliminate glitches has been solved by linear programs (LP) requiring an exponentially large number of constraints. By introducing two additional variables per gate, namely, the fastest and the slowest arrival times, besides the gate delay, we reduce the number of the LP constraints to be linear in circuit size. For example, the 469-gate c880 circuit requires 3,611 constraints as compared to the 6.95 million constraints needed with the previous method. The reduced constraints provably produce the same exact LP solution as obtained by the exponential set of constraints. For the first time, we are able to optimize all ISCAS'85 benchmarks. For the c7552 circuit, when the input to output delay is constrained not to increase, a design with 366 delay buffers consumes only 34% peak and 38% average power as compared to an unoptimized design. As shown in previous work, the use of delay buffers is essential in this case. The practicality of the design is demonstrated by implementing an optimized 4-bit ALU circuit for which the power consumption was obtained by a circuit-level simulator.

1. Introduction

The topic of this paper is the reduction of *dynamic power* in CMOS circuits. When an input vector is applied to the primary inputs (PI), the requirement for each gate is to produce one or no output transition. However, in reality they produce many transitions. These extra transitions are caused by the differential delays of paths leading to the inputs of the gates. This subject is widely discussed in recent books [7, 9, 16, 18, 20].

Among various methods for minimizing the dynamic power is the *balanced delay* method in which we equalize the delays of all paths incident on a gate. When a signal fans out, its delay affects several paths and balancing may require insertion of delay buffers on selected fanout branches. While buffers consume power they allow the balancing without increase in overall delay of the circuit. An alternative to the balanced delay method is the *hazard filtering* [1]. If a pulse of width lesser than the inertial delay of the gate is incident on a gate input then that would be suppressed or filtered by the gate and this is known as the *filtering effect* of a gate [26]. Thus, by adjusting the inertial delay to be greater than the differential path delay of arriving inputs at the gate, the glitches can be eliminated. Clearly, the overall delay constraint will increase.

A combination of both delay balancing and hazard filtering has been tried by Agrawal *et al.* [2]. They describe a linear programming model to generate constraints for hazard filtering while keeping the overall delay within 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:

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

Such a condition must be satisfied for all pairs of paths terminating at the inputs of all gates. Thus, if the sets $P1$ and $P2$ have $k1$ and $k2$ elements, respectively, then there are at least $k1 \times k2$ constraints for that gate. As the level of gate increases $k1$ and $k2$ increase and hence the number of constraints for the gate increases exponentially with the size of the circuit. Additional constraints are used to hold the

*Visiting Professor, Dept. of ECE, Rutgers University, Piscataway, NJ

Proceedings of the 16th International Conference on VLSI Design (VLSI'03)
1063-9667/03 $17.00 © 2003 IEEE

overall circuit delay within limits,

$$\sum_{PI \rightarrow PO \ path} gate\ delays \leq maxdelay$$

where *maxdelay* is a design parameter. Once again the number of these constraints increases exponentially. This high complexity limits the model from optimizing large circuits.

Another technique is known as *transistor sizing* [5, 6, 8, 10, 21, 24, 27]. By sizing, we mean the width and length of the transistor that change the driving capacity and thus the delay of the transistor. This method does not add any buffers but solves the problem in a large dimensional space by treating all transistors as parameters and numerical convergence is often a problem [23]. Another method is called *gate sizing* where each logic gate is modeled as an equivalent inverter [3, 4]. The gate sizes vary in a continuous manner. The main problems are the non-linearity of the model and the discrete gate sizes which cause mathematical difficulties especially for large circuits.

In this paper, we describe a new technique in which the constraint set size of the LP has a linear complexity [19].

2. The New Approach

In addition to the single delay parameter as was done earlier [1, 2, 4], we introduce two new variables for every gate, one for earliest time and the other for the most delayed time of arrival of signal at the output of the gate. The difference of these variables is a *timing window* within which the various signals arrive at the gate. Consider a gate i with n inputs. We define a variable T_i as the maximum time instant at which an event can occur at the output of the gate after the occurrence of an event at the PIs of the circuit. Similarly, t_i is the minimum time instant at which an event can occur at the output of the gate. This means that events always occur in the interval $[t_i,T_i]$ at the output of the gate i. This technique of minimum and maximum arrival time variables is similar to the non-enumerative static timing analysis [12].

Theorem 1: Consider a gate i with n inputs, receiving events from fanin gates 1, 2, ..., n at times m_1, m_2, ... m_n. Assuming that $T_1 \leq T_2 \leq ... \leq T_n$ and $t_1 \leq t_2 \leq ... \leq t_n$ are the arrival time parameters for the fanin gates, the number of events at the output of gate i cannot exceed

$$\min\{n, 1 + \lfloor \frac{T_n - t_1}{d_i} \rfloor\} \qquad (1)$$

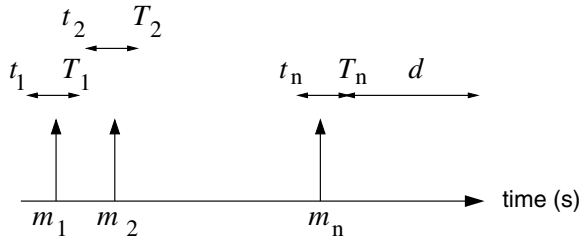


Figure 1: An arbitrary distribution of events.

where d_i is the inertial delay of gate i [2].

Proof: We consider two cases for the gate i with n inputs.

- *First upper bound.* The maximum number of events cannot be greater than the maximum number of possible events at the input, which is n in this case (the number of fanins).
- *Second upper bound.* At the inputs of gate i, n events can be arbitrarily placed in time. Without loss of generality, we order them as 1, 2, ... n in Figure 1. From the definition of *ideal delay* [2] an output event can occur only if the separation between successive events is greater than d_i. The largest window in which the events can occur is $[t_1,T_n + d_i]$, and the number of events is given by

$$\lfloor \frac{T_n - t_1 + d_i}{d_i} \rfloor = 1 + \lfloor \frac{T_n - t_1}{d_i} \rfloor \qquad (2)$$

Combining both upper bounds we get Equation 1. ■

From Theorem 1, the number of events takes the *least possible value* (the condition for minimum dynamic power) when

$$T_n - t_1 \leq d_i$$

According to Equation 1, the number of events at the output of the gate will then not exceed 1. Also, since

$$\begin{aligned} T_i &= T_n + d_i \\ t_i &= t_1 + d_i \end{aligned}$$

Hence, the condition for MTE can be written as:

$$d_i \geq T_i - t_i \qquad (3)$$

3. Linear Programming

A linear program determines a set of variables such that an objective function is minimized under the given constraints. We illustrate the linear programming model with the example of the adder circuit shown in Figure 2.

Proceedings of the 16th International Conference on VLSI Design (VLSI'03)
1063-9667/03 $17.00 © 2003 IEEE

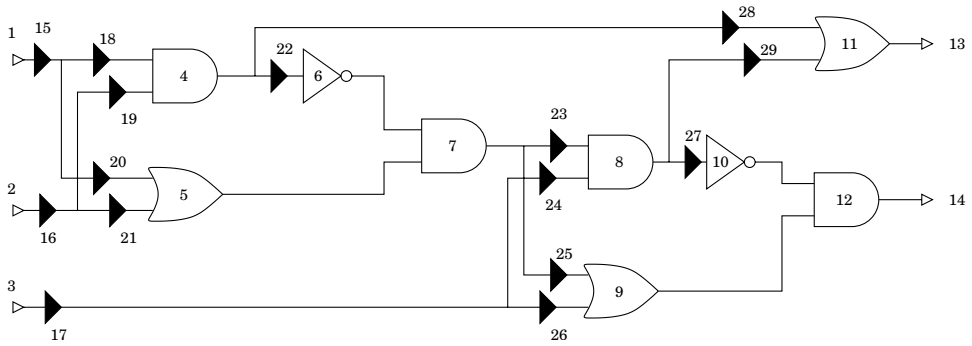


Figure 2: A 1-bit adder.

Delay buffer variables, whose number is to be minimized by LP, are assigned to the PIs and all fanout branches of signals (PIs or gates) with fanouts more than 1. It is to be remembered that these buffers are used as variables in the LP and may not be physically inserted into the circuit. The linear program is developed as follows.

3.1. Variables

The variables, whose values will be determined by LP, can be split into two categories: *gate variables* and *buffer variables*. The gate variables are a set of three variables for each gate i:

- T_i : This is the maximum time at which the output of gate i can produce an event after the occurrence of an event at the PIs.
- t_i : This is the minimum time at which the output of gate i can produce an event after the occurrence of an event at the PIs.
- d_i : This is the inertial delay of gate i. Its value will be obtained as the output of the optimizer.

A buffer also has a similar set of three parameters.

3.2. Objective function

The injection of buffers into the circuit increases the area, power consumption, and overall delay of the circuit and hence an obvious objective would be to reduce their number. But this becomes a non-linear objective. However, reducing the sum of all buffer delays is a linear objective and is often effective. The LP therefore minimizes the total delay of the buffers.

3.3. Constraints

3.3.1 Initial constraints

The lower bound is set for every parameter of the gate using these constraints. We write the constraints as: $d_i \geq 1$ for every gate i, $d_i \geq 0$ for every buffer i, $T_i \geq 0$ for every gate and buffer i, and $t_i \geq 0$ for every gate and buffer i.

3.3.2 Gate constraints

First, let us deal with the constraints for a gate with a single fanout. This set of constraints includes the buffers, too. Consider the buffer 19 in Figure 2. Its fanin is buffer 16. Hence its set of constraints would be:

$$\begin{aligned} T_{16} + d_{19} &= T_{19}; \\ t_{16} + d_{19} &= t_{19}; \end{aligned}$$

These constraints are self-explanatory as the maximum and minimum delays at the input of the gate would just be added to the delay of the gate (or buffer) as the signal proceeds to the output. Now consider the case where there are more than 1 fanins, as in gate 7. Then we have:

$$\begin{aligned} T_7 &\geq T_5 + d_7; \\ T_7 &\geq T_6 + d_7; \\ t_7 &\leq t_5 + d_7; \\ t_7 &\leq t_6 + d_7; \\ d_7 &\geq T_7 - t_7; \end{aligned}$$

The first four constraints ensure that the parameter T_7 settles at the value that is the maximum (T_5, T_6) and t_7 would settle at the minimum (t_5, t_6). The condition for MTE is ensured by the last constraint.

Proceedings of the 16th International Conference on VLSI Design (VLSI'03)
1063-9667/03 $17.00 © 2003 IEEE

3.3.3 Overall circuit delay constraints

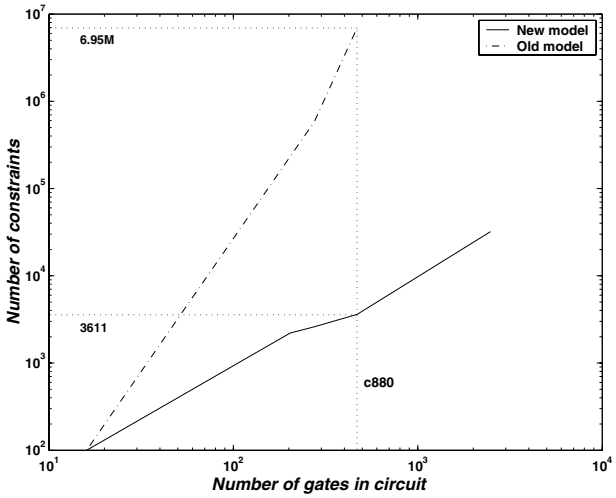
To ensure that the delay balancing does not slowdown the circuit beyond the specified limit we use a given upper bound on the maximum delay at the output. This can be ensured by placing that upper bound on parameter T of all gates feeding the primary outputs of the circuit. Thus we have additional constraints as:

$$T_{11} \leq maxdelay \quad \text{and} \quad T_{12} \leq maxdelay$$

where *maxdelay* is specified according to the application of the device and the amount of speed the user is willing to sacrifice for power savings. It is a user-defined parameter.

4. Why Is This Model Superior?

The main advantage of this technique is the linear size complexity of the constraint set with the size of the circuit. To illustrate the point we give the graph in Figure 3 with number of constraints for the ISCAS'85 benchmark circuits. As seen in the figure the benchmark circuit c880 needed 6.95 million constraints with the path enumeration model but only 3,611 constraints with the new model. The graph shows a linear increase with the number of gates for the new model. Though not obvious in the figure, the graph for the path enumeration model has an exponential rise and the constraint set could not be completed for some of the larger circuits (e.g., c6288 and c7552) due to the memory limit of the computer.



5. Results

5.1. Experimental procedure

We use a C++ program to parse the logic level circuit netlist and generate the constraint set in the AMPL format [11]. This constraint set is read into AMPL and the optimized delays of gates and buffers are obtained. We then use a power estimator [19] to compute the power savings obtained with new delays for the gates.

5.2. Experimental results

For the 1-bit adder circuit in Figure 2 the LP model in the AMPL format contained 94 gate constraints and 42 initial constraints. Since the minimum input-output delay is 6 units (assuming all gates of unit delay), there cannot be a solution for $maxdelay < 6$. When $maxdelay = 6$ is specified, we require two buffers. For $maxdelay = 7$ only one buffer is needed and for $maxdelay \geq 11$ no buffer is needed. In all cases, the condition of inequality in Equation 3 was satisfied at all multi-input gates by optimally combining delay balancing and hazard filtering to meet the overall *maxdelay* requirement. These results are exactly the same as was obtained by the path enumeration method [2].

5.3. Analysis of results

We have applied the above procedure to the ISCAS' 85 benchmark circuits and the results are tabulated in Table 1. The first row for every benchmark circuit shows results for the glitch elimination with no increase in overall delay and also specifies the number of buffers inserted for that *maxdelay*. The power estimation was done with a variable delay event-driven simulator that counts the number of all gate transitions produced by stuck-at fault vector set [13, 14, 17, 19]. The reference case (shown as unoptimized) is a unit delay circuit with no buffers added. The second row gives the result for the optimized circuit when the I/O delay is allowed to increase, sometimes as much as twice.

6. Transistor-Level Power Measurement

To evaluate the practicality of our delay assignment approach we implemented a 4-bit ALU circuit at the transistor level. This circuit contains 80 logic gates and the longest path in it has seven gates. The

Proceedings of the 16th International Conference on VLSI Design (VLSI'03)
1063-9667/03 $17.00 © 2003 IEEE

Table 1: Results from AMPL [11] for ISCAS'85 benchmark circuits.

Circuit	Average normalized power		Peak normalized power		No. of Vectors	*maxdelay* (#*gates*)	No. of Buffers
	Unoptimized	Optimized	Unoptimized	Optimized			
c432	1.0	0.72	1.0	0.67	56	17	95
	1.0	0.62	1.0	0.60	56	34	66
c499	1.0	0.91	1.0	0.87	54	22	48
	1.0	0.70	1.0	0.66	54	33	0
c880	1.0	0.68	1.0	0.54	78	24	62
	1.0	0.68	1.0	0.52	78	48	34
c1355	1.0	0.58	1.0	0.48	87	24	224
	1.0	0.57	1.0	0.48	87	48	192
c1908	1.0	0.69	1.0	0.59	144	40	219
	1.0	0.59	1.0	0.44	144	80	70
c2670	1.0	0.79	1.0	0.65	82	32	157
	1.0	0.71	1.0	0.58	82	64	35
c3540	1.0	0.64	1.0	0.44	200	47	239
	1.0	0.58	1.0	0.46	200	94	140
c5315	1.0	0.63	1.0	0.52	157	49	280
	1.0	0.60	1.0	0.45	157	98	171
c6288	1.0	0.40	1.0	0.36	141	47	294
	1.0	0.36	1.0	0.34	141	94	120
c7552	1.0	0.38	1.0	0.34	158	43	366
	1.0	0.36	1.0	0.32	158	86	111

LP contains 989 constraints. For $maxdelay = 7$ the solution used five delay buffers. When *maxdelay* was allowed to increase a no buffer solution was obtained for $maxdelay = 15$. It is this last design for which we implemented a transistor-level circuit.

The unoptimized version (used as a reference) was the circuit with the delay of each gate adjusted to 200ps. We connected a capacitor of 100pf, charged to 2.5V, as the power supply to the circuit. The voltage on the capacitor dropped gradually as the circuit consumed energy. This voltage drop, though not significant enough to change the operation of the circuit, gave an estimate of the energy consumed by the circuit [22].

The circuit was simulated on Spectre (by Cadence) for 50 random vectors and the energy consumed by the capacitor calibrated in picojoules is shown by the dotted curve in Figure 4. The gate delays determined by the LP were set for the CMOS implementations of the gates by appropriately varying the transistor widths. This optimized circuit was simulated for the same set of 50 vectors and the estimated energy consumption is shown by the solid curve in Figure 4. Comparing the slopes of straight lines approximating the two curves, we find that the optimized circuit consumes about 43% less power.

The aim of this experiment was to ascertain that the gate delay assignment can be successfully converted into a transitor level circuit. There are efficient techniques of *technology mapping* that can be employed [15, 25]. We are continuing this experiment to produce a layout and incorporate the routing delays in the constraint set.

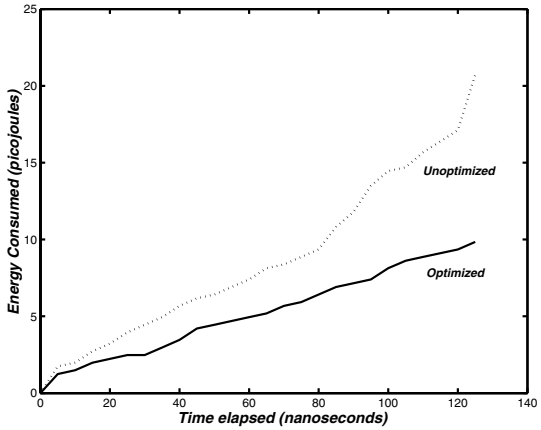


Figure 4: Energy consumption of optimized and unoptimized 4-bit ALU circuits obtained by circuit-level simulation.

7. Conclusion

The main result of this work is a new formulation of the LP with number of constraints that increase linearly with the number of gates in the circuit. Since the path enumeration has been dispensed with, the model can be used to solve larger circuits. Future

Proceedings of the 16th International Conference on VLSI Design (VLSI'03)
1063-9667/03 $17.00 © 2003 IEEE

directions of this work include working directly with transistor size parameters instead of gate delays, optimization of routing and layout, application to large synchronous datapath circuits, and power optimization of asynchronous circuits.

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 Mimimum 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 theProRISC 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] S. Datta, S. Nag, and K. Roy, "ASAP: A Transistor Sizing Tool for Area, Delay and Power Optimization of CMOS Circuits," in *Proc. of theIEEE International Symposium on Circuits and Systems*, May 1994, pp. 61–64.

[9] M. S. Elrabaa, I. S. Abu-Khater, and M. I. Elmasry, *Advanced Low-Power Digital Circuit Techniques*. Boston: Kluwer Academic Publishers, 1997.

[10] 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.

[11] R. Fourer, D. M. Gay, and B. M. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. South San Francisco, California: The Scientific Press, 1993.

[12] R. B. Hitchcock Sr., "Timing Verification and the Timing Analysis Program," in *Proc. of the 19th Design Automation Conf.*, June 1982, pp. 594–604.

[13] 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.

[14] S. M. Kang, "Accurate Simulation of Power Dissipation in VLSI Circuits," *IEEE Journal of Solid-State Circuits*, vol. 21, no. 5, pp. 889–891, Oct. 1986.

[15] K. Keutzer, "DAGON: Technology Binding and Local Optimization by DAG Matching," in *Proc. of the Design Automation Conference*, 1987, pp. 341–347.

[16] J. Monteiro and S. Devadas, *Computer-Aided Design Techniques for Low Power Sequential Logic Circuits*. Boston: Kluwer Academic Publishers, 1997.

[17] F. A. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Transactions on VLSI Systems*, vol. 2, no. 4, pp. 446–455, Dec. 1994.

[18] J. M. Rabaey and M. Pedram, *Low Power Design Methodologies*. Boston: Kluwer Academic Publishers, 1995.

[19] T. Raja and V. D. Agrawal and M. L. Bushnell, "A Reduced Constraint Set Linear Program for Low Power Design of Digital Circuits," Master's thesis, Rutgers, New Jersey, USA, 2002.

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

[21] 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.

[22] M. Shoji, *CMOS Digital Circuit Technology*. Upper Saddle River, New Jersey: Prentice Hall, 1988.

[23] J. M. Shyu, A. L. Sangiovanni-Vincntelli, 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.

[24] 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.

[25] C. Y. Tsui, M. Pedram, and A. M. Despain, "Technology Decomposition and Mapping Targeting Low Power Dissipation," in *Proc. of the Design Automation Conference*, June 1993, pp. 68–73.

[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 theIEEE International Symposium on Circuits and Systems*, May 2000.

Proceedings of the 16th International Conference on VLSI Design (VLSI'03)
1063-9667/03 $17.00 © 2003 **IEEE**