A Technique to Reduce Glitch Power during Physical Design Stage for Low Power and Less IR Drop

Vasantha Kumar B.V.P
J.N.T.U.H
Hyderabad
Andhra Pradesh, India

N. S. Murthy Sharma
J.N.T.U. H
Hyderabad
Andhra Pradesh, India

K. Lal Kishore
J.N.T.U. H
Hyderabad
Andhra Pradesh, India

ABSTRACT
A glitch compensation methodology is proposed in this paper which involves in reducing the undesired switching of combinational circuits in order to save dynamic power. The proposed methodology can be seamlessly integrated to existing physical design flow to reduce the glitch power which is one of the major contributing factors for both dynamic and IR drop. A glitch is an undesired transition that occurs before intended value in digital circuits. A glitch occurs in CMOS circuits when differential delay at the inputs of a gate is greater than inertial delay, which results into notable amount of power consumption. The glitch power is becoming more prominent in lower technology nodes. Introduction of buffers at the input of the Logic gate may reduce glitches, but it results into large area overhead and dynamic power. Hence, the proposed methodology will ensure low dynamic power consumption with less area. The pass transistor logic is used as a compensation circuit and a flow is also proposed for characterizing the pass transistor logic to cater different delay values. The proposed methodology has been validated using Synopsys 90nm SAED PDK.

General Terms
Physical Design, Compensation Circuit, Glitch Power, Dynamic Simulation and IR Drop Analysis.

Keywords
Dynamic power, digital logic circuits, low power design, CMOS delay devices, simulation, glitches.

1. INTRODUCTION
Complementary metal oxide semiconductor (CMOS) technology is used in all modern digital logic circuits. The power spent in CMOS can be classified as *dynamic power consumption* and *leakage or static power consumption*. Dynamic power consumption is due to the low impedance path between the rails formed through the switching devices. When input is given to the gate, there would be occurrence of one or more transitions at the output. At the output of the gate there are two types of transitions occurs, one which is due to actual transmission of the input signal resulting in desired functioning of the logic gate, is also called as *functional transition* Second, is due to transmission of unnecessary pulses through the logic gate resulting in undesired functioning of the gate, this is called as *spurious transition*. These spurious transition at the output of a logic gate is an outcome of difference in arrival time of various inputs. These unnecessary signals at the output of logic gate are known as glitches. Glitch power in modern circuits account for 20 to 70% [1] and it is 7 to 43% [2] of the dynamic power consumption. There are various published techniques to eliminate glitches in the logic circuits to accompany desired functioning of the logic circuit.

This work concentrates on elimination of the glitch power resulting due to spurious pulse transitions in the logic circuit using Synopsys design flow. Rest of the paper is organized as follows. Section 2 presents background of the concept and definitions. Section 3 presents prior work and existing techniques for this problem. Section 4 presents new formulation method used for glitch estimation method. Section 5 presents implementation of the proposed design and corresponding results obtained by simulation and analysis. Section 6 presents conclusions and scope of future research

2. BACKGROUND

2.1 Power optimizations:
The most prominent design aspect in this era is low power consumption. The continuing decrease in feature size and corresponding increase in chip density and operating frequency have made power consumption a major concern in VLSI design. As the technology advances, the transistor density of IC is moving at a rate proportional Moore's Law and the battery advancements are not at all proportional to the technology advancements. The incomparable battery advances will mandate low power methodologies and designs. In this perspective, the designer should undergo power optimization at each and every level of the design for a digital logic circuit [3]. Many different levels like algorithmic level optimizations in which designer could try to modify the functioning of the algorithm, RTL level optimizations in which designer could try to alter the implementation of the algorithm to realize the desired functioning of the logic circuit, circuit level optimizations in which designer could deal with the details of the basic circuit cells that are widely used for the desired functioning of the digital logic circuit, and package level optimizations. In this paper we need to explore the possibility of optimization ~~on~~ at transistor level to reduce the glitches occurring on input/outputs ports of a logic gate.

2.2 Delay of a gate:
The primary contributing element in the design aspect for digital logic circuits is Gate. Gate is majorly implemented in the complementary metal oxide semiconductor (CMOS) technology. Different combinations of logic gates can be used for performing all logic functions. A CMOS gate is constructed by combination of MOSFTs to realize a logic function. But MOSFET is not an *ideal* switch, as it provides a large but finite resistance (R_{on}) when it is *open* and it provides a small nonzero resistance when it is *closed* between source and drain terminals of the logic gate. For a CMOS

gate, the change in output signal follows the change in input signal with certain delay constraint. The on and off switching activity of the transistors in the logic gate depends on the slope of the input signals. So, the change in output signal depends on the low resistance (R_{on}) path provided by the "ON" MOSFETSs and charging or discharging of the output load capacitance (C_L). The delay of a logic gate depends on the amount of resistance and capacitance offered by current path, is called *gate delay* or *inertial delay.*

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

As explained above the inertial delay of logic gate is given by $R_{on}\times C_{L,}$ it can be varied by changing the width and length of the transistor [4], [5], [6]. In a logic gate, the rising and falling of signals transitions at the output depends on the switching activity of the pFETs and nFETs respectively in that gate. The output signal rises when current flows in the pFETs, similarly the output falls when current flows in the nFETs of the logic gate. So, the delay of rising transitions can be varied by changing the sizes of the pFETs and the delay of falling transitions can be varied by changing the sizes of the nFETs in the logic gate. By increasing the length and width of the transistor we can increase the resistance in the critical path, so that the delay of the current path will be increased. The delay is effectively changed by manipulating the length and width of the transistors in the gate.

2.3 Glitches and dynamic power:

Glitches are the spurious transitions which occur due to difference in arrival times of signals at the gate inputs. There have been a number of attempts made in the past to eliminate these spurious transitions. These are not needed for the correct functioning of the logic circuit. Power consumed by glitches is called as Glitch power. Every signal net of a gate needs to be transmitted at most once in every clock cycle. But in the real scenario there are output transitions switching more than once in every clock cycle and these unnecessary transitions will also consume power. So it is very advantageous to eliminate glitches in the circuits as power consumption is critical in today's chips. The flow of glitch in a digital logic circuit gate is shown in fig (1), [7].

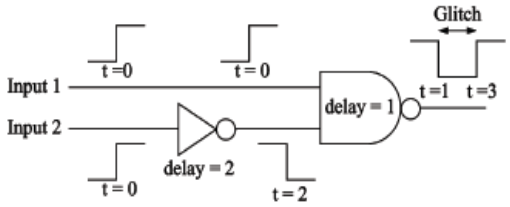


Fig. 1. Circuit showing the formation of glitches. The inverter has a delay of 2 units and the NAND gate has a 1 unit delay. Due to differing arrival times at the inputs of the NAND gate, the output produces a glitch consisting of two transitions [7].

In a logic gate, the number of edges in the transients at the output of the gate may equal to the number of arriving signals at the gate. The maximum difference in the arrival time of the signals at the inputs of the gate is called as *differential path delay*. It is also the maximum width of the possible glitch at the circuit output.

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

Consider fig (1), in the circuit we can see the unbalanced arrival times of the inputs due to the inverter circuit in the lower input path of the NAND gate. Thus the differential delay of the NAND gate is 2 units. This differential delay makes the NAND gate to switch 2 times more than the required functioning forming spurious transitions at the output which consume some dynamic power.

3. PRIOR WORK

Till now there are have been many techniques developed to eliminate glitches in a logic circuit, like delay balancing, hazard filtering, gate sizing, transistor sizing and minimum dynamic power LP technique etc.

3.1 Delay balancing:

In this method the inputs are made to arrive at the same time by inserting extra delay buffers on selected paths [8], [9], [10], [11], [12].

3.2 Hazard filtering:

In this method the gate delay is made greater than the differential delay at the inputs of the gate to filter the glitch [13].

3.3 Gate sizing:

In this method every gate is assumed to be an equivalent inverter [14-19].

3.4 Transistor sizing:

This method treats every transistor's size as a variable and tries to find a glitch-free design [20-25].

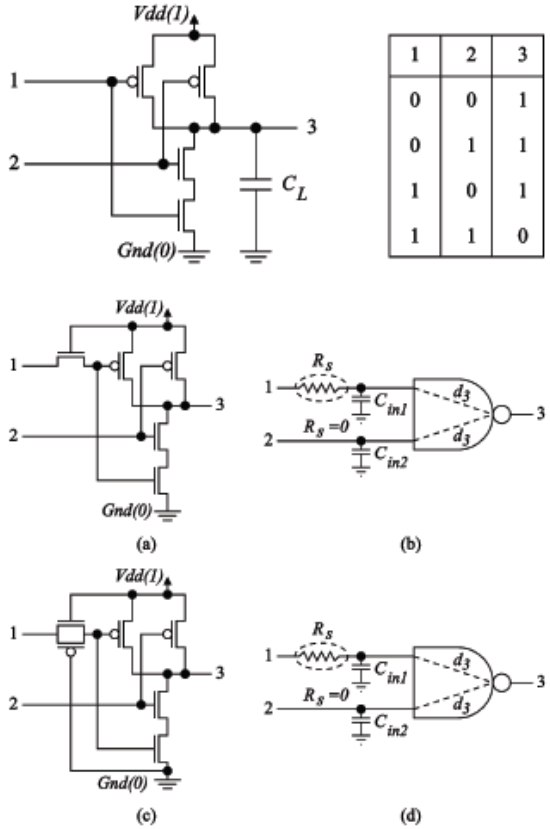
3.5 Linear programming:

In this method the gate delays are treated as variables and optimum delays are found by solving a linear program (LP) [26-28] which is implemented using AMPL programming method [29].

Drawbacks: from 3.5-3.4 these techniques are either greedy approaches or have non linear convergence problems. This problem was eliminated using *linear programming* (LP) technique converting the nonlinear convergence into linear convergence with less no. of constraint set. But it uses the delay buffers to optimize delay in the critical path, this is an overhead and which consumes some dynamic power.

3.6 Variable input delay method

Raja *et al* [7] proposed a new technique to design a gate which gives differing delays along the different input-output paths, is also called variable input delay method. Basic cell design is altered and a new design with pass transistor inserting at one of the inputs of the gate was made, this designed was used to replace the basic cells where the differential delay of the input of a gate is less than inertial delay of the gate. By varying the length and width of the pass transistor in the bounded limits required delay is obtained. Consider the NAND gate given in fig (2), [7] the delays through different I/O paths is given as:



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

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

$$d_{1\rightarrow3}=R_{on}\times C_{in1}+d_3$$
$$d_{2\rightarrow3}=R_{on}\times C_{in2}+d_3$$

Where C_{in1} and C_{in2} are input capacitances, R_{on} is series resistance at the input of the gate.

Drawback: The main drawbacks in using this variable input delay method is altering design cells (or) creating new library. This method is not flexible for re-spin designs

4. SIMPLIFIED GLITCH POWER REDUCTION PROPOSAL

In the proposed methodology instead of modifying existing standard cells we insert a compensation circuit with appropriate delay to mitigate the effect of glitch due to difference in input arrival to a digital logic gates. We emphasize on analyzing the effect of glitch power based on IR drop numbers during physical design stage with the presence of real parasitics. In the standard based design flow, once the design reaches the place-and route stage we do not have the freedom to alter the any parameter of the cells. But, here there is a flexibility to introduce new elements into the actual design of the circuit at the physical design stage. During physical design if the IR drop numbers are with the expectation then regular flows should be followed and there no need for checking the glitch power effect unless there is a will to reduce the power lower than specification. With respect to these thoughts we propose the following methodology described the flow chart as shown in Fig 3.

A typical implementation flow will start with a RTL simulation and moves on to place and route after successful completion of synthesis and timing. After completing place and route IR drop analysis is performed. If the reported power numbers are higher than the specification, then we move on to the proposed methodology to reduce the total power dissipation by reducing the glitch power component in it. Glitch power reduction flow starts with extraction of transistor level post layout netlist using Synopsys physical verification tools. Transient simulation is performed using HSPICE/HSIM. Based on the netlist active device count true spice or fastspice can be chosen for the transistor level simulation. The vectors for this transient simulation are provided directly from RTL simulation.

After completion of transient simulation, each data path in the waveform data is analyzed for glitches manually or through scripting using CustomWaveview®, a waveform viewer from Synopsys. Manual inspection is only recommended only for smaller designs. When using Scripting method a table can be created with potential glitch nets and corresponding glitch width. Based on the glitch width, designer can insert appropriate pass transistor cells with specific pre characterized delay and redo the flow again from place and route.

5. RESULTS

The proposed approach needs pre characterization of resistance cell created using pass gate. For this we used Synopsys 90nm SAED PDK for resistance cells creation. HSPICE® is used to measure the effective resistance values of this pass gate structure and tabulated in (Table 1). Different timing models and standard cells are formulated for different dimensions of pass gate structure.

LVDS architecture with sufficient physical data paths were taken to corroborate the proposed methodology. Design Compiler® is the tool used for carrying out synthesis; IC Compiler® is used for Floorplan, placement and routing. Prime Rail® is used for obtaining rail analysis to find out the potential IR drop hot spots. In order to obtain circuit simulation HSPICE/HSIM® is used to find out potential pins at which glitch power is more. As of now, using the above Table-1, the compensation cell is taken and inserted in the layout at problem pin. Hence, the results of the experiments with and without compensation in the problem cells are tabulated in TABLE II. LVDS design layout after place and route is shown in fig (4). IR drop results after place and route is overlayed in IC Compiler layout as shown in fig (5). The schematic of potential problem cell which is subjected to glitch is shown in fig (6). After finding the required delay and inserting the resistance cell IR drop is analyzed again and shown in fig (7). The schematic of the problem cell with compensation cell in shown in fig (8).

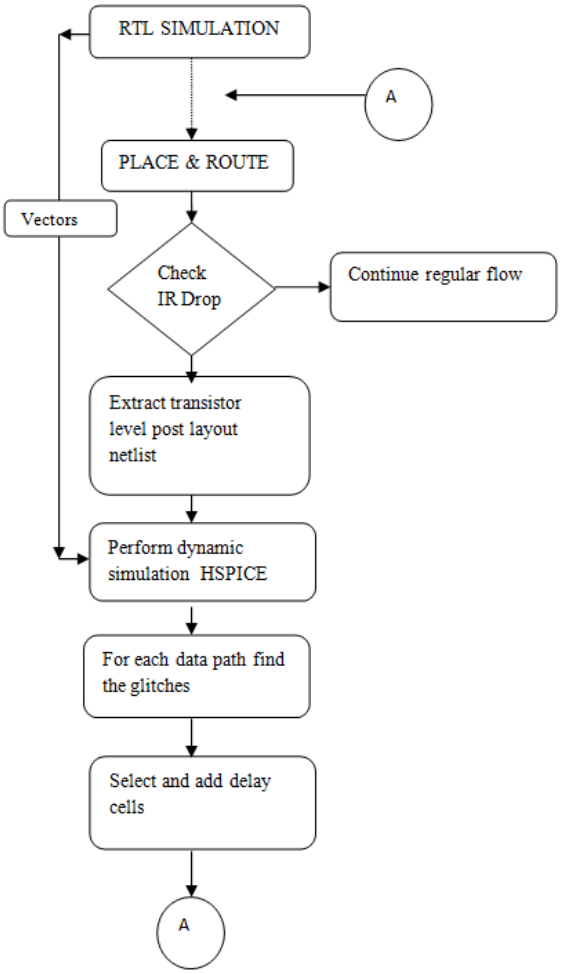


Fig 3: Flow diagram for Proposed Methodology

Table 1. Synthesized Resistance LUT For Compensation Circuit

S.No.	Generic 90nm Design Library, 1.2v MOS devices		
	W/L ratio for Tgate structure in µm		Equivalent series resistance value in Ω
	NMOS	PMOS	
1	0.12/0.1	0.36/0.1	3.01k
2	0.2/0.1	0.6/0.1	1.79k
3	0.4/0.1	1.2/0.1	894
4	0.6/0.1	1.8/0.1	597
5	1.0/0.1	3.0/0.1	361
6	2.0/0.1	6.0/0.1	184
7	4.0/0.1	12.0/0.1	95.6
8	6.0/0.1	18.0/0.1	66.2
9	10.0/0.1	30.0/0.1	42.7
10	12.0/0.1	36.0/0.1	36.8
11	14.0/0.1	42.0/0.1	32.6
12	20.0/0.1	60.0/0.1	25.1
13	40.0/0.1	120.0/0.1	16.3

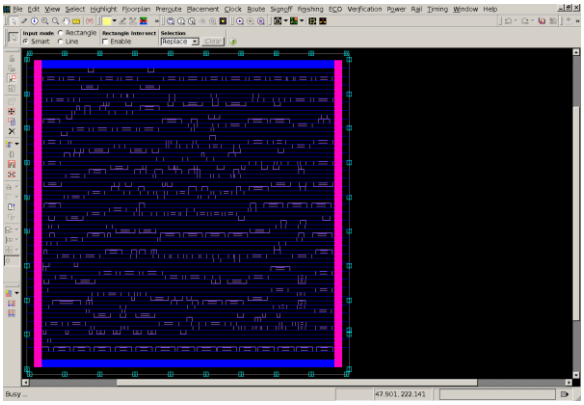


Fig 4: Initial lvds layout

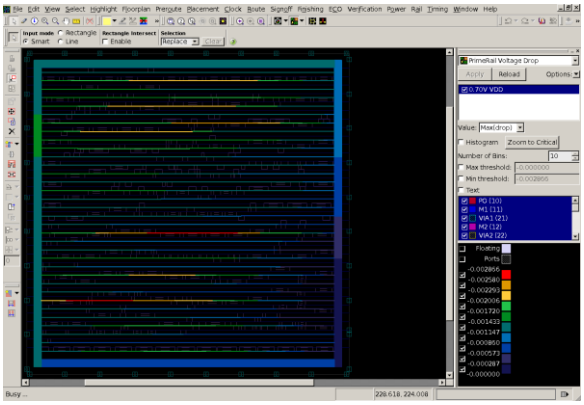


Fig 5: Overlayed IR drop results before compensation

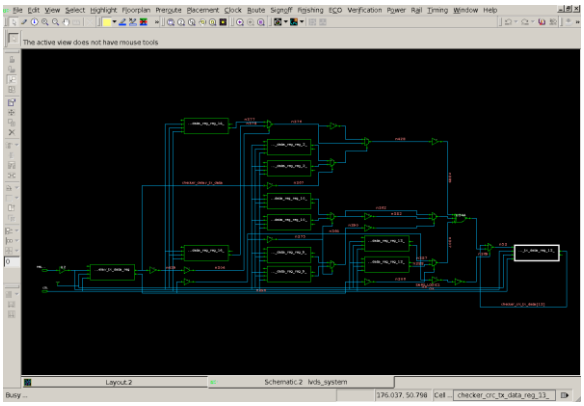


Fig 6: Schematic of layout highlighting cell with higher IR drop

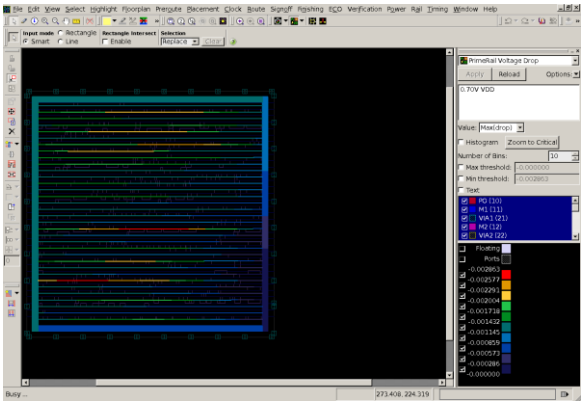


Fig 7: Overlaid IR drop results after compensation

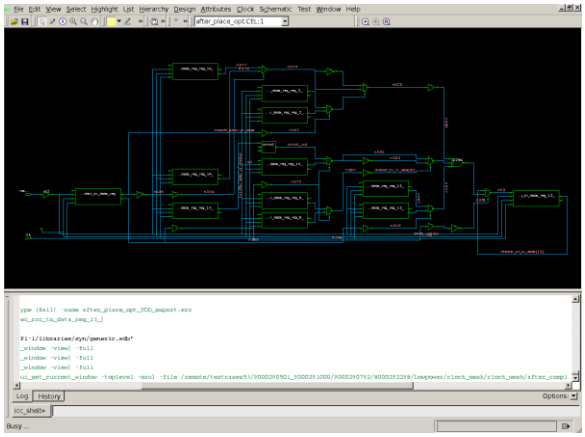


Fig 8: Schematic of layout highlighting cell after compensation

Table 2. Tabulated IR Drop Value With and Without Compensation Circuit

S. No	Peak IR Drop Cell	IR Drop Value (without compensation)	IR Drop Value (with compensation)
1	checker_crc_rx_data_reg_13_	2.868 mW	2.068 mW
2	checker_crc_rx_data_reg_7_	2.785 mW	2.185 mW

6. CONCLUSIONS

By adopting the proposed methodology the glitch power which is one of the major contributing factor for both dynamic and IR drop can be effectively reduced with less overhead and minimum impact on the existing flow. The proposed methodology will aid in bringing down the glitch power in the design seamlessly along with the strategies followed during logic synthesis

7. ACKNOWLEDGMENTS

We would like to thank Harish Balan, Director, Uno V. Nellore, Sr. Manger from Synopsys (India) Pvt. Ltd. for their encouragement and continuous support in doing this research using Synopsys tools.

8. REFERENCES

[1] A..Shen,A.Ghosh,S.Devadas and K.Keutzer, "On average power dissipation and random pattern testability of CMOS combinational logic networks,"*proc. ICCAD,pp.402-407,1992.*

[2] D.Brad and C.Vishweshwaraiah, "Inaccuracies in power estimation during logic synthesis," *proc. ICCAD,pp.388-394,1996.*

[3] 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.

[4] 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.

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

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

[7] T.Raja, V.D.Agarwal "Variable input delay CMOS logic for low power design", *IEEE Trans. Circuits syst.,vol.-17,no.10,October 2009.*

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

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

[10] 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.

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

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

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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

[26] 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.

[27] 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.

[28] 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.