A New Statistical Approach for Glitch Estimation in Combinational Circuits

Ahmed Sayed and Hussain Al-Asaad
Department of Electrical and Computer Engineering
University of California
Davis, USA

***Abstract* — Low-power consumption has become a highly important concern for synchronous standard-cell design, and consequently mandates the use of low-power design methodologies and techniques. Glitches are not functionally significant in synchronous designs, but they consume a lot of power. By reducing glitching activity, we can reduce the dominant term in the power consumption of CMOS digital circuits. In this paper, we present a new method to estimate the glitching activity for different circuit nodes. The method is robust and produces accurate glitch probability numbers early in the design cycle. It does not have much overhead and it alleviates existing compute-intensive algorithms/methods.**

I. INTRODUCTION

Low power consumption has become a highly important design aspect in this era and will become more and more important as we move to all mobile computing and communications. The transistor density of IC is growing at Moore's law rate and the incomparable battery advances will mandate lower power methodologies and designs.

There are different levels at which a designer could approach the problem of power consumption of a digital system. First, there are algorithmic level optimizations which try to modify the way an algorithm is functioning. Second, there are architecture and RTL level optimizations which alter the way an algorithm is implemented or the way the design is implemented to accomplish the same function, for example clock distribution to different functional blocks would be an architectural system level decision. Third, there are circuit level optimizations which deal with the intricacies of basic cells that are widely used in any digital system and the circuit implementation of the desired function. Fourth, there are the package level power optimizations and other levels as well. In this paper, we are mainly concerned with the circuit level, where we aim to estimate glitches at circuit nodes.

There are three major sources of power consumption in a digital CMOS circuit. These are summarized in the following equation:

$$P_{total} = p_t(C_L V V_{dd} f_{clk}) + I_{sc} V_{dd} + I_{leakage} V_{dd}$$

The first term of the equation represents the dominant switching component of power, where C_L is the sum of all switched loading capacitances, f_{clk} is the clock frequency or data rate of the circuit node and p_t is the probability that a power consuming transition occurs. In most cases, the voltage swing V is the same as the supply voltage V_{dd}. The second term is caused by the direct path short circuit current I_{sc}, which arises when both the NMOS and PMOS networks are on, conducting current from the supply V_{dd} to ground. The last term in the equation corresponds to the leakage current $I_{leakage}$, a factor that is growing more and more important as we develop deep submicron technologies.

According to the synchronous design paradigm, enough settling time is given for all intermediate transient values called the setup time before the clock event, which brings that stable value to the outside world with no glitches. The unfortunate part of this is that all these glitches still cause unnecessary power consumption in the form of short circuit current. Reducing glitching and switching is an ongoing area of research. In [1] we can see the significant effect glitching could have on the total switching of a circuit. In [2] and [3] different approaches to minimize glitches are discussed.

The problem with estimating glitches is that they are very process, voltage and temperature (PVT) dependent. This causes the model accuracy to become a very important factor in the estimation process. We assume in this work that we are at the typical corner. Estimating glitches at the typical corner is a very reasonable practice, since it is the expected silicon corner for operation. From [4] we see that the published statistical treatment for glitch estimation produces significant errors. This motivated us to use a more efficient methodology to reduce errors and be able to make reasonable predictions based on the models used.

This paper describes a method to estimate the glitches in combinational circuits and our future research will focus on the reduction of the wasted power due to glitches. The rest of this paper is organized as follows. Section II presents some definitions and assumptions. Section III presents the model used for the glitch estimation process. Section IV presents the simulation and estimation experiments. Section V presents the proposed methodology and its corresponding results. Finally, Section VI presents conclusions and future research.

II. Definitions and Assumptions

We define a glitch to be any spurious transient gate output in combinational circuits. There are various phenomena that can cause glitches and the main one is hazards in combinational circuits. A combinational circuit is said to contain a hazard if the output signals depend on the internal circuit delays, elements and interconnects [5]. We assume that all primary input signals of a synchronous system arrive at the same time. Please refer to [1] to qualify how much glitching might occur in a circuit.

Each gate type delay is modeled as follows:

$$\text{Gate Delay} = ID + \alpha * N_i + \beta * N_o$$

Where N_i (N_o) is the number of inputs (outputs) of the gate. The parameter α is estimated as the average effect of multiple inputs on the delay of the gate (often determined for the different gates from the library characterization). The parameter β is estimated as the average effect of the number of loads or output being driven by the gate, where the underlying assumption is that the length of the gate output net is proportional to the number of fan-out that load that net. Inertial delay *ID* is the average for different gate sizes. The averaging of each of the above constants in the model is averaged over the different transition values in the characterization of the library elements or cells.

Based on this delay model we can propagate the delays along all circuit paths to different inputs of each and every gate. The delay of the gate is dependent on the load, being it wires or interconnects or gate caps which are driven by this gate.

III. Glitch Generation and Propagation

For a glitch to be generated, the difference between the arrival times of different signals at the gate inputs should be greater than the inertial delay of the gate itself. Another condition is that the input switching pattern should be one of the patterns that generate a glitch. These two conditions must exist for a glitch to be produced on the gate output. The probability of a glitch to exist will be the decomposed into pattern and propagation probabilities [4]:

$$P(G) = P_{patt} * P_{prop}$$

Where P_{patt} is the probability that the pattern appearing at the inputs of the gate under consideration causes a glitch on the output of that gate. The pattern probability is fixed for a gate and it does not change, since it is inherent to that gate. For example the pattern probability of a 4 input AND gate with inputs *ABCD* is discussed below. From the K-Map of a 4-input AND gate only 12 transitions generate a glitch, these are first order transitions i.e. only two inputs change. Other gates pattern glitch probabilities are obtained similarly. Second order glitches will increase the run time and have a very insignificant effect on the estimation accuracy which will be made up for by finding the best fit model.

The pattern probability is dependent on the probability of the inputs of that specific gate to follow the glitch generation pattern. For example the transition from *ABCD* = 0111 to *ABCD* = 1101 through *ABCD* = 1111 will produce a static hazard glitch. We assume inputs' independence spatially and temporally and we assume that any input changes at most once every cycle of consideration. Given the probabilities of the primary inputs, the probabilities can be propagated throughout the circuit to all internal nodes and outputs. The propagation is done using a propagation algorithm [6] and not simulation as traditionally done.

The value of the P_{prop} is obtained as the number of pairs of paths out of all possible path pairs that have a delta delay that would cause a glitch. It was observed and assumed that the delta delay distribution is a normal distribution. This is true considering all paths leading to a gate, though it is more apparent in higher level gates. All the paths are taken one by one, if it leads to an input of the gate under consideration it will be compared to all other paths that lead to other inputs of that gate.

IV. Simulation and Estimation Experiments

There are two sets of experiments; the first set is simulation experiments that were done as gate level simulations. This means that the circuit under consideration is transformed first to a technology library and then a random set of stimuli (5000 vectors) is applied to the primary inputs. While the stimuli are being applied, the glitches are caught using a spy process on all the outputs of the gates under consideration. After that, the number of glitches are obtained and the ratio of input vectors to the number of glitches is calculated, which represents the probability of a glitch occurring. Different numbers of random inputs were tested and gave same glitch probability and different random inputs were tested and same glitch probabilities were obtained as well. These extra experiments were done to be able to quantify the effect of noise and nuisance variables on the simulation results.

The second set is running experiments for the estimation of glitches using a tool that we scripted in TCL to use the models described above and calculate the glitch occurrence probability. Since these calculations are not dependent on any random inputs, repeating the experiments generates the exact same numbers. The procedure of the script is as follows: first we calculate $P(x)$ at each and every node x in the circuit [6]. A node is defined as a gate output. P_{patt} would be calculated based on the gate type, as the example mentioned earlier. Then, we calculate the paths and their delays to each and every node of interest. From the paths' delay distribution, we obtain the glitch generation propagation probability at gate outputs. We finally estimate the glitch occurrence probability for the nodes of interest.

A key observation was that the glitch count is highly correlated to the level of the gate under consideration in the circuit being investigated. This was realized after analyzing and reasoning the results and the factors causing the results.

V. PROPOSED METHODOLOGY AND RESULTS

Due to the high errors seen in [4], we propose the method shown in Fig. 1 to accurately estimate glitches in combinational circuits.

The process of finding the best fit for the level of gate and glitching calculated data is done through Microsoft Excel built-in functions. Because the model is based on the gate type and the probability calculations are based on the same factor, it would be reasonable to look at groups of these gates by type. Note that cross gate type correlations might not be clear in the resulting numbers. To find the best fit, the variables and their combinations are considered for the best polynomial that models the independent variable. Fig. 2 shows the best fit results vs. the simulation results for 20 nodes of the ISCAS-85 C2670 circuit [7]. They are all NAND nodes and similarly we can get the best fit for the OR, AND and other gate types. It is worth noting that any negative values for glitch probability in Fig. 2 will be forced to zero value.

The benefit of finding the best fit model is to minimize the square error between the estimation data and the experimental data, and to consider higher order correlations in the variables considered and to include additional variables that might have an effect on the outcome of the experiments. The main benefit is that we can use that fit to predict the glitch probabilities for other gates and circuits.

The results are much better using this methodology. Table I shows the maximum absolute error in glitch probability for the 20 nodes of NAND gates and the percentage of that error with respect to the simulated value. Other gate types result in similar level of error.

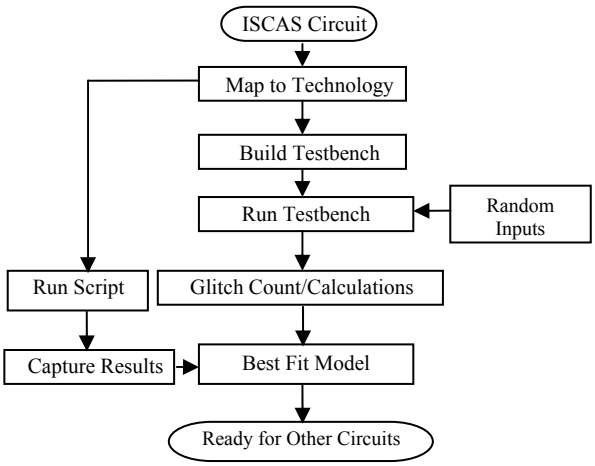

Figure 1. New proposed methodology.

TABLE I. THE NEW METHOD RESULTS FOR NANDS.

Max ABS Error	0.03568442
%	10.21729881

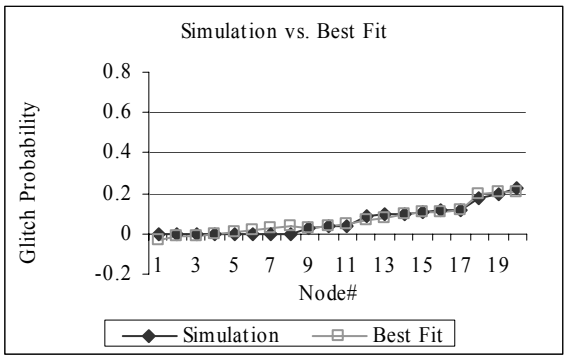

Figure 2. NAND gates simulated vs. best fit model results.

When we compare two sets of data one actual and one model produced, we care about the mean and the variance of both statistics of the calculated and measured values. We care about identifying the adequacy of the model to the actual values. Even if the absolute glitch probability numbers are not matching, if the relative comparison is meaningful, this could be a very insightful for a what-if analysis. Relative comparisons are more important for the next phase of our research where we extend this to dynamic power estimation for different design implementations.

The means are all equal between the best fits and the simulation results. The variance in Fig. 3 shows slight differences between the best fits and the simulations. These differences are not significant and do not affect the capability of predicting the glitching of other nodes in other circuits. It is worth noting that the variances of the models are slightly less than those of the simulations which mean that the calculated glitching will be slightly pessimistic using the model.

We investigated the correlation coefficients between the results of the actual simulation and those of the estimation for the three types of gates as shown in Table II and we have observed a very high correlation which shows the adequacy of the best fit model.

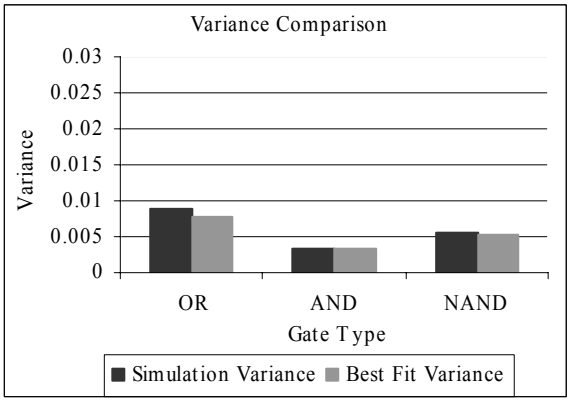

Figure 3. Simulation vs. best fit variance comparison.

TABLE II. CORRELATION COEFFICIENTS COMPARISON.

	Gate Type		
	OR	AND	NAND
Correlation	0.981826051	0.984194163	0.976010854

Residuals were investigated to show any inadequacy in the models used. They show that there is no dependency between the actual simulation results and the errors caused by the model calculations. This signifies that the model is adequate to actually model the simulations results.

Table III shows the coefficient of determination (r^2), which would indicate a strong relationship between the independent and the dependent variables. The F-distribution and F-values for each gate type are also shown in the table. Using Microsoft Excel's FDIST we find that an F value this high occurred for real. Then, we can conclude, that the regression equation is useful in predicting the assessed value of glitch probability.

To be able to evaluate the proposed methodology and models, we need to validate their prediction capability for other nodes in other circuits. We used both the best estimations based on the models above and ran simulations for the ISCAS-85 C7752 circuit [7]. We took three random sample nodes for each gate type. Table IV shows the high correlation of the results. Note that some nodes might correlate slightly better or worse than the best fit.

VI. CONCLUSIONS AND FUTURE RESEARCH

The Best-Fit models established are very good representations of the simulated data. These models were verified experimentally and showed accurate results in predicting the glitch probability of other gates in other circuits under consideration.

TABLE III. STATISTICS FOR GATE TYPES.

	Gate Type		
	OR	AND	NAND
r^2	0.963982395	0.968638151	0.952597188
F	45.88148585	52.94721452	34.44993376
Fdist	1.00638E-07	4.4314E-08	5.09945E-07

TABLE IV. CORRELATION COEFFICIENTS FOR THE C7752 NODES.

Gate Type	Correlation Coefficient
OR	0.999994
AND	0.960406
NAND	0.961078

The methodology proposed is robust and presents a way to get accurate glitch probability numbers early in the design cycle. The inaccuracies introduced by taking first order glitch generating transitions or gate delay models are accounted for and balanced by the statistical treatment to minimize the overall error. The methodology does not have much overhead and alleviates compute intensive algorithms or methods.

We are planning to extend the work presented in this paper to estimate the switching activity of combinational circuits and average dynamic power consumption. It is important to note that the relative switching and glitching of one design vs. the other is enough to compare them from a dynamic power perspective. We are expecting that our future research will lead to low power optimizations for some of the high switching nodes in the circuits under consideration.

REFERENCES

[1] J. Leijten, J. Van Meerbergen, and J. Jess, "Analysis and reduction of glitches in synchronous networks", *Proc. European Design and Test Conference*, 1995, p. 398.

[2] M. Hashimoto, H. Onodera, and K. Tamaru, "A power optimization method considering glitch reduction by gate sizing", *Proc. International Symposium on Low Power Electronics and Design*, 1998, pp. 221 – 226.

[3] H. Lee, H. Shin and J. Kim, "Glitch elimination by gate freezing, gate sizing and buffer insertion for low power optomization circuit", *Proc. Annual Conference of the IEEE Industrial Electronics Society*, 2004, pp. 2126 – 2131.

[4] Y. J. Lim and M. Soma, "Statistical estimation of delay-dependent switching activities in embedded CMOS combinational circuits", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 5, No. 3, pp. 309 - 319, September 1997.

[5] E. J. McCluskey, *Logic Design Principles with Emphasis on Testable Semi-Custom Circuits*, Prentice Hall, 1986.

[6] F. N. Najm, "Transition density, a stochastic measure of activity in digital circuits", *Proc. Design Automation Conference*, 1991, pp. 644 – 649.

[7] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran", *Proc. International Symposium on Circuits and Systems*, 1985, pp. 695 – 698.

[8] D. C. Montgomery, *Design and Analysis of Experiments*, 5th Edition John Wiley &sons, Inc., 2003.

[9] D. Rabe and W. Nebel, "New approach in gate-level glitch modelling", *Proc. European Design Automation Conference*, 1996, pp. 67 – 71.

[10] H. Mehta, M. Borah, M. Owens, and M. J. Irwin, "Accurate estimation of combinational circuit activity", *Proc. Design Automation Conference*, 1995, pp. 618 – 622.

[11] B. Welch, K. Jones, and J. Hobbs, *Practical Programming in TCL and TK*., 4th Edition, 2003.