

Input-specific Dynamic Power Optimization for VLSI Circuits

Fei Hu*
Intel Corporation
Folsom, CA 95630, USA
frank.hu@intel.com

Vishwani D. Agrawal
Auburn University
Auburn, AL 36849, USA
vagrawal@eng.auburn.edu

ABSTRACT

Literature proposes linear programming (LP) methods for glitch-less design of digital circuits. Considering the worst-case these methods ensure absence of glitches for any arbitrary state of primary input as well as internal signals. In this paper, we examine an unexplored aspect, i.e., glitch-free design with respect to a specific set of vectors (patterns). Introducing the logic-level concepts of glitch-generation patterns and glitch-generation probability, which are analyzable through logic simulation, we remove glitch filtering requirements from gates on which the given set of input vectors cannot produce glitches. We relax constraints of any existing LP either selectively or probabilistically. Such input-specific design from an LP model without process variation and another with process variation reduced the number of delay buffer overhead by up to 80% and 63%, respectively, while maintaining the power reduction and overall delay.

Categories and Subject Descriptors: J.6 [Computer-Aided Engineering]: Computer-aided design (CAD)

General Terms: Algorithms, Design

Keywords: Input specific, dynamic power optimization, glitch reduction

1. INTRODUCTION

Reduction of switching power dissipation of a circuit involves, among other things, glitch reduction. In conventional CMOS circuits, the spurious transitions at the output of a gate due to the differential delay of input paths are called *glitches* or *hazards*. Removal of such transitions can reduce the switching activity of a circuit and hence the switching power. The principal idea in glitch reduction is to find delay assignment for all gates in the circuit to reduce the differential path delays at gate inputs with respect to the inertial delays. Optimization techniques for glitch

reduction are the *balanced delay* [5] and *hazard filtering* [1] methods, implemented through a variety of algorithms such as *transistor sizing* [12, 14], *gate sizing* [3, 6], and *linear programming* [2, 7, 8].

A linear programming (LP) technique has the advantage that an LP solver derives a globally optimal solution in a relatively short time from the given model of the problem. Agrawal *et al.* [2] combined path balancing and hazard filtering in their LP model to determine the delay assignment for each gate. In subsequent work, their group proposed [8] an improvement reducing the complexity of the constraint set from exponential to linear in the circuit size. In another recent work, the use of a random delay model allows a robust glitch-free circuit design under given process-tolerance [7].

In all the previous LP models [2, 7, 8], the glitch optimization of the circuit is considered under arbitrary gate inputs. The LP solution ensures the absence of a glitch for any input vector sequence and for all input signal combinations at all gates. Such constraints result in an “overdesign” in the sense that glitches are virtually suppressed even for those signal states that are either impossible or only occur with very small probability. Restrictions on signal states can be due to two reasons, namely, circuit structure and functionally-relevant subsets of primary inputs. For a circuit where the total propagation delay is restricted, the conventional LP solution requires the insertion of many delay elements in non-critical paths. To reduce the additional power consumed by these elements, Raja *et al.* [9, 10, 11] have proposed new type of gates with different IO delays incorporating transmission-gates. Uppalapati *et al.* [13] used customized resistive feedthrough cells as delay elements, which consumed negligible amount of switching power. However small in number, delay elements [10, 13] add some capacitive loading that increases the per-transition dynamic power. Besides, they increase the total circuit area. Any reduction in the delay elements is therefore desirable.

In this paper, we explore a new aspect of the above problem, the application-specific circuit optimization. That is, we may only optimize the circuit for certain input sequences that will be applied to the circuit, e.g., functional vectors. Optimization of the circuit for these vector sequences ensures the low power dissipation when the circuit is in use and it can lead to a better solution because the optimization is customized to the application. In our experiments on ISCAS’85 benchmark circuits, when the input-specific optimization was considered in the previously published LP models of Raja *et al.* [8] and Hu [7], the number of required delay buffers (overhead) dropped by up to 80% and 63%, re-

*Formerly with Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED’06, October 4–6, 2006, Tegernsee, Germany.

Copyright 2006 ACM 1-59593-462-6/06/0010 ...\$5.00.

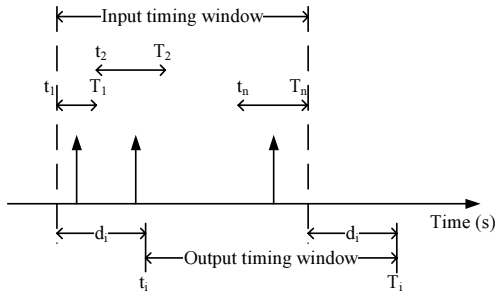


Figure 1: Illustration of timing window at gate i .

spectively, while maintaining the power reduction and overall circuit delay.

2. BACKGROUND AND MOTIVATION

Previous LP modeling [7, 8] considers the optimization of the circuit in the *worst-case*. As shown in Figure 1, a timing window of signal arrival time $[t_i, T_i]$ is propagated throughout the circuit [8], where t_i is the earliest arrival time and T_i is the latest arrival time for gate i . A constraint $d_i > T_i - t_i$ is imposed for each gate inertial delay d_i . Therefore, the LP solution ensures that the gate is free from glitches for *any* arbitrary signal transition at the inputs of the gate. However, we observe, this worst-case optimization may have introduced too much pessimism into the solution. For a circuit where the total propagation delay is restricted by design, the LP solution may require insertion of a large number of buffers on non-critical paths. As we know, the insertion of buffers is costly and their number should be kept as small as possible because it either increases the total power dissipation of the circuit (assuming conventional buffers) or the total area of the circuit (assuming resistance type of buffers [13]).

In general, the worst-case optimization could mean overdesign. We may not need the circuit to be optimized for *all* possible input sequences. On the contrary, we may only want the circuit be optimized for the set of input sequences that will actually be applied to the circuit while it is working, for example, the functional vectors. These input sequences can be a highly biased set depending on the system environment. Optimization of a circuit specific to such vector sequences ensures that the optimized circuit maintains the low power dissipation under the given system environment. At the same time, we are able to achieve a better solution with reduced overhead because the optimization is more customized.

3. GLITCH GENERATION

We discuss the generation of glitches and introduce the concepts of *glitch-generation pattern* and *glitch-generation probability*.

3.1 Glitch-generation pattern

Glitches and *hazards* refer to the spurious transitions at a gate output caused by differential delays of paths arriving at its inputs. Two factors are essential for glitch generation, i.e., transitions and path delays. Our treatment here is similar to that in path delay testing where only signal transitions and not the specific delays are considered [4]. We define a *glitch-generation pattern* for a gate as the input vector pair

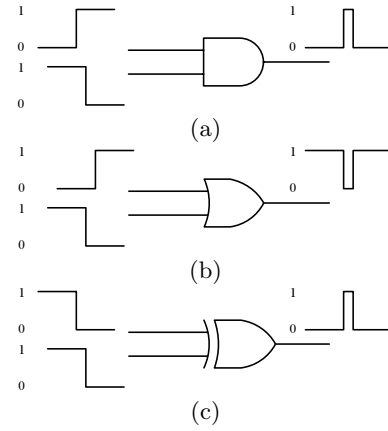


Figure 2: Glitch-generation in two-input gates.

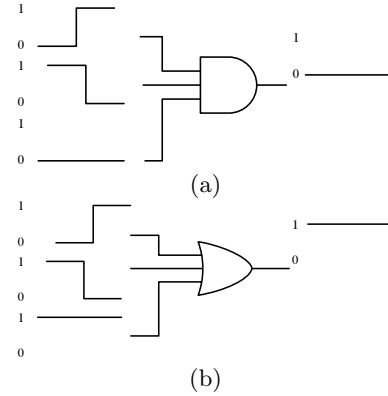


Figure 3: Glitch-suppression in multi-input gates by controlling value.

that can potentially generate a glitch at the output of the gate for some arbitrary input and inertial delays.

As shown in Figure 2, glitch-generation patterns for a two-input AND/OR gate are those vector pairs that produce two opposite transitions on different inputs. However, for a two-input XOR gate, a glitch can be potentially generated as long as both inputs have transitions.

For a gate with more than two inputs, a glitch cannot be generated if there is a steady controlling value (e.g., 0 for a AND gate) at any input of the gate. Therefore, the glitch-generation patterns for a multi-input AND gate will be those vector pairs that produce opposite transitions at any two inputs and no constant 0's at any other input. Similarly, the glitch-generation patterns for a multi-input OR gate will be those vector pairs that produce opposite transitions at any two inputs and no constant 1's at any other input. Since there is no controlling value for an XOR gate, the glitch-generation patterns for a two-input XOR gate are those vector pairs that produce transitions on both inputs. Figure 3 shows the effect of a controlling value on glitch generation.

3.2 Glitch-generation probability

We define *glitch-generation probability* P_g for a gate as the probability that a glitch-generation pattern-pair occurs at the inputs of that gate. The occurrence of a glitch means that the *steady-state* signal values during two consecutive clock periods at inputs of the gate match a glitch-generation

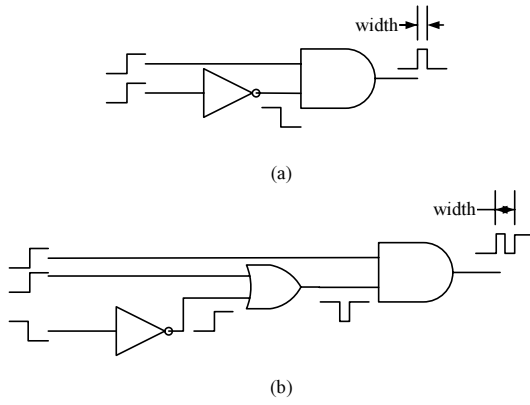


Figure 4: Hazard generation in logic circuits: (a) static hazard, (b) dynamic hazard.

pattern for that gate type. For a given set of N primary input vectors, glitch-generation probability for all gates can be obtained through zero-delay logic simulation of the circuit. Let us denote the number of times a glitch-generation pattern occurs at the input of gate i by $N_g[i]$, the glitch-generation probability for gate i , $P_g[i]$, is calculated as

$$P_g[i] = \frac{N_g[i]}{N} \quad (1)$$

4. INPUT-SPECIFIC OPTIMIZATION

With the measure of glitch-generation probability, we can selectively relax the constraints for gates where glitches are unlikely to occur. This input-specific optimization technique is applied first to the basic LP model [8] and then to the process-variation-resistant LP model [7].

4.1 Application to the basic LP model

First, we apply the input-specific optimization to the previous basic LP model [8]. This will achieve a glitch-free circuit under the given set of input sequence. Our input-specific optimization is a “static” analysis, meaning that only probabilities (and not the signal values) of glitch generation are the basis for eliminating (relaxing) some LP constraints. As shown in Figure 4, glitches in a practical circuit can be either generated at a gate or propagated from the previous stages of the circuit. Our definition of glitch-generation probability only captures potential glitch generation and ignores possible glitch propagation from the previous stage.

Clearly, the accuracy of the glitch-generation probability to represent the chance that a glitch can be produced is strongly affected by the ratio of propagated glitches. Only when propagated glitch does not exist or has a negligible probability, can our glitch-generation probability represent the chance correctly. For the relaxed constraints, we assume that no (or a negligibly small number of) glitches are propagated from the previous stages of the circuit.

4.1.1 Selectively relaxed LP constraints

Assuming that no glitch is being propagated throughout the circuit, glitch-generation probability of a gate represents the chance that a glitch can be produced at the output of the gate if no proper path balancing or glitch filtering is done. For gates with zero glitch-generation probability, a

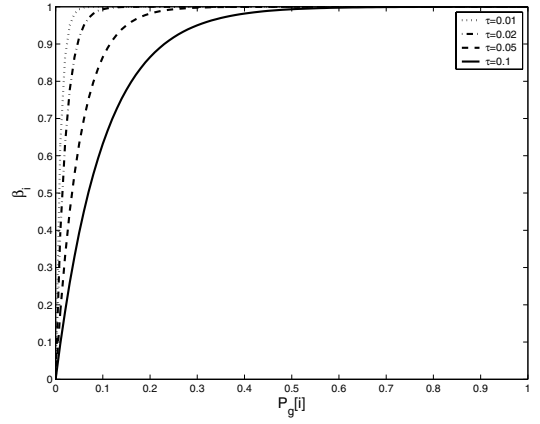


Figure 5: Function β_i for various selectivity factors τ .

glitch-generation pattern will never be produced at the gate inputs by the given primary input vector sequence. It also means that a glitch will never occur no matter how path delays or gate delays change. Under this circumstance, we remove the glitch-filtering constraint for that gate from the LP.

The original glitch-filtering constraint for gate i has the form [8]:

$$d_i > T_i - t_i \quad (2)$$

In the input-specific optimization, it is modified to

$$d_i > (T_i - t_i) \cdot \beta_i \quad (3)$$

where $\beta_i \in \{0, 1\}$ is a constant determined by the glitch-generation probability of gate i :

$$\beta_i = \begin{cases} 0 & \text{if } P_g[i] = 0 \\ 1 & \text{if } P_g[i] > 0 \end{cases} \quad (4)$$

This essentially retains the glitch-filtering constraints only for gates with non-zero glitch-generation probability. Note that such selective relaxation of constraints does not change the totally glitch-free property (i.e., no glitches are generated) in the resulting circuit because there is no need to suppress glitch propagation given that none is generated.

4.1.2 Probabilistically relaxed LP constraints

The selection of gates for glitch-elimination can be probabilistically generalized to allow even more relaxed constraints. The resulting LP solution will not guarantee that the circuit is totally glitch-free. However, it provides designers a trade-off between glitch power dissipation and cost (number of delay elements inserted) for a given critical delay requirement. We now replace the step function in Equation 4 with

$$\beta_i = 1 - e^{-P_g[i]/\tau} \quad (5)$$

Here, β_i is an exponential function of the glitch-generation probability $P_g[i]$ with a selectivity factor τ . The function β_i with τ as parameter is illustrated in Figure 5. The adoption of an exponential function has two advantages. First, for gates where glitches are more likely to occur, the glitch-filtering constraint is strictly enforced ($\beta_i = 1$). Second, for gates where glitches are less likely to occur, the glitch-filtering constraint is relaxed accordingly. The fast rising slope of the exponential function for small $P_g[i]$ ensures that

only a small number of glitches will be generated and propagated to the subsequent stages, which supports our assumption on neglecting the propagation of glitches.

By varying the selectivity factor τ , $0 \leq \tau \leq \infty$, a designer can adjust the slope of the function β_i . For a larger τ and milder slope of the function β_i , the circuit will consume relatively more power by allowing some glitches. At the same time, it will reduce the number of inserted delay elements for the same critical delay requirement. Designers can adjust the value of τ to obtain the desired solution according to their specific needs.

4.2 Application to process-variation LP model

Next, we apply the input-specific optimization to a process-variation-resistant LP model. The original LP formulation [7] considers intra-die variations of gate delays. Gate delays d_i are random variables and are assumed to have truncated normal probability distributions. A gate i has a nominal (also the mean) delay μ_{d_i} and standard deviation σ_{d_i} . All gates are assumed to have the same normalized standard deviation given by $r = \sigma_{d_i}/\mu_{d_i}$. The time window (Figure 1) at the output of a j -input gate i , $W_i = \max\{T_j\} - \min\{t_j\}$, is also a random variable with mean μ_{W_i} and standard deviation σ_{W_i} . The following inequality in the LP, which determines the nominal gate delays μ_{d_i} 's, ensures that gate i can produce a glitch with only a very small probability [7]:

$$\mu_{d_i} - \mu_{W_i} > 3 \cdot k(\sigma_{W_i} + r \cdot \mu_{d_i}) \cdot \alpha \quad (6)$$

where k is a constant ($1/\sqrt{2} \leq k \leq 1.0$) whose value is taken as 0.85, and $\alpha \leq 1.0$ is an optimism factor. This glitch-filtering requirement for gate i ensures that the inertial delay to gate exceeds the timing window in spite of the process variation. We modify the Inequality 6 for all gates i as,

$$\mu_{d_i} > [\mu_{W_i} + 3 \cdot k(\sigma_{W_i} + r \cdot \mu_{d_i}) \cdot \alpha] \cdot \beta_i \quad (7)$$

This glitch-filtering requirement on the delay of gate i is relaxed by a factor β_i . When $\beta_i = 0$, glitch-filtering constraint is altogether removed. As before, β_i is a function of $P_g[i]$ and can be chosen from Equation 4 or Equation 5.

It should be noted that constraints 6 and 7 require that the glitch-filter condition is satisfied even when the delays vary as much as three times standard deviation [7]. Such constraints are pessimistic and sometimes lead to “no solution” that meets the overall circuit delay budget. Values of $\alpha < 1.0$ reduce the pessimism to permit a solution while allowing a small number of glitches.

4.2.1 Optional tuning

Under process variation, the overall delay for an optimized circuit will not be a constant. The delays of critical paths are random variables and, therefore, the overall delay of the optimized circuit is a random variable with certain mean and variance. As illustrated below, under process variation, a solution to the input-specific optimization can lead to cases that must be avoided.

Consider the example shown in Figure 6. Under the input-specific optimization, glitch-filtering constraints for all AND and NAND gates are removed because the second PI to lower AND gate is always 0 in the specified input vectors. Delays for these AND/NAND gates are all set to the minimum value, $d_i = 1$, by the LP. The signal arrival time for the AND gate is between 20 and 40 due to the logic enclosed in the cloud. Given that the overall delay of the circuit

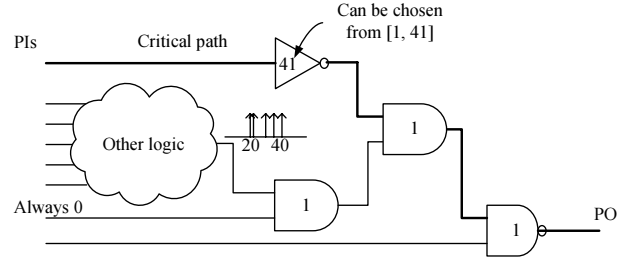


Figure 6: An undesirable solution under process variation when the input-specific optimization is applied directly. Bold lines indicate the critical path. The numbers on gates are their inertial delays.

should not exceed 43, the delay of the inverter can be chosen anywhere from a minimum value $d_i = 1$ to a maximum value of $d_i = 43 - 2 = 41$. However, in some cases, the LP solver will choose $d_i = 41$ if no constraint prevents it from doing so. This solution is undesired under process variation. The critical path PI, inverter and PO is unnecessary. This path will dominate the critical delay of the circuit under the process-variation and result in the degradation of critical delay distribution.

To avoid this undesirable solution, we include an additional term in the objective function. The original objective of the the LP model was to minimize the total buffer delays (which is a linear approximation for the number of delay buffers):

$$\text{Minimize } \sum_j d_j, \quad j \in \text{all delay buffers} \quad (8)$$

For input-specific optimization under process-variation, this is replaced by

$$\text{Minimize } \sum_j d_j + \frac{TF}{N} \sum_i d_i, \quad j \in \text{delay buffers}, \quad i \in \text{gates} \quad (9)$$

where the constant $TF \geq 0$ is a tuning factor, N is the total number of gates other than delay buffers.

When $TF > 0$, the tuning option is turned on. The value of TF is kept much smaller than 1.0 so that its impact on the overall optimization is minimized. However, as long as $TF > 0$, the LP solver is forced to minimize those gate delays that do not affect any constraints. With this tuning option, the gates on the dominating paths will be assigned minimum (rather than arbitrary) delays.

5. EXPERIMENTAL RESULTS

ISCAS'85 benchmark circuits were optimized by the input-specific optimization methods. Two input-specific optimization methods are illustrated. “IS-Opt1” is the input-specific optimization added to the previous basic LP model [8]. “IS-Opt2” is the input-specific optimization added to the previous process-variation-resistant LP model [7]. Results are compared to “un-optimized” circuits (“Un-opt”) and “optimized” circuits from the basic LP model [8] (“Opt1”) or the process-variation-resistant LP model [7] (“Opt2”). Same as in the published work [8], we use a unit-delay circuit as the un-optimized circuit, where each gate has a delay of one unit. Due to the space limitation, experimental results for the probabilistic relaxation method (Section 4.1.2) are not included.

Table 1: Input-specific optimization of ISCAS’85 benchmark circuits without process-variation.

Cir.	Max delay	Opt1 [8]			IS-Opt1		
		Avg. Pwr.	Cir. Delay	No. Buf.	Avg. Pwr.	Cir Delay	No. Buf.
c432	34	0.74	34	66	0.74	35	66
	68	0.74	68	58	0.74	69	41
c499	22	0.94	22	48	0.94	22	33
	33	0.94	33	0	0.95	33	0
c880	48	0.54	51	35	0.54	49	32
	120	0.54	121	30	0.54	122	24
c1355	48	0.93	48	192	0.93	48	113
	120	0.93	121	128	0.93	120	25
c1908	80	0.53	82	62	0.54	86	52
	200	0.54	203	34	0.53	204	3
c2670	64	0.74	65	34	0.74	66	30
	160	0.74	163	9	0.74	162	1
c3540	94	0.59	95	139	0.59	101	122
	235	0.59	239	78	0.59	239	73
c5315	98	0.56	100	167	0.56	104	170
	245	0.56	249	53	0.56	250	52
c6288	228	0.13	226	870	0.13	228	870
	620	0.13	620	857	0.13	620	853
c7552	86	0.52	89	91	0.52	88	84
	215	0.52	220	44	0.52	221	38

5.1 Input-specific optimization

The power dissipation and critical delay for “Opt1” and “IS-Opt1” are shown in Table 1. “IS-Opt1” adopted the selectively relaxed LP solution (Section 4.1.1). Similar to that in [8], circuits were simulated using an event-driven logic simulator with sequences of input vectors. Meaningful vectors would have been functional inputs. However, in the absence of such vectors or the functional information for these circuits, we either used test vectors or random vectors. For smaller circuits (i.e., c432 to c1355) complete gate level test vectors (with 100% stuck fault coverage) were used. For larger circuits, 50 random vectors with signal probability of 0.5 were used. Load capacitances for gates were assumed to be in proportion to the number of fanouts. Inserted delay buffers were assumed to be of resistance type and any additional power consumption by them was neglected. The average power for each circuit was normalized to the power dissipated by its un-optimized version, i.e., the un-optimized circuit has the power dissipation value of 1.

In Table 1, “Maxdelay” is the maximum specified critical delay parameter supplied to the LP. Clearly, the input-specific optimization is able to reduce the number of buffers inserted while maintaining the same performance in terms of power dissipation and critical delay. Depending on the vectors and circuits, a varying degree of improvement is achieved. In some cases the number of buffers inserted is reduced by up to 80%. Meanwhile, the power dissipation and critical delay values are the same or very close for “Opt1” and “IS-Opt1” in most cases.

5.2 Input-specific optimization under process-variation

5.2.1 Power analysis

Power dissipation and number of buffers inserted by “Opt2” and “IS-Opt2” are shown in Table 2. Under the process-variation, power dissipation of a circuit varies from sample to sample. Monte-Carlo simulation method is used where 1,000 sample cases of the optimized circuit were simulated. For each of these samples, as in [7], gate delays were in-

dependently sampled from normal distributions assuming 15% intra-die and 5% inter-die delay variation. In these experiments, “IS-Opt2” uses the selectively relaxed LP of Section 4. The tuning option of the objective Function 9 was turned on only for c1908, c3540, and c6288, where TF is chosen to be $\frac{1}{D_{max}}$. D_{max} is the maximum critical delay parameter.

In Table 2, “Nom. Pwr.” represents the nominal power dissipation when no process-variation exists; “Mean Pwr.” represents the mean value of the power distribution; and “Max Dev.” represents the difference ratio between the maximum value of the power distribution and the power dissipation under no process-variation. “Max Dev.” shows the degree of the deviation of average power from its design value due to the process-variation. All power values were normalized to the power dissipation of the un-optimized circuit.

We see that in all cases power dissipation of optimized circuits by “Opt2” and “IS-Opt2” is either the same or has only a slight difference. However, “IS-Opt2” achieves a solution with smaller number of delay buffers. Our technique, when compared to the reported input independent optimization [7] requires 40-60% fewer delay elements. The reduction of buffers is more obvious for larger D_{max} for each circuit. This is because for a smaller D_{max} , the optimization is more difficult. Removing of glitch-filtering constraint has a smaller effect on the reduction of buffers. Up to 63% reduction in the number of buffers is achieved for c2670 circuit.

Note that some of these examples use random inputs for demonstration. In an actual design, the input to combinational logic, when extracted from the system level simulation, will be further restricted. For example, the state-space of control logic may be much smaller than what is modeled by random inputs. Therefore, greater savings by this technique can be expected.

5.2.2 Delay analysis

The critical delays under process-variation are shown in Figures 7 and 8. “Nom. Delay” indicates the critical delay of the circuit under no process-variation, i.e., the nominal value of the critical path delay. We also show the maximum deviation (“Max. Dev.”) of the critical delay from its intended value under the process-variation. We see that “Opt2” and “IS-Opt2” have equivalent performances in all cases. From the power dissipation results in Table 2 and these figures, we can conclude that the input-specific optimization method “IS-Opt2” achieves a better solution for a given input sequence. It maintains the same power and delay performance while reducing the overhead in terms of the number of delay buffers inserted.

6. CONCLUSION

In this paper, we have explored a new aspect of low-power optimization for VLSI circuits and proposed the input-specific optimization techniques. We consider optimizing the circuit for a given input sequence that may be specified for the circuit. We define the concept of glitch-generation probability. By observing the glitch generation probability for each gate, we can adaptively relax the glitch-filtering constraint. The experimental results show that we are able to obtain a better solution with fewer delay buffer insertions while maintaining similar power reduction and delay performance as before. Up to 80% and 63% reductions in delay buffer overheads have been achieved in our experiments.

Table 2: Power dissipations and number of delay buffers inserted for input-specific optimization of ISCAS’85 benchmark circuits under process variations.

Cir.	D _{max}	Un-opt	Opt2 [7]				IS-Opt2			
		Nom. Pwr.	Nom. Pwr.	Mean Pwr.	Max Dev. (%)	No. Buf.	Nom. Pwr.	Mean Pwr.	Max Dev. (%)	No. Buf.
c432	50	1.0	0.74	0.76	11.1	88	0.74	0.76	9.3	81
	99	1.0	0.74	0.74	3.7	106	0.74	0.74	3.3	76
c499	32	1.0	0.94	0.95	2.0	88	0.94	0.95	1.9	88
	48	1.0	0.94	0.95	1.0	129	0.94	0.95	1.8	58
c880	70	1.0	0.54	0.59	18.2	57	0.54	0.59	20.4	38
	174	1.0	0.54	0.55	8.6	62	0.54	0.56	9.0	38
c1355	70	1.0	0.93	0.98	10.2	305	0.93	1.01	13.1	253
	174	1.0	0.93	0.94	3.0	305	0.93	0.95	4.7	160
c1908	116	1.0	0.52	0.64	35.8	135	0.52	0.64	34.7	107
	290	1.0	0.52	0.58	21.4	190	0.52	0.57	18.4	104
c2670	93	1.0	0.74	0.80	13.6	249	0.73	0.79	11.3	186
	232	1.0	0.73	0.76	6.2	211	0.73	0.75	4.3	79
c3540	137	1.0	0.59	0.66	17.8	281	0.59	0.65	15.6	247
	341	1.0	0.59	0.62	10.1	311	0.59	0.61	7.4	188
c5315	143	1.0	0.55	0.63	20.8	399	0.55	0.63	21.0	389
	356	1.0	0.55	0.60	13.4	418	0.55	0.60	13.2	413
c6288	331	1.0	0.13	0.38	223.8	1121	0.13	0.38	225.2	1115
	899	1.0	0.13	0.26	125.3	1473	0.13	0.26	125.5	1243
c7552	125	1.0	0.52	0.59	18.7	481	0.52	0.58	18.1	389
	312	1.0	0.52	0.56	11.8	645	0.52	0.55	10.9	520

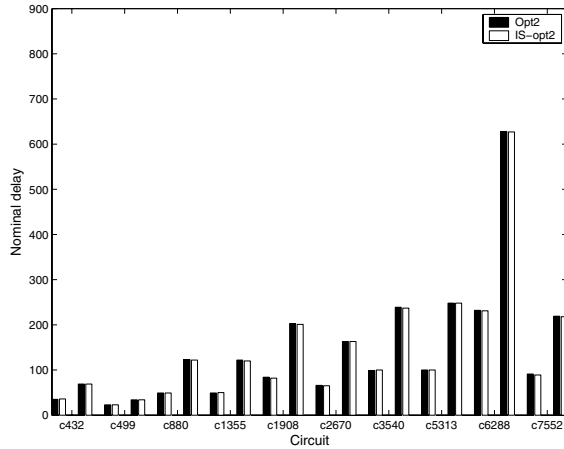


Figure 7: Nominal critical delay for optimized ISCAS’85 circuits.

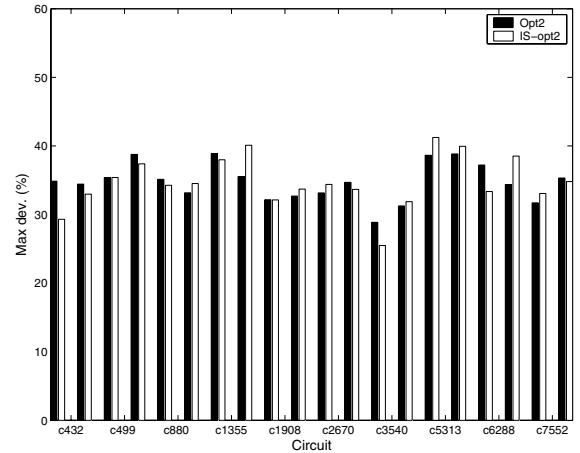


Figure 8: Maximum deviation of critical delay for optimized ISCAS’85 circuits.

7. REFERENCES

- [1] V. D. Agrawal, Low Power Design by Hazard Filtering, In *Proc. Intl. Conf. on VLSI Design*, pp. 193–197, 1997.
- [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. Intl. Conf. on VLSI Design*, pp. 434–439, 1999.
- [3] M. Berkelaar and E. Jacobs, Using Gate Sizing to Reduce Glitch Power, In *Proc. the ProRISC Workshop on Circuits, Systems and Signal Processing*, pp. 183–188, 1996.
- [4] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*, Springer, Boston, 2000.
- [5] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, Boston, 1995.
- [6] S. Dutta, S. Nag and K. Roy, ASAP: A Transistor Sizing Tool for Area, Delay and Power Optimization of CMOS Circuits, In *Proc. IEEE ISCAS*, pp. 61–64, 1994.
- [7] F. Hu, *Process-Variation-Resistant Dynamic Power Optimization for VLSI Circuits*, Ph.D. Dissertation, Auburn University, Dept. of ECE, Auburn, Alabama, May 2006.
- [8] T. Raja, V. D. Agrawal and M. L. Bushnell, CMOS Circuit Design by a Reduced Constraint Set Linear Program, In *Proc. Intl. Conf. on VLSI Design*, pp. 527–532, 2003.
- [9] T. Raja, V. D. Agrawal and M. L. Bushnell, CMOS Circuit Design for Minimum Dynamic Power and Highest Speed, In *Proc. Intl. Conf. on VLSI Design*, pp. 1035–1040, 2004.
- [10] T. Raja, V. D. Agrawal and M. L. Bushnell, Variable Input Delay CMOS Logic for Low Power Design, In *Proc. Intl. Conf. on VLSI Design*, pp. 596–604, 2005.
- [11] T. Raja, V. D. Agrawal and M. L. Bushnell, Transistor Sizing of Logic Gates to Maximize Input Delay Variability, *Journal of Low Power Electronics*, vol. 2, pp. 121–128, April 2006.
- [12] C. V. Schimpfle, A. Wroblewski and J. A. Nassek, Transistor Sizing for Switching Activity Reduction in Digital Circuits, In *Proc. the Euro. Conf. on Theory and Design*, 1999.
- [13] S. Uppalapati, M. L. Bushnell and V. D. Agrawal, Glitch-Free Design of Low Power ASICs using Customized Resistive Feedthrough Cells, In *Proc. VLSI Design and Test Symp.*, pp. 41–48, 2005.
- [14] A. Wróblewski, C.V. Schimpfle, O. Schumacher and J. A. Nassek, Minimizing Spurious Switching Activities with Transistor Sizing, *J. of VLSI Design*, vol. 15, pp. 537–546, no. 2, 2002.