Dynamic Power Estimation for Deep Submicron Circuits with Process Variation

Quang Dinh, Deming Chen, Martin D. F. Wong

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
e-mail: {qdinh2, dchen, mdfwong}@illinois.edu

Abstract - Dynamic power consumption in CMOS circuits is usually estimated based on the number of signal transitions. However, when considering glitches, this is not accurate because narrow glitches consume less power than wide glitches. Glitch width and transition density modeling is further complicated by the effect of process variation. This paper presents a fast and accurate dynamic power estimation method that considers the detailed effect of process variation. First, we extend the probabilistic modeling approach to handle timing variations. Then the power consumption of a logic gate is computed based on the transition waveforms of its inputs. Both mean values and standard deviations of the dynamic power are estimated with high confidence based on accurate device characterization data. Compared with SPICE-based Monte Carlo simulations for small circuits, our power estimator reports power results within 3% error for the mean and 5% error for the standard deviation with six orders of magnitude speedup. For medium and large benchmarks, it is impossible to run Monte Carlo simulations with enough samples due to very long runtime, while our estimator can finish within minutes.

I. INTRODUCTION

As process technology advancing towards the deep submicron scales, power consumption becomes a major issue. To be successful, the IC design flow needs to be aware of power consumption early on, at higher levels of the design hierarchy. This calls for fast and accurate power estimation methods. Another challenge arisen from aggressive technology scaling is process variation, or variations of device parameters in the fabricated ICs. This results in variations in timing delays and also in power consumption.

Power consumption in CMOS circuits is divided into *static power* and *dynamic power* consumptions. Static power is the consumed power when there is no signal activity in the circuit. Dynamic power is the consumed power associated with signal transitions. In this paper, we look at the problem of estimating dynamic power. Signal transitions are of two types, *functional transitions* and *spurious transitions*, or *glitches*. Functional transitions are those required to perform the logic functions from one clock cycle to the next. Glitches, on the other hand, are the unnecessary transitions caused by input signal transitions at different times. Accurate dynamic power estimations need to handle both types of transitions.

The dynamic power can be modeled by the following formula:

$$P_{dynamic} = \frac{1}{2} f_{clk} V_{dd}^{\ 2} \sum_{i=1}^{N} C_i S_i \qquad (1)$$

where N is the total number of gates, f_{clk} is the clock frequency, V_{dd} is the supply voltage, C_i is the load capacitance for gate i, and S_i is the switching activity for gate i. Switching activity is the average number of transitions (0 → 1 or 1 → 0) a signal switches per unit time. The switching activity includes both functional transitions and glitches.

This formula assumes that all the transitions are full-swing, or rail-to-rail. For narrow glitches, however, there is not enough time for complete swings. This partial-swing results in less power consumption comparing to the full-swing cases. The relationship between glitch power and glitch width is even more complicated under process variation, as the variation in delays results in variation in glitch widths. Fig. 1 shows an example of the dependence of dynamic power on glitch width. This is the result from gate characterization described later in section III.D.2, with the details of technology, process variation, and computing platform discussed in section IV. Note that the standard deviation can vary as much as five folds between partial-swing (glitch width of 50ps) and full-swing (glitch width of 100ps).

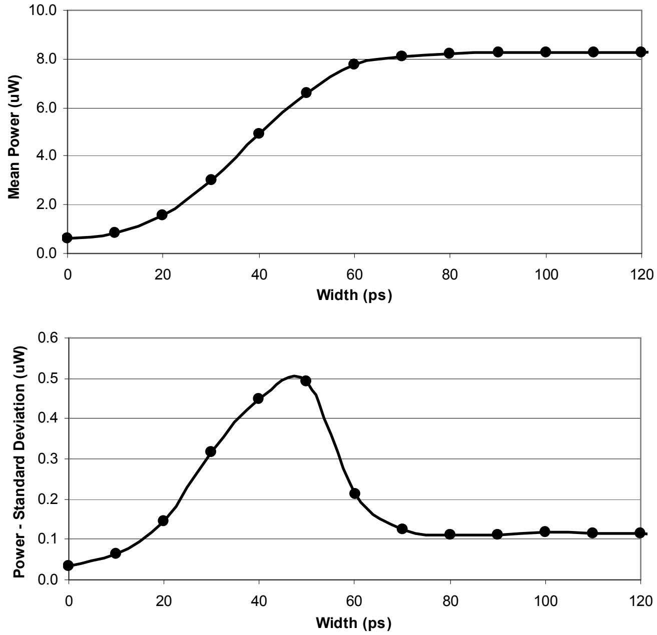

Fig. 1. Mean and standard deviation of the dynamic power of a logic gate as functions of input glitch width. Standard deviation can vary as much as five times.

In this paper, we propose a novel fast and accurate dynamic power estimation method, which can model multiple full- and partial-swings under process variation. We divide the signal waveform into multiple switching segments, based on the transition waveform presented in the tagged probabilistic simulation method in [1] and [2]. We then estimate power consumption by each switching segment. The major contributions of our paper are summarized below:

1. We introduce timing variation to the probabilistic transition waveforms.
2. We use device characterization to accurately capture dynamic power and its variation under full- and partial-swings by different glitch widths.
3. We develop formulas to estimate both the mean and the standard

deviation of dynamic power for a logic gate having multiple probabilistic transitions.

The rest of the paper is organized as follows. Section II discusses some related works. Section III presents the details of our power estimation method. The experimental results are shown in section IV. Finally we conclude the paper with section V.

II. Related Works

Many initial works on probability-based dynamic power estimation are based on the zero-delay model [3][4][5][6]. All transitions in the transient interval are ignored, only functional transitions are counted. Ignoring glitch power leads to underestimation of the power consumption. Real delay model is required to properly handle glitches. Works on dynamic power estimation under real delay model can be found in [1][2][7][8]. Theodoridis et al. extended the zero-delay techniques to handle real delay model [7]. Najm proposed a transition density approach, but did not model simultaneous signal transitions [8]. In the tag probabilistic simulation method by Tsui et al. [1][2], transition waveforms and probability waveforms were developed to model the statistical signal activity, including glitches, during different clock cycles. Although these works consider glitches, all transitions are assumed to be full-swing. Partial-swing effect, which is considered by our estimation method, is not accounted for. Dynamic power estimation under coupling effect are studied in [18]. All of these works assume fixed delay model, which is no longer true under process variation.

Process variation has been considered in many contexts, such as critical path delay analysis, logic verification, and delay testing. Circuit power under process variation has also been investigated, mainly focusing on leakage power [12][13][19]. A work on dynamic power estimation considering process variation was introduced by Sapatnekar et al. [11]. In this work, the clock cycle is divided into a number of time slots, and then a transition probability is computed for each slot. However, this formulation can only estimate mean dynamic power. It cannot be used to estimate variation of dynamic power, so no such variation result is reported. Furthermore, in their comparison study, they did not mention specifically which power simulator was used for verification. Another approach to dynamic power estimation under process variation was presented in [17], which requires simulation of input vectors. The variation model is very simple: only minimum and maximum bounds for delays are considered. Furthermore, partial swings are not considered.

For this work, we handle delay and timing variation by a statistical transition waveform model. This model allows both mean and variation of dynamic power to be estimated. Our method uses device characterization approach to accurately model power consumption of both full-swing and partial-swing glitches. This is similar to the power macro-modeling technique for high-level power estimation by Gupta et al. [9] and Barocci et al. [10]. A look-up table is created to store power estimates for different input statistics. In this paper, we use look-up table to store the mean and standard deviation of dynamic power for several different input glitch widths. This allows us to capture the variation of dynamic power of individual gates.

III. Power Estimation Method

A. Overview

To estimate dynamic power with partial-swing glitches and under process variation, we first use transition waveforms (discussed in section III.B) to capture the probability, timing and timing variation of possible transitions. We propagate the transition events through the circuit, taking into account the variation in gate and wire delays. Then we partition the transition waveforms into simple switching segments that form the basic units for our power estimation. From the width and variation of a switching segment, we formulate the way to estimate its dynamic power consumption. This is presented in section III.D.2. Finally, we show how to combine the estimation for individual switching segments to derive the dynamic power for a transition waveform in section III.D.3.

B. Transition Events and Transition Waveforms

B.1. Transition Events

As the power consumption of a gate greatly depends on the timing differences of the transitions at its inputs, we would need to keep track of these transitions with proper timing information. To consider process variation, the transition times are modeled as random variables in normal distribution, specified by mean values and standard deviations. We represent a single transition with a transition event.

A *transition event* at a particular node *n* in the circuit describes a possible transition at that node. We do not distinguish between 0→1 and 1→0 transitions, a transition event can be either of these two types. It is represented by three values (p, t, σ_t) as follows:

- p is the probability for the transition to occur.
- t is the mean of the time of the transition, with reference to the clock edge.
- σ_t is the standard deviation of the transition time.

For examples, a transition event (0.2, 0.42 ns, 0.21 ns) means that there is a likelihood of 20% that a transition occurs at a time around 0.42 ns, with a standard deviation of 0.21 ns.

B.2. Transition Waveforms

For each node in the circuit, there will be multiple possible transitions, caused by different transitions at the primary inputs propagating through different paths with different delays. The set of all transition events for a node are collectively represented by a *transition waveform*: $\{(p_1, t_1, \sigma_{t1}), (p_2, t_2, \sigma_{t2}),\ldots, (p_N, t_N, \sigma_{tN})\}$. This transition waveform has N transition events: $E_1, E_2,\ldots, E_N$. The events E_i of (p_i, t_i, σ_{ti}) are sorted according to the increasing order of t_i. Fig. 2 shows an example of a realization Boolean waveform for a transition waveform. In this example, events E_2, E_3 and E_N correspond to actual transitions, while events E_1 and E_{N-1} do not.

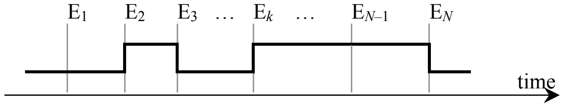

Fig. 2. A realization of a transition waveform $\{E_1, E_2,\ldots, E_N\}$ into a Boolean waveform.

B.3. Switching Segments

For a realization of a transition waveform, its dynamic power consumption can be estimated by partitioning the waveform into multiple switching segments. A *switching segment* is the duration from the beginning of a transition to just before the beginning of the next transition. A switching segment corresponds to exactly one transition. For example, in Fig. 2, there are four switching segments: (E_2, E_3), (E_3, E_k), (E_k, E_N), and the last switching segment starting at E_N.

In each switching segment, the signal starts with either a raising or a falling transition, thus consume dynamic power. Each switching segment has a determined duration, or width. It is the width of the switching segment that allows us to consider partial-swing glitches. Full-swing switches have wide widths, while partial-swing switches have narrow widths. In Fig. 3, for example, the segment (E_1, E_2) is a partial swing transition, and it has a narrow width (less than 60 ps, for example, for the gate shown in Fig. 1). The segment (E_6, E_7) is a full-swing transition, and it has a wide width (more than 60 ps). Based on the width of a switching segment, its dynamic power consumption can be estimated. Then we need to combine the dynamic power for all the switching segments.

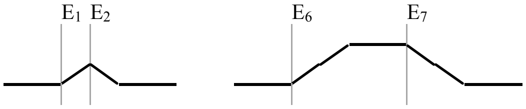

Fig. 3. Switching segment for partial-swing transition (E_1, E_2) and full-swing transition (E_6, E_7).

C. Propagation of Transition Waveforms

In a circuit, signal transitions originate at the primary inputs (and outputs of latches). Then these transitions propagate, with delay, through the paths in the circuits. At the logic gates, these transitions can either propagate or disappear. We would like to simulate these effects with the propagation of the transition waveforms. The transition waveforms will be calculated from the primary inputs to the primary outputs in topological order, by propagating through connecting wires and logic gates.

At each primary input, if no additional information is given, we can set the transition waveform to have only one transition event (there are no glitches), with a probability of the typical input switching activity. Mean time and standard deviation can be set according to the timing characteristics of latch elements.

C.1. Propagation through a Wire Segment

Each transition event E_S (p_S, t_S, σ_{tS}) in the transition waveform at the source node of the wire segment is propagated to a corresponding transition event E_T (p_T, t_T, σ_{tT}) in the transition waveform at the sink node. The transition probability is the same, and the timing information is propagated with the wire delay:

$$\begin{aligned} p_T &= p_S \\ t_T &= t_S + d \\ \sigma_{tT} &= \sqrt{\sigma_{tS}^{\ 2} + \sigma_d^{\ 2}} \end{aligned} \tag{2}$$

where d and σ_d are the mean and standard deviation of the wire segment, respectively.

C.2. Propagation through a Logic Gate

To propagate the transition waveforms through a logic gate, we first propagate each input to the output, and then we combine the transition events on the output to get the final transition waveform for the output.

A transition event E_I (p_I, t_I, σ_{tI}) at an input is propagated to a corresponding event E_O (p_O, t_O, σ_{tO}) at the output. Timing is propagated according to gate delay (d, σ_d), similar to wire delay:

$$\begin{aligned} t_O &= t_I + d \\ \sigma_{tO} &= \sqrt{\sigma_{tI}^{\ 2} + \sigma_d^{\ 2}} \end{aligned} \tag{3}$$

The transition probability is updated based on whether the transition from this input can propagate or not. This is computed based on the signal probability of the Boolean difference of the gate's logic function. For a logic gate with n inputs $x_1, x_2,\ldots, x_n$ and the output function is $f(x_1, x_2,\ldots, x_n)$, the Boolean difference of f with respect to an input x_i is defined as:

$$\frac{\partial f}{\partial x_i} = f_{x_i=0} \otimes f_{x_i=1} \tag{4}$$

where $f_{x_i=0}$ (or $f_{x_i=1}$) is the function f when $x_i = 0$ (or $x_i = 1$).

Signal probability is the ratio of the time the signal is in logic 1 to the total observation time. The Parker-McCluskey algorithm [16] can be used to compute signal probabilities. With the help of signal probability of the Boolean difference $P\left(\frac{\partial f}{\partial x_i}\right)$, the transition probability is propagated as follows:

$$p_O = p_I \times P\left(\frac{\partial f}{\partial x_i}\right) \tag{5}$$

After all transition events for all inputs are propagated to the output, we collect them into the transition waveform for the output. This transition waveform, together with the load capacitance of the gate, is used to estimate the dynamic power consumed by this gate. The detail of this estimation is presented in section III.D.

After power estimation, we need to process this waveform to carry out glitch filtering due to gate inertial delay. Glitches that are narrower than the inertial delay do consume some dynamic power, but do not actually appear at the output or propagate to the fan-outs. So we need to filter these glitches before we propagate the transition waveform to the fan-out nodes.

We look at all pairs of adjacent transition events. If two adjacent events E_A (p_A, t_A, σ_{tA}) and E_B (p_B, t_B, σ_{tB}) are closer than the inertial delay d: $t_B - t_A < d$, we combine these two events into one. There are four possible cases:

1. Neither E_A nor E_B transitions occur, resulting in no transition.
2. Both E_A and E_B transitions occur, they cancel each other out due to glitch filtering.
3. Only E_A occurs, with a probability of $p_{AnB} = p_A(1 - p_B)$.
4. Only E_B occurs, with a probability of $p_{BnA} = p_B(1 - p_A)$.

We combine the transitions in case 3 and 4 into one single transition event E_C. The new transition probability is:

$$p_C = p_{AnB} + p_{BnA} \tag{6}$$

The new mean of the transition time is the weighted sum:

$$t_C = \frac{p_{AnB}}{p_C} t_A + \frac{p_{BnA}}{p_C} t_B \tag{7}$$

And the new standard deviation is derived from the weighted sum of the second moments, similar to the explanation in section III.D.2:

$$\sigma_{tC} = \sqrt{\frac{p_{AnB}}{p_C}\left[\left(t_A - t_C\right)^2 + \sigma_{tA}^{\ 2}\right] + \frac{p_{BnA}}{p_C}\left[\left(t_B - t_C\right)^2 + \sigma_{tB}^{\ 2}\right]} \tag{8}$$

This combination step helps controlling the size of the transition waveforms as we propagate to the primary outputs. Since there will be no transitions that are closer than the inertial delay d, the number of transition events N in the transition waveforms of a gate's output is bounded by:

$$N \le \frac{t_{\max} - t_{\min}}{d} \tag{9}$$

where $t_{\max}$ is the latest arrival time, and $t_{\min}$ is the earliest arrival time of the gate's output. In our implementation, we limit the size of the transition waveforms further by keeping only the 20 most likely transition events.

D. Dynamic Power Estimation for a Gate from Its Transition Waveform

D.1. Overview

With the transition waveform available at the output of a logic gate, we can estimate the dynamic power consumption by this gate. If the transition waveform has N transition events, then there will be 2^N possible waveforms, many are complex with multiple transitions. An example is demonstrated in Fig. 4, with $N = 3$. Because we treat rising and falling edges in the same way, the illustration only shows waveforms starting at logic 0. Waveforms starting at logic 1 are considered to be the same. Clearly, the direct approach of enumerating all possible waveforms and estimating their power consumption is not feasible, due to the exponential complexity.

Our approach first considers each possible switching segment (a pair of transitions) independently, and then combines all possible switching segments together to derive the overall power. In Fig. 4, for example, there are three possible switching segments: the segment (E_1, E_2) which appears in waveform G and H, the segment (E_2, E_3) which appears in waveform D and H, and the segment (E_1, E_3) which appears in waveform F. The functional transitions at E_3 (in waveform B, D, F, H), at E_2 (in waveform C, G), and at E_1 (in waveform E) can also be considered as special switching segments. Summing over all possible switching segments for a transition waveform also gives us the total power consumption. This dividing method has a polynomial complexity of $O(N^2)$. The following section discusses the power consumption by a switching segment, and then the next section presents the combination of all possible segments.

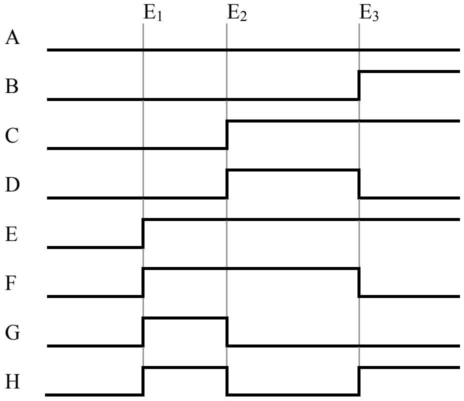

Fig. 4. 8 possible waveforms (A to H) from a transition waveform of 3 transition events {E_1, E_2, E_3}.

D.2. Dynamic Power by a Switching Segment

In this section, we investigate the problem of estimating dynamic power consumed by a gate from a single switching segment (two transitions (t_1, σ_{t1}) and (t_2, σ_{t2})) at the output. As mentioned before, this dynamic power *DPower*(w) depends on the glitch width w. Due to process variation, the glitch width w is a random variable. Even for a fixed *input glitch width* w_i (the time difference of the two transitions among the n inputs that cause the glitch at the output), the dynamic power *DPower*(w_i) itself is already a random variable, due to process variation. This variation is the result of variation in the load capacitance and variation in the gate delay from different inputs to the output. To achieve high quality estimation, we propose to use actual device characterization data instead of simply counting the number of transitions. This is to capture the true relationship between power consumption and glitch width.

Gate Characterization: In this paper, we use HSPICE and Monte Carlo simulations to collect characterization data. If actual devices are available, we can use physical power measurements on a large number of gates. We simulate 1000 gates with random process variation, excited by input vector pairs that cause a single glitch. The details of the variation are discussed in section IV. Then we collect and calculate the mean value and standard deviation of the dynamic power consumption. As the output waveform has two switching segments (for a single glitch), we assume that each segment consumes equal dynamic power. This means the dynamic power by one switching segment is half of the total dynamic power. We repeat this simulation for several different input glitch widths (20 different widths, from 0ps to 190ps) and several different load capacitances. From this characterization data, we can look-up and interpolate the mean $\mu_C(w_i)$ and standard deviation $\sigma_C(w_i)$ of dynamic power for a given input glitch width w_i and a given load capacitance.

It is not possible to directly use the characterization data as the power for the gate. This is because the input glitch width w is not fixed, but varies due to timing variation. For a switching segment by the two transitions (t_1, σ_{t1}) and (t_2, σ_{t2}), its width is a random variable of normal distribution with mean μ_w and standard deviation σ_w:

$$\mu_w = t_2 - t_1 \tag{10}$$

$$\sigma_w = \sqrt{\sigma_{t1}^2 + \sigma_{t2}^2} \tag{11}$$

The next problem is how to use μ_w, σ_w and the characterization data to derive the mean and standard deviation of the power consumption.

We discretize the width into several values w_0, w_1,…, w_L. The probability that $w = w_i$ is:

$$\Pr\left\{w = w_i\right\} = \frac{1}{\sigma_w\sqrt{2\pi}}\exp\left(-\frac{\left(w_i - \mu_w\right)^2}{2\sigma_w{}^2}\right) \tag{12}$$

according to the normal distribution formula. If we want to use a general distribution for timing variation, we can replace this formula with the discretized general distribution. We then need to normalize this probability as follows:

$$PN_i = \frac{\Pr\left\{w = w_i\right\}}{\sum_j \Pr\left\{w = w_j\right\}} \tag{13}$$

For each w_i, we can look-up and interpolate the corresponding mean power $\mu_C(w_i)$ and standard deviation $\sigma_C(w_i)$ from the characterization data. These can be interpreted as the statistical parameters of the dynamic power under the condition that $w = w_i$:

$$\mu_C\left(w_i\right) = \mathrm{E}\left[DPower\left(w\right)\middle| w = w_i\right] \tag{14}$$

$$\sigma_C\left(w_i\right) = \sqrt{\mathrm{E}\left[\left(DPower\left(w\right) - \mu_C\left(w_i\right)\right)^2\middle| w = w_i\right]} \tag{15}$$

Given PN_i and $\mu_C(w_i)$, the overall (unconditional) mean dynamic power μ_{DP} is simply the weighted sum:

$$\mu_{DP} = \mathrm{E}\left[DPower\left(w\right)\right] = \sum_i PN_i \times \mu_C\left(w_i\right) \tag{16}$$

The overall (unconditional) standard deviation σ_{DP} is, however, more complicated:

$$\sigma_{DP} = \sqrt{\mathrm{E}\left[\left(DPower\left(w\right) - \mu_{DP}\right)^2\right]} \tag{17}$$

For each w_i, $\mathrm{E}\left[\left(DPower\left(w_i\right) - \mu_{DP}\right)^2\right]$ is the statistical second moment of *DPower*(w_i) about μ_{DP}. Assume that the distribution of *DPower*(w_i) is normal, this moment can be derived as follows:

$$\mathrm{E}\left[\left(DPower\left(w_i\right) - \mu_{DP}\right)^2\right] = \left(\mu_C\left(w_i\right) - \mu_{DP}\right)^2 + \sigma_C\left(w_i\right)^2 \tag{18}$$

With this observation, we can compute the overall standard deviation σ_{DP} from the weighted sum of the second moments:

$$\sigma_{DP} = \sqrt{\sum_i PN_i \times \left[\left(\mu_C\left(w_i\right) - \mu_{DP}\right)^2 + \sigma_C\left(w_i\right)^2\right]} \tag{19}$$

To recap, the mean μ_{DP} and the standard deviation σ_{DP} of the dynamic power for a gate from a single switching segment at the output can be calculated from equation (16) and (19), respectively. The values $\mu_C(w_i)$ and $\sigma_C(w_i)$ are obtained from characterization data. PN_i are computed by following equations (10) to (13).

D.3. Dynamic Power by a Probabilistic Combination of Glitch Segments

After we can compute dynamic power for a single pair of transitions, we look at the problem of dynamic power consumed under many probable transitions at the output of the gate, as represented by the transition waveform. Our approach is to consider all possible pairs of transitions. For each pairs, we compute the dynamic power if that pair occurs at the output. We then compute the probability for this to happen. The overall power consumption by the transition waveform is then the weighted sum of the power by each pair.

For a gate with the transition waveform $\{(p_1, t_1, \sigma_{t1}), (p_2, t_2, \sigma_{t2}),\ldots, (p_N, t_N, \sigma_{tN})\}$, the mean and standard deviation of total dynamic power is computed, with a complexity of $O(N^2)$, as:

$$\mu_{DPgate} = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} Pr\left(i,j\right)\times \mu_{DP}\left(i,j\right) \qquad (20)$$

$$\sigma_{DPgate} = \sqrt{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} Pr\left(i,j\right)\times \sigma_{DP}\left(i,j\right)^2} \qquad (21)$$

In these equations, $\mu_{DP}(i,j)$ and $\sigma_{DP}(i,j)$ are the mean and standard deviation of dynamic power caused by the switching segment between transitions E_i and E_j, as presented in III.D.2. $Pr(i,j)$ is the probability that the switching segment (E_i, E_j) occurs. This switching segment occurs only if there are transitions at both E_i and E_j, and there are no transitions between E_i and E_j. The remaining transition events outside of [E_i, E_j] do not affect the switching segment. Fig. 5 illustrates this condition. Under the assumption that all transitions are independent random events, this conditional probability is computed as follows:

$$Pr\left(i,j\right) = p_i \times p_j \times \prod_{k=i+1}^{j-1}\left(1-p_k\right) \qquad (22)$$

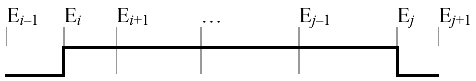

Fig. 5. Condition of transition events for the switching segment between E_i and E_j to occur.

To account for the power consumed by the final transition, which is a functional transition, we model this final transition as belonging to a switching segment, with the other end being a virtual transition at time $t = \infty$ and probability $p = 1$. This allows us to use the same set of equations (20) to (22) for both glitches and functional transitions. This is implemented by adding a virtual transition event E_∞ at the end of the transition waveform: ($p_{N+1} = 1$, $t_{N+1} = \infty$, $\sigma_{tN+1} = 0$). This virtual transition is illustrated in Fig. 6. Note that this virtual transition does not contribute any dynamic power. It is used only for the dynamic power formulas. For example, the functional transition at E_1 is handled as the switching segment (E_1, E_∞). This switching segment only occurs if there are no transition at E_2, which is equivalent to E_1 being the final, functional transition.

E. Dynamic Power Estimation for a Circuit

The total dynamic power consumption of a circuit is simply the sum of the power by its gates. Assume that the power for the gates are independent random variables of normal distribution, the total dynamic power is:

$$\mu_{DPcircuit} = \sum_{\text{all gates}} \mu_{DPgate} \qquad (23)$$

$$\sigma_{DPcircuit} = \sqrt{\sum_{\text{all gates}} {\sigma_{DPgate}}^2} \qquad (24)$$

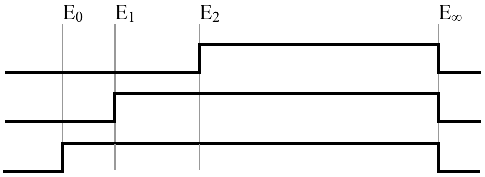

Fig. 6. Virtual transition event E_∞ to handle functional transitions (at E_0, E_1, or E_2) as switching segments.

IV. Experimental Results

To evaluate the accuracy of our dynamic power estimation model, we compare our estimated power with results from Monte Carlo simulation with HSPICE. We use a 45nm technology library provided in the design kit Nangate FreePDK45 Generic Open Cell Library [14]. We introduce 20% random variation of normal distribution to four process parameters of the SPICE transistor models: W_g, L_{eff}, T_{OX}, and *na*. Similar variation is added to the unit R and C for wires. This degree of variation is compatible with the prediction in [15]. To have enough confidence in the Monte Carlo simulation results, we use 5 small synthetic circuits, each with 10000 sample runs. Each sample run is driven for 1000 clock cycles with randomly generated input vectors, simulating the transitions of 999 random vector pairs. The input signal probabilities are 0.5, with switching probabilities of 0.75. The distributions of gate and wire delays are also obtained from Monte Carlo simulation. Circuit 4 and 5 are shown in Fig. 7. To evaluate the runtime, we run our estimator on the circuits from the ISCAS89 benchmark suites.

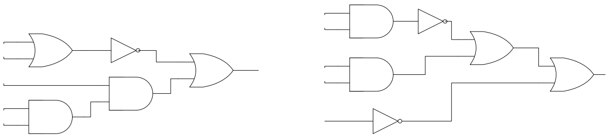

Fig. 7. Circuit 4 and Circuit 5 used in HSPICE-based Monte Carlo simulation comparison.

Assume that each gate has an average of 5 transistors, and each transistor has 4 random parameters, this sums up to 20 random variables per gate. For a small circuit of 5 gates, there are about 100 different random variables to be sampled by Monte Carlo simulation. To adequately sample this random space, we need to run 10000 samples. Similarly, 999 random input vector pairs are about just enough to provide unbiased statistics for 4 or 5 inputs. For larger circuits with more inputs, 999 random vector pairs are not enough to provide unbiased statistics.

The time to run SPICE simulation of Circuit 5 (with only 6 gates) for 10000 Monte Carlo samples, each with 1000 clock cycles, is already more than 400 days (it takes three weeks running simultaneously on 20 different CPUs). For a slightly larger circuit of 8 gates, it already takes 20 minutes to simulate 100 clock cycles for one sample circuit. We can linearly extrapolate that in order to simulate 10000 Monte Carlo samples for 1000 clock cycles (this would not be enough for unbiased random input switching), it would take almost four years. Clearly, it is impossible to run enough SPICE-based Monte Carlo simulation for any circuits of 8 gates or more.

Table 1 shows dynamic power comparison between Monte Carlo simulation and our estimation results. Our estimation can provide highly accurate estimates for all five circuits. The average of absolute estimation error is 3% for the means and 5% for the standard deviations. The maximum errors are 4.3% and 6.4%, respectively.

The difference in runtime to generate the results for Table 1 is reported in Table 2. Our estimation is more than six orders of magnitude faster than Monte Carlo simulation.

Finally, Table 3 shows our estimation results for some large circuits (from the ISCAS89 benchmark suites). Our estimation can compute within a few minutes. It is impossible to run Monte Carlo simulation for these circuits, due to the very large size of the SPICE netlist, and we would need to simulate these circuits with very long input vectors (as compared to 1000 input vectors for the small circuits) to have statistical confidence.

Table 1. Dynamic power estimation comparison with HSPICE-based Monte Carlo simulation

	Dynamic Power (μW)				Differences	
	HSPICE		Our Model		(%)	
	mean	*σ*	mean	*σ*	mean	*σ*
Circuit 1	2.649	0.2346	2.719	0.2479	–2.64	–5.67
Circuit 2	3.094	0.3816	2.961	0.3996	4.30	–4.72
Circuit 3	3.481	0.3865	3.514	0.4114	–0.95	–6.44
Circuit 4	3.186	0.3371	3.097	0.3539	2.79	–4.98
Circuit 5	4.491	0.4829	4.664	0.4970	–3.85	–2.92
Average of Absolute Differences					**2.91**	**4.95**

Table 2. Runtime comparison with HSPICE-based Monte Carlo simulation

	HSPICE (hours)	Our Model (seconds)	Speed-up $(\times 10^6)$
Circuit 1	1692	2	3.0
Circuit 2	1824	2	3.3
Circuit 3	4326	2	7.8
Circuit 4	6120	2	11.0
Circuit 5	9776	2	17.6
Average			8.5

Table 3. Estimation results for large circuits from the ISCAS89 benchmark suite

	Dynamic Power		Runtime
	Mean (mW)	*σ* (%)	(sec)
C1355	1.10	36.7	8
C1908	2.56	16.8	11
C2670	4.24	16.1	43
C3540	6.86	34.0	40
C432	0.85	20.5	4
C499	1.01	32.5	7
C5315	10.46	19.0	90
C6288	20.16	24.4	64
C7552	13.42	17.2	119
C880	1.02	27.4	8

V. Conclusions

In this paper, we present a novel dynamic power estimation method for circuits with process variation. Our estimation takes into account the difference between partial-swing and full-swing glitches, which is considerable under process variation. We first compute the transition waveform for each node in the circuit, including timing variation information. Then we partition the transition waveform into switching segments, the dynamic power by each of which can be estimated based on accurate gate characterization data. Then we show how to combine the individual switching segments together to estimate the total dynamic power by a transition waveform. We compare our estimation results with accurate Monte Carlo simulation based on HSPICE. We get high accuracy estimates, with average errors of 3% for the means and 5% for the standard deviations. Our estimation is also very fast, more than six orders of magnitude comparing to SPICE-Monte Carlo, and can estimate dynamic power for large circuits within minutes.

Acknowledgements

This work is partially supported by a grant from Altera Corp., by NFS grant CCF-0746608, and by NSF grant CCF-0701821.

References

[1] Tsui, C., Pedram, M., and Despain, A. M. Efficient estimation of dynamic power consumption under a real delay model. In *Proc. ICCAD*, Nov. 1993, 224-228.

[2] Ding, C-S, Tsui, C-Y, Pedram, M. Gate-level power estimation using tagged probabilistic simulation. *IEEE Trans. on CAD*, 17, 11, Nov 1998, 1099-1107.

[3] Tiwari, V., Ashar, P., Malik, S. Technology mapping for low power. In *Proc. DAC*, Jun 1993, 74-79.

[4] Roy, K., Prasad, S. SYCLOP: synthesis of CMOS logic for low power applications. In *Proc. ICCD*, Oct 1992, 464-467.

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

[6] Marculescu, R., Marculescu, D., and Pedram, M. Efficient power estimation for highly correlated input streams. In *Proc. DAC*, Jun 1995, 628-634.

[7] Theodoridis, G., Theoharis, S., Soudris, D., Stouraitis, T., Goutis, C. An efficient probabilistic method for logic circuits using real delay gate model. In *Proc. ISCAS*, Jul 1999, 286-289.

[8] Najm, F. N. Transition density: a new measure of activity in digital circuits. *IEEE Trans. on CAD*, 12, 2, Feb 1993, 310-323.

[9] Gupta, S. and Najm, F. N. Power macromodeling for high level power estimation. In *Proc. DAC*, Jun 1997, 365-370.

[10] Barocci, M., Benini, L., Bogliolo, A., Ricco, B. and De Micheli, G. Lookup table power macro-models for behavioral library components. In *Proc. VOLTA*, Mar 1999, 173.

[11] Pilli, S., Sapatnekar, S. S. Power estimation considering statistical IC parametric variations. In *Proc. ISCAS*, Jun 1997, 3, 1524-1527.

[12] Chang, H. and Sapatnekar, S. S. Full-chip analysis of leakage power under process variations, including spatial correlations. In *Proc. DAC*, Jun 2005, 523-528.

[13] Mukhopadhyay, S., Roy, K. Modeling and estimation of total leakage current in nano-scaled-CMOS devices considering the effect of parameter variation. In *Proc. ISLPED*, Aug 2003, 172-175.

[14] OSU design flows for MOSIS SCMOS_SUBM design flow FreePDK 45nm variation-aware design flow, http://vcag.ecen.okstate.edu/projects/scells/

[15] Nassif, S. Design for variability in DSM technologies. In *Proc. ISQED*, 2000, 451-454.

[16] Parker, K. P. and McCluskey, E. J. Probabilistic treatment of general combinational networks. In *IEEE Trans. on Computers*, 24, 6, Jun 1975, 668-670.

[17] Alexander, J. D. and Agrawal, V. D. Algorithms for estimating number of glitches and dynamic power in CMOS circuits with delay variations. In *Proc. ISVLSI*, May 2009, 127-132.

[18] Sinha, D., Khalil, D., Ismail, Y., and Zhou, H. A timing dependent power estimation framework considering coupling. In *Proc. ICCAD*, Nov 2006, 401-407.

[19] Li, T., Zhang, W., and Yu, Z. Full-chip leakage analysis in nano-scale technologies: mechanisms, variation sources, and verification. In *Proc. DAC*, Jun 2008, 594-599.