# On the Premises and Prospects of Timing Speculation

Rong Ye, Feng Yuan, Jie Zhang and Qiang Xu

CUhk REliable Computing Laboratory (CURE)
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
Email: {rye,fyuan,jzhang,qxu}@cse.cuhk.edu.hk

## ABSTRACT

*Timing speculation (TS), being able to detect and correct circuit timing errors at runtime, is a promising alternative solution to mitigate the ever-increasing variation effects in nanometer circuits. The potential energy-efficiency improvement, however, is limited by the circuit 'timing wall", a critical operating point caused by conventional circuit optimization techniques (e.g., gate sizing). With a given circuit netlist, we study the bound of the potential benefits provided by TS techniques in this work, which facilitate designers to decide whether it worths the effort to implement a timing-speculative circuit. Experimental results on benchmark circuits demonstrate the effectiveness of the proposed methodology.*

## 1. INTRODUCTION

Power minimization is a primary objective in the design of integrated circuits (ICs) nowadays, which is achieved with CMOS technology scaling and low-power design techniques at various levels. Prior works suggest that the best and probably the only effective way to achieve power reduction at design time is to eliminate waste whenever possible. In existing worst-case oriented designs, much energy is wasted to guarantee "*error-free*" computations. If we can over-clock the frequency and/or reduce the supply voltage of the circuit with correct computational results under nominal timing conditions while conducting online error detection and correction when timing errors occur under worst-case conditions, the potential circuit energy efficiency gain can be significant. Such *better-than-worst-case* (BTWC) design methodology [1] thus received lots of research attention, wherein the key enabling technique used to effectively tradeoff reliability with performance/power of the circuit to achieve "*error-resilient*" computations is called *timing speculation* (TS) [2–7].

For a well-tuned circuit, there usually exists a large number of speed-paths (i.e., critical or near-critical paths) after timing and power optimization, which manifest themselves as a wall in the timing slack histogram, referred to as "*timing wall*" [8]. Such phenomenon, however, limits the effectiveness of timing speculation due to the performance/energy penalties associated with timing error correction [9]. That is, when errors occur in a timing-speculative circuit, the system needs to be rolled back to a pre-error state for re-computation (usually with slower frequency), which incurs both performance penalty and extra energy consumption. Consequently, a timing-speculative circuit needs to operate at a voltage-frequency combination with small timing error rate (TER). The timing wall basically dictates the threshold beyond which there are massive amount of timing errors and the associated penalties would outweigh its benefits. To mitigate this issue, a number of optimization techniques for timing-speculative circuits were proposed to reshape the circuit path delay distribution for effective timing speculation [10–13].

Timing speculation [2] is *not* orthogonal to other circuit-level power optimization techniques. For example, given a circuit netlist, we could downsize those gates on non-critical paths for power reduction [14], which, however, increases the height of the timing wall with more speed-paths in the circuit. Alternatively, we could upsize those gates on frequently-sensitized speed-paths and turn them into non-critical paths for effective timing speculation. As both methods can reduce power consumption and their impacts are interrelated, an interesting question is that, *given a circuit netlist, whether the potential energy efficiency gains provided by timing speculation (over conventional circuit-level power optimization techniques) is significant enough to warrant the effort to make it timing-speculative?* To the best of our knowledge, this problem, although important and relevant, has not been explicitly investigated in the literature.

As it is not possible to derive an optimal timing-speculative circuit for evaluation purpose, in this work, we try to answer the above question by studying the premises and prospects of timing speculation instead. To be specific, we develop novel algorithms to obtain the minimum and maximum potential benefits achievable with TS techniques which facilitate designers to explore preferred power optimization techniques. Experimental results on various benchmark circuits demonstrate the efficacy of the proposed methodology.

The remainder of this paper is organized as follows. In Section 2, we present the preliminaries and motivation of this work. The general optimization problem for timing-speculative circuits is formulated in Section 3. The premise problem and prospect problem on the potential benefit of TS are then detailed in Section 4. Next, Section 5 presents our experimental results on various benchmark circuits. Finally, Section 6 concludes this paper.

## 2. RELATED WORK

Various optimization techniques for timing-speculative circuits have been presented in the literature for TER reduction under a specified voltage-frequency combination. EVAL [12] uses a so-called high-dimensional dynamic adaptation technique that trades error rate for processor frequency by tilting, shifting, or reshaping the path distributions of various functional units. Blueshift [4] identifies and optimizes the most frequently exercised critical paths by on-demand selective biasing and path constraint tuning. DynaTune [5] optimizes the most frequently-sensitized critical paths of the circuit by assigning low threshold voltage to those critical gates that are strongly related to the occurrence of timing errors. To mitigate the impact of timing wall, *Kahng et al.* [10] proposed a slack redistribution strategy to achieve a gradual delay distribution that is able to better serve TS techniques.

978-3-9815370-4-8/DATE15/©2015 EDAA

The above solutions conduct optimization for timing speculation with a fixed circuit netlist. Recently, several logic synthesis techniques were proposed [6, 11, 15]. By intentionally manipulating the circuit structure for timing speculation, the circuit energy efficiency can be further improved.

# 3. GENERAL PROBLEM FORMULATION

Before introducing how to obtain the minimum and maximum potential benefits of TS, let us first formulate the optimization problem for a given timing-speculative circuit.

**Problem:** *Given the netlist of a timing-speculative circuit, equipped with timing speculators such as Razor flip-flops [2], and a performance constraint $f_c$, determine the size $w_i$ and the threshold voltage $v_i$ of each gate $G_i$, and the supply voltage $v_{dd}$ and operational clock frequency $f$ of the entire chip, so that the energy consumption $E_{total}$ is minimized under performance constraint.*

As re-computation is needed when timing errors occur, the energy consumption of timing-speculative circuits is:

$$E_{total}(\vec{w}, \vec{v}, v_{dd}, f) = P \cdot \frac{1}{f} \cdot (1 + error \cdot penalty) \quad , \quad (1)$$

where $\vec{w}$ is the vector whose element represents the size of each gate, $\vec{v}$ is the vector whose element represents the threshold voltage of each gate, $P$ is the power function, *error* is the function of timing error probability, and *penalty* is the cost including both the cycles of wasted execution that must be discarded and the time spent on checkpointing and re-execution. Meanwhile, we need to ensure the performance constraint:

$$f_{eq} = \frac{f}{(1 + error \cdot penalty)} \geq f_c \quad , \quad (2)$$

where $f_{eq}$ is the equivalent clock frequency considering performance penalty of timing error correction.

Note that, without loss of generality, we only consider the optimizations on gate size, threshold voltage and supply voltage in this work. The optimization objective function (see Eq. 1) requires the models of power consumption and timing error probability that have been discussed in prior work [13]. The proposed methodology, however, is applicable for any optimizations that have a closed-form objective function of energy consumption.

# 4. PREMISES AND PROSPECTS OF TIMING SPECULATION

As it is impossible to obtain an optimal solution for the problem defined in Section 3, we, instead, propose to investigate the minimum and maximum potential energy benefits of TS techniques. The minimum potential benefit establishes the premise for timing speculation while the maximum potential benefit presents the prospects of timing speculation.

## 4.1 The Premises

The premise problem to calculate the minimum potential benefit can be tackled by conducting an effective optimization method onto the general formulation in Section 3. In this work, we develop a novel technique consisting of two stages to solve it. The first stage is based on gradient-descent method (GDM) [18] considering continuous solution space of parameter setting, while the second stage optimizes the discrete parameters with the help of steepest descent method (SDM) [16].

### 4.1.1 Exploring Continuous Space by GDM

GDM is a first-order optimization algorithm that utilizes the gradient vector $\nabla f(\vec{x})$ to determine the search direction for each iteration. The simplest and most famous GDM algorithm takes steps proportional to the negative/positive of the gradient (or, the approximate gradient) of the function at current iteration to minimize/maximize $f(\vec{x})$.

With minor modifications to the timing error probability model proposed in prior work [13], we can now compute the gradients of objective function (see Eq. 1) with respect to parameters. For the sake of clear presentation, we use $\vec{x}$ to represent all the parameters ($\vec{w}$, $\vec{v}$, $v_{dd}$ and $f$) without distinguishing them from each other. As we would like to minimize energy consumption, we use the negative of the computed gradients to update the parameters at each iteration as follows:

$$x_\ell^{new} = x_\ell + \eta \cdot (-\frac{\partial E_{total}(\vec{x})}{\partial x_\ell}),\ \forall x_\ell \in \vec{x} = (x_1, \cdots, x_n)\ , \quad (3)$$

where $\eta$ is the learning rate.

### 4.1.2 Exploring Discrete Space by SDM

The above GDM-based technique can effectively optimize timing-speculative circuits in continuous space. However, it is not practical to assume arbitrary continuous values are allowed for parameters such as gate size $\vec{w}$ and threshold voltage $\vec{v}$, since a look-up table gate model with only a few number of discrete parameter values is the standard in most industrial designs. The GDM-based technique is used to achieve the setting of supply voltage $v_{dd}$ and clock frequency $f$. Its output on $\vec{w}$ and $\vec{v}$ would be further optimized by a novel SDM-based technique discussed as follows.

Given a set of discrete values for gate size and threshold voltage, we first discretize the output of GDM-based technique to the closest value within the set, providing an initial solution in discrete space. Then, we formulate a discrete search problem and resort to heuristic search algorithm based on SDM.

The solution representation is naturally described using the vector of parameter setting $\vec{x} = (x_1, \cdots, x_n)$, wherein each element is either gate size or threshold voltage of a certain gate. As for the move, it is simply defined as the change of $x_i$ ($1 \leq i \leq m$). This definition of move guarantees the completeness of traversing the entire discrete solution space.

To evaluate solutions during search process, we use the objective function defined by Eq. 1. It is worth noting that the change of $x_i$ affects only a small part of the calculation of objective function. For example, when conducting gate sizing to a gate, only its preceding gates and itself are influenced. Therefore, we only have to update the calculation under influence, dramatically saving computational effort.

With the above definitions, this problem can naturally be solved by search algorithms (e.g., random search, simulated annealing) [16]. In this work, we resort to SDM, a discrete analogue of GDM, because it is typically able to converge in a few steps.

## 4.2 The Prospects

As we are to estimate the maximum potential benefit to find the prospect of timing speculation, we can simplify the original problem as long as the solution of the simplified problem will still be an upper bound of the original one.

### 4.2.1 Estimation Algorithm

The proposed algorithm to estimate the minimum energy consumption is described in Fig. 1. We start the estimation from the setting with the minimum power/energy, i.e., the setting with all the gate sizes set to the minimum allowed value and all the threshold voltage set to the maximum allowed value (see Line 1 $\sim$ 3). In such case, it is very likely to have rather large error probability that exceeds the error constraint specified by performance constraint in Eq. 2 , and

```
#   W_min, the minimum allowed gate width
#   V_max, the maximum allowed threshold voltage
#   S, sensitivity
1.  Initialize w_i = W_min and v_i = V_max, ∀i
2.  Compute power consumption P
3.  Initialize estimated energy consumption E = P/f
4.  Compute error constraint ec = (f/f_c − 1)/penalty
5.  Compute error probability e with current setting
6.  Compute error debt ed = e − ec
7.  FOR each gate G_i
8.      Compute G_i.S = min(−ΔE/Δerror)
9.      Record G_i.E = ΔE when achieving minimum S
10.     Record G_i.error = −Δerror when achieving minimum S
11. REPEAT for each iteration until ed ≤ 0
12.     FOR each gate G_i with G_i.visited == 0
13.         IF G_i.S < S
14.             S = G_i.S
15.             m = i
16.     ed− = G_m.error
17.     E+ = G_m.E
18.     Set G_m.visited = 1
```

**Figure 1: The proposed algorithm to estimate the minimum energy consumption.**

hence owe an "error debt" that is defined as the difference between the current error probability and the specified error constraint (see Line 4 $\sim$ 6). To ensure the error constraint is satisfied and the eventually-estimated TS benefit is an upper bound, we have to guarantee that such "error debt" is paid off in the most energy-efficient manner. That is to say, we want to reduce error probability until it reaches error constraint at the minimum expense of energy increase.

To achieve the above, we define a metric *sensitivity* to describe the ratio of energy increase over error reduction due to the change of a certain parameter:

$$S = -\frac{\Delta E}{\Delta error} = \frac{E - E_0}{error_0 - error} \quad , \tag{4}$$

where $E_0$ and $error_0$ are the energy consumption and error probability with initial setting, and $E$ and $error$ are the energy consumption and error probability after the parameter change. If we can ideally obtain the minimum sensitivity of each gate (see Line 7 $\sim$ 10) and then take action to the gates one by one in the sensitivity-ascending order until error debt is paid off (see Line 11 $\sim$ 18), it is definitely the most energy-efficient way and the estimated value would be guaranteed to be a lower bound of TS energy consumption. Note that, the methodology of obtaining the minimum sensitivity of each gate would be detailed in the following.

#### 4.2.2 Problem Simplification

One of the difficulties to efficiently obtain the minimum sensitivity of each gate is due to the complex impact of the parameter change. For example, the size of a gate would influence not only its own load capacitance (and hence energy consumption) but also that of its preceding gates. More importantly, assuming there are $k$ parameter choices for each gate, the size of solution space is $k^n$, exponentially increased with respect to gate number $n$. Obviously, such solution space is too large to efficiently explore. To tackle this problem, we propose to eliminate the influence between gates and simultaneously reduce solution space by simplifying the problem.

We simplify the calculation of energy consumption, on the basis of an observation that the gates impact each other's energy consumption through affecting the load capacitance. For instance, if we downsize a gate, all the load capacitances and hence the energy consumption of its preceding gates would be reduced. Therefore, when calculating the load capacitance of a gate, if we always use the minimum sizes of its succeeding gates no matter what sizes they actually have, the load capacitance would be affected by only the gate itself but not its succeeding gates any more. With such a model that intentionally underestimates load capacitances, the estimated value of energy consumption can be less than the actual one, ensuring that it is still a lower bound of TS energy consumption. Note that, using similar methodology we can also simplify the calculation of error probability.

With the above simplifications, the energy consumption and error probability of a gate are both influenced by the parameters of the gate itself. Consequently, the calculation of the minimum sensitivity becomes very easy, as we only need to consider the impact locally. Especially in the scenario of discrete space, as the size of solution space for each gate is only equal to its choice number $k$ (typically, $k$ can be about 100), we can even enumerate all the combinations of parameters to achieve its minimum sensitivity. It is worth noting that the size of solution space for the entire chip is now reduced to $k \times n$, only linearly increased with respect to gate number $n$.

## 5. EXPERIMENTAL RESULTS

### 5.1 Experimental Setup

We conduct experiments on several large ISCAS'89 and ITC'99 benchmarks. Particularly, these ITC'99 benchmarks are the subsets of processors. We synthesize these circuits and obtain timing information using Synopsys EDA tools. To take process variation effects into consideration, we perform Monte Carlo simulation to inject gate-level delay variation following Gaussian distribution with standard deviation equal to 8%. Random inputs are used in our experiments and each simulation is performed with 100,000 cycles. All the experiments are conducted on a *2.8GHz* PC with *4GB* RAM.

For comparison, we provide two baseline solutions: (i) conventional technique without TS [17] , denoted as $CT_{baseline}$; and (ii) TS technique with conventional optimization [2], denoted as $TS_{baseline}$. The proposed optimization technique in Section 4.1 is denoted as $TS_{opt}$ and the estimation algorithm in Section 4.2 is denoted as $TS_{bound}$. To equip some of the flip-flops as timing speculators, we can simply resort to a simple scheme that equips all the flip-flops whose timing slacks are less than 20% of operational clock period. The hardware cost to equip a timing speculator is assumed to be 10 gates. The *penalty* of error recovery in Eq. 1 is assumed to be 10 clock cycles according to [7].

### 5.2 Results and Discussion

We report the results on hardware cost and algorithm runtimes in Table 1. As can be seen, the average hardware costs for $TS_{baseline}$ and $TS_{opt}$ to equip timing speculators are 5.32% and 5.66%, respectively. The hardware cost for $TS_{opt}$ is a little higher than $TS_{baseline}$, but still within an acceptable range. The runtimes of $TS_{opt}$ and $TS_{bound}$ are all less than one hundred seconds, both listed in Table 1.

To demonstrate the effectiveness of $TS_{opt}$, in Table 2 we present the energy consumptions[1] that have considered both the penalties of error recovery and the costs of equipping timing speculators. When compared to $TS_{baseline}$, $TS_{opt}$ further reduces energy consumption by 0.306 on average. The improvement room between $TS_{opt}$ and $TS_{bound}$ is only 0.091, showing that the proposed optimization technique $TS_{opt}$ is rather close to the "optimal".

Finally, we take the energy cost of timing speculators into account to calculate the minimum benefit and maximum potential benefit. It can be found in Table 3 that, after considering such costs, the minimum benefit achieved by $TS_{opt}$ is

[1]For clear presentation, all the energy consumptions are normalized to that of the case without any conventional and TS optimizations.

| Bench. | TG# | TFF# | $TS_{baseline}$ | | $TS_{opt}$ | | $RT_{opt}$ (s) | $RT_{bound}$ (s) |
|---|---|---|---|---|---|---|---|---|
| | | | $TTS\#$ | Cost (%) | $TTS\#$ | Cost (%) | | |
| s1494 | 680 | 6 | 2 | 2.94 | 2 | 2.94 | 10.91 | 0.21 |
| s5378 | 3042 | 179 | 20 | 6.57 | 21 | 6.90 | 22.06 | 1.32 |
| s9234 | 5866 | 228 | 22 | 3.75 | 25 | 4.26 | 23.65 | 1.99 |
| s13207 | 8803 | 638 | 10 | 1.14 | 15 | 1.70 | 27.06 | 5.58 |
| s15850 | 10470 | 597 | 89 | 8.50 | 92 | 8.79 | 33.25 | 9.67 |
| s35932 | 18148 | 1728 | 144 | 7.93 | 151 | 8.32 | 39.25 | 15.42 |
| s38584 | 21021 | 1426 | 168 | 7.99 | 179 | 8.52 | 44.05 | 21.24 |
| s38417 | 24341 | 1564 | 91 | 3.74 | 115 | 4.72 | 48.22 | 18.38 |
| b20 | 20226 | 490 | 121 | 5.64 | 116 | 5.42 | 45.13 | 13.65 |
| b21 | 20571 | 490 | 119 | 5.47 | 125 | 5.73 | 52.56 | 15.16 |
| b22 | 29951 | 735 | 153 | 4.86 | 156 | 4.95 | 57.88 | 19.41 |
| AVERAGE | | | | 5.32 | | 5.66 | | |

TG#: total gate count; TFF#: total flip-flop count; TTS#: total timing speculator count;
Cost: hardware cost ratio for equipping timing speculators;
$RT_{opt}$: runtime of $TS_{opt}$; $RT_{bound}$: runtime of $TS_{bound}$.

**Table 1: Hardware cost and algorithm runtimes.**

| Bench. | $TS_{baseline}$ | $TS_{opt}$ | | $TS_{bound}$ | |
|---|---|---|---|---|---|
| | | $En.$ | $\Delta_1$ | $En.$ | $\Delta_2$ |
| s1494 | 0.708 | 0.468 | -0.240 | 0.420 | -0.048 |
| s5378 | 0.859 | 0.578 | -0.281 | 0.506 | -0.072 |
| s9234 | 0.822 | 0.495 | -0.327 | 0.375 | -0.120 |
| s13207 | 0.726 | 0.480 | -0.246 | 0.346 | -0.134 |
| s15850 | 0.871 | 0.533 | -0.338 | 0.445 | -0.088 |
| s35932 | 0.866 | 0.596 | -0.270 | 0.458 | -0.138 |
| s38584 | 0.862 | 0.505 | -0.357 | 0.405 | -0.100 |
| s38417 | 0.709 | 0.455 | -0.254 | 0.342 | -0.113 |
| b20 | 0.810 | 0.525 | -0.285 | 0.420 | -0.106 |
| b21 | 0.854 | 0.473 | -0.381 | 0.400 | -0.073 |
| b22 | 0.825 | 0.436 | -0.389 | 0.428 | -0.009 |
| AVERAGE | 0.810 | 0.504 | -0.306 | 0.413 | -0.091 |

$En.$: energy consumption;
$\Delta_1$: energy difference of $TS_{opt}$ over $TS_{baseline}$;
$\Delta_2$: energy difference of $TS_{bound}$ over $TS_{opt}$.

**Table 2: Energy consumptions of TS techniques with the cost of timing speculators included.**

about 0.058 on average, while the maximum potential benefit estimated by $TS_{bound}$ is about 0.149 on average. Assuming an amount of benefit $\eta$ ($\eta = 0.1$) is considered to deserve design efforts by IC designers, we can conclude that: (i) TS is preferred for the optimization of the benchmarks $s13207$, $s38584$, $s38417$, $b22$, since they are proved by $TS_{opt}$ to have a benefit more than 0.1; and (ii) conventional technique is preferred for the benchmarks $s5378$, $s15850$ and $s35932$, because their maximum potential benefits, indicated by $TS_{bound}$, are all less than 0.1. As for the other benchmarks, the proposed methodology, unfortunately, is not able to conclude the applicability of TS on it, if the criteria $\eta = 0.1$ is given. Such difference in TS benefit is due to the diverse delay distributions of circuit netlists, motivating this work to differentiate which circuits are suitable for TS.

## 6. CONCLUSION

Timing speculation is a promising solution to combat the ever-increasing variation effects, but how much benefits can be provided is strongly related to the circuit structure itself. Considering the non-trivial design effort to make a circuit timing-speculative, for a given circuit, it is essential to evaluate the potential benefits at early design stage. In this work, we propose novel algorithms to study the premise and prospects of timing speculation to tackle this problem. Experimental results based on various benchmarks demonstrate the effectiveness of the proposed methodology.

## 7. ACKNOWLEDGEMENT

This work was supported in part by the Hong Kong S.A.R. General Research Fund (GRF) under Grant 418111 and Grant 418112.

| Bench. | $CT_{baseline}$ | $TS_{opt}$ | | $TS_{bound}$ | |
|---|---|---|---|---|---|
| | | $En.$ | $MB$ | $En.$ | $MPB$ |
| s1494 | 0.522 | 0.468 | 0.053 | 0.420 | 0.102 |
| s5378 | 0.544 | 0.578 | -0.034 | 0.506 | **0.038** |
| s9234 | 0.505 | 0.495 | 0.010 | 0.375 | 0.130 |
| s13207 | 0.706 | 0.480 | **0.227** | 0.346 | 0.360 |
| s15850 | 0.506 | 0.533 | -0.027 | 0.445 | **0.061** |
| s35932 | 0.525 | 0.596 | -0.071 | 0.458 | **0.066** |
| s38584 | 0.616 | 0.505 | **0.111** | 0.405 | 0.211 |
| s38417 | 0.585 | 0.455 | **0.130** | 0.342 | 0.243 |
| b20 | 0.573 | 0.525 | 0.048 | 0.420 | 0.153 |
| b21 | 0.557 | 0.473 | 0.084 | 0.400 | 0.157 |
| b22 | 0.544 | 0.436 | **0.108** | 0.428 | 0.117 |
| AVERAGE | 0.562 | 0.504 | 0.058 | 0.413 | 0.149 |

$MB$: The minimum benefit of TS;
$MPB$: The maximum potential benefit of TS.

**Table 3: TS benefits in terms of energy consumption.**

## 8. REFERENCES

[1] T. Austin and V. Bertacco. Deployment of better than worst-case design: solutions and needs. In *Proc. ICCD*, pp. 550–555, 2005.
[2] D. Ernst, *et al.* Razor: a low-power pipeline based on circuit-level timing speculation. In *Proc. Micro*, pp. 7-18, 2003.
[3] K. Bowman, *et al.* Energy-efficient and metastability-immune timing-error detection and instruction-replay-based recovery circuits for dynamic-variation tolerance. In *Proc. ISSCC*, 2008.
[4] B. Greskamp, *et al.* Blueshift: Designing processors for timing speculation from the ground up. In *Proc. HPCA*, 2009
[5] L. Wan and D. Chen. Dynatune: Circuit-level optimization for timing speculation considering dynamic path behavior. In *Proc. ICCAD*, 2009.
[6] Y. Liu, *et al.* On Logic Synthesis for Timing Speculation. In *Proc. ICCAD*, pp. 591-596, 2012.
[7] M. de Kruijf, S. Nomura, and K. Sankaralingam. A unified model for timing speculation: Evaluating the impact of technology scaling, CMOS design style, and fault recovery mechanism. In *Proc. DSN*, pp. 487-496, 2010.
[8] X. Bai, *et al.* Uncertainty-Aware Circuit Optimization. In *Proc. DAC*, pp. 58-63, 2002.
[9] J. Patel. CMOS Process Variations: A Critical Operation Point Hypothesis. *Online Presentation*, 2008.
[10] A.B. Kahng, S. Kang, R. Kumar, and J. Sartori. Designing a processor from the ground up to allow voltage/reliability tradeoffs. In *Proc. HPCA*, pp.1-11, 2010.
[11] Y. Liu, F. Yuan and Q. Xu. Re-synthesis for cost-efficient circuit-level timing speculation. In *Proc. DAC*, pp. 158-163, 2011.
[12] S. Sarangi, B. Greskamp, A. Tiwari, and J. Torrellas. EVAL: Utilizing processors with variation-induced timing errors. In *Proc. Mirco*, pp. 423-434, 2008.
[13] R. Ye, F. Yuan, H. Zhou, and Q. Xu. Clock skew scheduling for timing speculation. In *Proc. DATE*, pp. 929-934, 2012.
[14] M. Borah, R.M. Owens, and M.J. Irwin. Transistor sizing for low power CMOS circuits. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.15, no.6, pp.665-671, Jun 1996.
[15] J. Cong and K. Minkovich, Logic synthesis for better than worst-case designs. In *Proc. VLSI-DAT*, 2009.
[16] M. C. Vanier and J. M. Bower. A Comparative Survey of Automated Parameter-Search Methods for Compartmental Neural Models. In *Journal of Computational Neuroscience* , pp. 149-171, 1999.
[17] J. Hu, *et al.* Sensitivity-guided metaheuristics for accurate descrete gate sizing.. In *Proc. ICCAD*, 2012.
[18] J. A. Snyman. Practical mathematical optimization: An introduction to basic optimization theory and classical and new gradient-based algorithms. *Springer*, 2005.