CHAPTER VIII

MICROARCHITECTURE-AWARE CIRCUIT PLACEMENT WITH STATISTICAL RETIMING

Process variations in digital circuits make circuit timing validation an extremely challenging task. Variations on several high-impact intra-die process parameters such as effective gate length, thin-oxide thickness, wire width/height, and so forth can easily invalidate the timing predictions made before fabrication [12, 80, 84, 77, 2]. Therefore, statistical timing analysis tools that model the gate and wire delay as probability distribution functions have become increasingly popular for tackling timing validation under these process variations [62, 37, 10, 20, 4, 90, 73, 125, 91, 71, 3, 23, 65]. However, most of the existing studies focus on combinational circuits or subcircuits (after flip-flop removal) and fail to address sequential circuit timing validation directly. Recent work on statistical timing analysis for sequential circuits [36] allows the user to model flip-flops (FFs) and use them to predict the timing information *after* retiming. This work achieves a significant performance improvement by exploiting retiming-aware timing slack.

As pointed out in Chapter 6, retiming is an important technique that allows circuits to meet the timing target. This is even more important for microarchitecture-aware physical planning when the system assumes that the target clock period can be met. With process variations, the deterministic retiming technique as discussed in the last two chapters must be modified. In this chapter, Statistical Retiming-based Timing Analysis (SRTA) for the timing verification and optimization of sequential circuits under process variations is developed. First, a Statistical Bellman-Ford (SBF) algorithm is proposed for computing the longest path length distribution for directed graphs with negative cycles. It is proved in Section 8.1 that a statistical extension of the original Bellman-Ford algorithm correctly computes the longest path length distribution for the true distribution but at a very slow rate. Next, it is shown that two straightforward extensions of the Bellman-Ford algorithm for statistical analysis cannot guarantee the correctness of the results. Then, a SBF algorithm is proposed that closely approximates and efficiently computes the statistical longest path length distribution if there exists no positive cycle or detects one if the circuit is likely to have a positive cycle. The SBF algorithm is integrated into SRTA. SBF checks for the feasibility of the target clock period distribution for retiming analysis. Finally, it is shown that the final critical path delay distribution after retiming is the statistical maximum among all primary outputs and all feedback vertices. Monte Carlo simulation is used to validate the accuracy of the SRTA algorithm.

The SRTA algorithm is integrated into a mincut-based global placer to optimize statistical longest paths in sequential circuits. The placer performs multi-level bipartitioning recursively until the desired number of partitions is obtained. Then, the SRTA algorithm for computing statistical critical paths that considers retiming is performed. The placer then tries to place these paths into a single partition.

The remainder of this chapter is organized as follows. Section 8.1 describes our statistical Bellman-Ford algorithm. Section 8.2 presents the statistical retiming-based algorithm and its application in mincut-based global placement. The experimental results are presented in Section 8.3. Section 8.4 concludes this chapter.

8.1 Statistical Longest Path Analysis

8.1.1 Statistical Bellman-Ford Algorithm

First, some quantities in probability theory that are required to develop algorithms are defined. For more precise definitions of these quantities, see [40, 11]. Then, a statistical version of the Bellman-Ford algorithm that correctly solves the statistical longest path problem defined below is introduced.

Let Ω be the set of outcomes of a fabrication process. A subset of Ω is called an event. Let $\mathbf{P}$ be a function that assigns a probability to each event. A random variable $\mathbf{X} : \Omega \to \mathbb{R}^*$ maps each outcome to a number in the extended real line $\mathbb{R}^* \triangleq [-\infty, \infty]$. The probability that a random variable $\mathbf{X}$ takes a value in a subset M of $\mathbb{R}^*$ is $\mathbf{P}[\varpi | \mathbf{X}(\varpi) \in M]$. Assume that the probability $\mathbf{P}$ determines the joint (and hence, the marginal and conditional) distribution of all random variables of interest.

Let $G = (V, E)$ be a directed graph with a source node s and a sink node t, and $w : E \times \Omega \to \mathbb{R}$ be an associated edge-length function, which is a random variable for each edge $(u, v) \in E$. Without loss of generality, it is assumed that there is no weight on the nodes (since the weights on nodes can be pushed to their fan-in edges). Let K denote the number of directed *simple* (i.e., no cycles) $s - t$ paths in G, and $l_i : \Omega \to \mathbb{R}$ denote the length of the i^{th} path, $i = 1, \ldots, K$. Also, let $G(\varpi)$ be the graph G with length $w(u, v)(\varpi)$ on edge $(u, v) \in E$. If $\mathbf{X}$ is the longest path of G, it is defined as follows: for each $\varpi \in \Omega$, $\mathbf{X}(\varpi) = \max\{l_1(\varpi), \ldots, l_K(\varpi)\}$, if there is no positive cycle in $G(\varpi)$, and $\mathbf{X}(\varpi) = \infty$, otherwise. The distribution of $\mathbf{X}$ is determined by the probability $\mathbf{P}$ as mentioned above. The *Statistical Longest Path Problem* is defined as that of finding the distribution of the longest $s - t$ path in $G = (V, E)$ with edge-length function $w : E \times \Omega \to \mathbb{R}$.

First, the Bellman-Ford (BF) algorithm is extended to obtain the outcome-by-outcome Statistical Bellman-Ford algorithm (oSBF). An illustration is shown in Figure 60. As opposed to updating a certain set of *constants* (such as arrival times $a[i]$) in the BF, at each step of the oSBF, updating certain random variables that are *functions* of Ω is required by updating their values for each outcome $\varpi \in \Omega$. As a result, the complexity of the oSBF is high when the number of possible outcomes is large. In fact, when the random variables are continuous, such as uniform random variables, Ω has uncountably many elements and the oSBF cannot be carried out in practice. However, its properties, which are proved next, are useful for the approximation algorithm that is presented in the following section. Note that the Monte Carlo simulation can be considered as an approximated version of the oSBF in which certain elements of Ω are sampled according to the probability $\mathbf{P}$. The following theorem proves the correctness of the oSBF algorithm.

Theorem 2 *If $\mathbf{P}[\varpi | G(\varpi)$ has a positive cycle$] = 0$, then $a[i]$ from the oSBF has the same distribution as that of the random variable representing the longest path from s to i, $i \in V$. Otherwise, the oSBF returns FALSE.*

Proof: For each outcome $\varpi \in \Omega$, the oSBF is equivalent to the BF, which correctly

```
oStatistical Bellman-Ford(G, w, s, t)

Initialization Step
for (each v ∈ V)
  a[v](ϖ) ← −∞, ∀ϖ ∈ Ω;
  a[s](ϖ) ← 0, ∀ϖ ∈ Ω;
Stop(ϖ) ← NO, ∀ϖ ∈ Ω;
g(ϖ) ← −1, ∀ϖ ∈ Ω;
iter ← 1;

Relaxation Step
while (Stop(ϖ) = NO for some ϖ ∈ Ω and
iter < |V|)
  iter ← iter + 1;
  Ω̃ ← {ϖ ∈ Ω|Stop(ϖ) = NO};
  Stop(ϖ) ← YES, ∀ϖ ∈ Ω̃;
  for (each v ∈ V)
     for (each ϖ ∈ Ω̃)
        for (each edge (u, v) ∈ E)
           if    a[v](ϖ) < a[u](ϖ) + w(u, v)(ϖ);
           then  a[v](ϖ) ← a[u](ϖ) + w(u, v)(ϖ);
                 Stop(ϖ) ← NO;


Positive Cycles Detection Step
for (each (u, v) ∈ E)
  for (each ϖ ∈ Ω)
     if (a[v](ϖ) < a[u](ϖ) + w(u, v)(ϖ))
        g(ϖ) ← +1;

Output Step
if (P[ϖ|g(ϖ) = +1] > 0)
  then return FALSE;
  else return TRUE and a[t];
```

Figure 60: A description of outcome-by-outcome Statistical Bellman-Ford (oSBF) algorithm

calculates the lengths of the longest $s-i$ paths $a[i](\varpi), i \in V$ or correctly identifies a positive cycle in $G(\varpi)$. Hence, when the oSBF terminates, the function (random variable) $a[i]$ and the longest $s-i$ path are equal on $\Omega_1 \triangleq \{\varpi \in \Omega | G(\varpi)$ has no positive cycles$\}$ and are different on $\Omega_2 \triangleq \{\varpi \in \Omega | G(\varpi)$ has a positive cycle$\}$. If $\mathbf{P}[\Omega_2] = 0$, then they are equal with probability one and hence have the same distribution. Otherwise, there is a high probability that there is a positive cycle, and therefore the oSBF returns FALSE. ∎

Backward edges and backward nodes are defined as follows:

Definition 1 *For a given order of nodes P, $(u,v) \in E$ is a forward edge if u precedes v in P, and a backward edge, otherwise. In the latter case, node u is called a backward node.*

Lemma 3 *If a node order $L = \{s, v_1, \ldots, v_n, t\}$ is used in the relaxation step of the oSBF, after j relaxation iterations, $a[i]$ from oSBF is the random variable representing the longest (possibly not simple) $s-i$ path in G that contains $j-1$ or fewer (possibly repeated) backward edges.*

Proof: Let B denote the set of all backward edges associated with the order L. Then the subgraph $G_B \triangleq (V, E \setminus B)$ is a directed acyclic graph (DAG). The update step for node i in the relaxation step is

$$a[i] := \max_{u \in FI(i)} \Big[a[u] + w(u,i) \Big], \tag{68}$$

where $FI(i) \triangleq \{u \in V | (u,i) \in E\}$ is the set of *fan-ins* of node i. At the first iteration of the relaxation step, when $a[i]$ is updated, $a[u] = -\infty$ for all *backward fan-ins* $u \in FI_b(i) \triangleq \{u \in FI(i) | (u,i) \in B\}$ because from Definition 1, nodes $u \in FI_b(i)$ come after node i in the order and hence have not been updated. Thus, at the first iteration of the relaxation step, it is sufficient to perform relaxation on G_B. Since G_B is a DAG, and L is a topological order of G_B, $a[i]$ represents the longest $s-i$ path that contains zero backward edges after the first relaxation step.

Now suppose that the result of Lemma 3 is true up to some $j \geq 1$ (Induction Hypothesis 1: IH 1). At iteration $j+1$ of the relaxation step, it can be proved by induction that after $a[i]$ is updated using (68), it represents the longest $s-i$ path containing, at most, j backward edges. Consider the update for node v_1. Since s is the only node that precedes v_1 in L, all other nodes $u \in FI(v_1)$ are all updated after v_1. By IH 1, $a[u]$ represents the longest $s-u$ path with, at most, $j-1$ backward edges for all $u \in FI(v_1)$. From (68), the updated $a[v_1]$ is the longest $s-v_1$ path with, at most, j backward edges because any $s-v_1$ path with $c > 0$ backward edges has the last edge being a backward edge, and removing such an edge results in a path with $c-1$ backward edges.

Suppose that $a[v_i], i = 1, \ldots, r$ are now the longest $s-v_i$ paths with, at most, j backward edges for some $r \geq 1$ (Induction Hypothesis 2: IH 2). Similarly, from (68), the updated $a[v_{r+1}]$ is the longest $s-v_{r+1}$ path with, at most, j backward edges because any $s-v_{r+1}$ path with $c > 0$ backward edges either has the last edge being a backward edge or or has the last edge being a forward edge. If the last edge is a backward edge, then removing such an edge results in a path with $c-1$ edges (this case corresponds to $u \in FI_b(v_{r+1})$, whose $a[u]$ are, from IH 1, the longest $s-u$ paths with, at most, $j-1$ backward edges), If the last edge is a forward edge, then removing such an edge results in a path with c backward edges (this case corresponds to $u \in FI_f(v_{r+1}) \triangleq \{u \in FI(v_{r+1}) | (u, v_{r+1}) \notin B\}$, whose $a[u]$ are, from IH 2, the longest $s-u$ path with, at most, j backward edges). This completes the proof. ■

The following theorem improves the bound on the number of iterations of the relaxation step when there is no positive cycle. oSBF automatically terminates once the number of relaxation iterations reaches this bound (by the condition $Stop(\varpi) =$ YES for all $\varpi \in \Omega$). However, this is not true when the distribution of each $a[i]$ is approximately updated. Hence, the bound from this theorem will be useful for our approximation algorithm.

Theorem 4 *If* $\mathbf{P}[\varpi | G(\varpi)$ *has a positive cycle*$] = 0$, *and a node order* $L = \{s, v_1, \ldots, v_n, t\}$ *is used in the relaxation step of the oSBF, then after* $k+1$ *iterations,* $a[i]$ *from oSBF has the same distribution as that of the random variable representing the longest path from* s *to* i, $i \in V$, *where* k *is the maximum number of connected backward nodes that can be in a simple* $s-t$ *path in* G.

Proof: Since G has no positive cycles with probability 1, the longest $s-t$ path is a simple path with probability 1. According to the definition of k, the longest path has, at most, k backward nodes and hence, at most, k backward edges. The theorem follows from Lemma 3. ■

The result from Theorem 4 can be applied to approximation methods that are similar to the oSBF, except at each iteration, $a[i], i \in V$ is *approximately* updated. More specifically, if $a[i]$ is a good approximation to the actual $a[i]$ obtained from the oSBF, then after $k+1$ iterations, $a[i]$, obtained from the approximation method, is also a good approximation of the actual longest $s-i$ path. Hence, when there is no positive cycle with probability 1, we can stop the approximation algorithm after $k+1$ iterations.

8.1.2 Limitation of the Bellman-Ford Extensions

To the best of our knowledge, all proposed analytical models for statistical timing analysis suffer from the error introduced by the maximum function. This is because the output of the maximum function results in a new form of distribution. Unlike the true distribution, the Bellman-Ford algorithm can return incorrect results because of the error from approximated distributions (such as the normal distribution approximation for the delay distribution). This is because the original Bellman-Ford algorithm is not designed to tolerate errors resulting from statistical computation.

More precisely, it is typically assumed that the joint distribution of arrival times is fully characterized by vectors of parameters $\theta_i, i \in V$, which belong to a certain set Θ, which is closed under addition[1]. For example, when each node is assumed to be independently normally distributed, a two-dimensional vector $[\mu_i, \sigma_i^2]' \in \mathbb{R} \times [0, \infty)$ can be used to describe the mean and variance of the arrival time of node i. Let $f_{\max} : \Theta \times \Theta \to \Theta$ denote the maximum function that approximates the distribution of the maximum function by the distribution characterized by a vector in Θ. In this subsection, two examples are used to demonstrate the drawback of simple extensions of the Bellman-Ford algorithms.

The first extension is to use the Bellman-Ford algorithm to report positive cycles, as in statistical timing analysis. An illustration of the oSBF algorithm is shown in Figure 61(a). In this case, it reports a positive cycle when the original graph has no positive cycle. At the first iteration, an input arrives at node A with value a. Flip-flop B, buffer C, and gate D have delay b, c, and d, respectively. The output value of node D is $D^{(1)} = f_{\max}(a, C^{(0)}) + d = a + d$, where $C^{(0)}$ denotes the vector describing the distribution of the arrival time of gate C at iteration 0. After the signal propagates through flip-flop B and gate C, the new value of node D becomes $D^{(2)} = f_{\max}(a, a + d + b + c) + d$. Let $\triangle^{(2)} = D^{(2)} - D^{(1)}$ denote the change in the value of node D. Now if $d+b+c$ is originally negative with probability 1, but its distribution is approximated by that of a negative-mean random variable with a small probability of being positive (for example, a normal random variable with mean equals

[1] A set A is closed under addition if for any $a, b \in A$, we have $a + b \in A$. This assumption leads to an efficient propagation procedure which, however, can be relaxed.

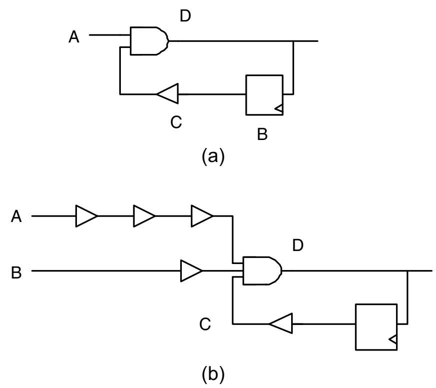

Figure 61: Illustration of Bellman-Ford update

-10 and variance equals 9), then it cannot be concluded that $\triangle^{(2)}$ will be a zero vector. Alternatively, even if the true distribution is initially used, the error from the maximum function could result in the same situation. After one more iteration of the Bellman-Ford, $D^{(3)} = f_{\max}(a, D^{(2)} + b + c) + d = D^{(2)} + \triangle^{(3)}$. A similar argument shows that $\triangle^{(3)}$ may not be a zero vector either. It can be concluded from related experiments that $\triangle^{(i)}, i = 1, 2, \ldots$ are small but might not become zero vectors after the $|V|$ iterations; consequently, the algorithm reports a positive cycle.

The second extension to the Bellman-Ford algorithm is to introduce the error bound on the longest path length distribution updates, which is referred to as the error-bounded Statistical Bellman-Ford (eSBF) algorithm.[2] Specifically, when the change in the distribution (for example the norm of $\triangle^{(i)}$) is less than such a bound, it is considered as no update. Although imposing a positive error bound δ can help the Bellman-Ford algorithm terminate when the graph has no positive cycle, it could cause the Bellman-Ford algorithm to stop too early, when δ is too large. Consequently, from Lemma 3, some paths are not considered if the algorithm stops before $k + 1$ iterations. Figure 61(b) shows an example in which the delay from path A is ignored as follows: if the change of the arrival time at D resulting from delay propagated through A, which is $\triangle^{(i+1)} = f_{\max}(f_{\max}(D^{(i)} + c, b), a) + d - D^{(i)}$, is considered to be small with respect to the error bound δ, and the arrival time of node A has no further update, the algorithm could terminate without updating D. Hence, the

[2]We use this algorithm in comparison with other Bellman-Ford extensions.

information from path A is not propagated to the calculation of some other arrival times. As a consequence, the distribution of some $s-t$ paths that contain path A is not accounted for. Depending on the circuit structure, the total error resulting from this early termination could result in large error in arrival times.

8.1.3 Statistical Bellman-Ford Algorithm

The last extension of the Bellman-Ford algorithm, referred to as the k-Statistical Bellman-Ford (kSBF) algorithm, is shown in Figure 62. This is an algorithm that closely approximates and efficiently computes the longest path length distribution of directed graphs with negative cycles. kSBF is used in the statistical retiming-based timing analysis (SRTA) introduced in the next section. First, a depth-first search algorithm is called to identify all backward edges and sort all nodes in a topological order. For each backward node, the depth-first search DFS' is called by setting this backward node as a source node. DFS' returns the maximum number of connected backward nodes reachable by a simple path from the given source. The maximum number of connected backward nodes of the graph (=k) is the largest number obtained by the DFS' algorithm. Note that this reachability algorithm needs to be performed only once. If all backward nodes are likely to be connected, the reachability step is not required, and instead, the total number of backward nodes can be used.

After the maximum number of connected backward nodes of the graph is found, the arrival times of all nodes are initialized. Next, the relaxation step is called. For the statistical longest path computation, $k + 1$ iterations are required. After the relaxation is done, all simple paths from source to sink are considered according to Theorem 4. Then, the statistical positive cycle detection algorithm proposed in [27] is used. If the probability of having no positive cycle is less than an acceptable probability, the algorithm returns FALSE, otherwise it returns TRUE.

```
Statistical Bellman-Ford(G, w, s, t, Pa)

Reachability Check Step
DFS(s); // find all backward edges
backward_node ← 1;
for (each v ∈ V)
  if (backward edge connected to v)
     backward_node ← backward_node + 1;
     list_backward_node ← v;
max_k ← 0;
for (each v ∈ list_backward_node)
  k ← DFS'(v); // backward nodes connected with v
  if (max_k < k)
     k ← max_k;

Initialization Step
for (each v ∈ V)
  a[v] ← −∞;
a[s] ← 0;

Relaxation Step
for (iter ← 1 to k + 1)
  for (each v ∈ V)
     a[v] ← max_{u∈FI(v)} [a[u] + w(u, v)];

Checking Positive Cycles Step
P(cycle) ← check_pos_cycle();

Output Step
if (P(cycle) ≤ Pa)
  then return FALSE;
  else return TRUE and a[t];
```

Figure 62: k-Statistical Bellman-Ford algorithm (kSBF) used in our statistical retiming-based timing analysis(SRTA)

8.2 Global Placement with Statistical Retiming

8.2.1 Modeling Delay Distribution

In this chapter, the delay distribution model is based on [125]. It is assumed that the delay of each gate and wire has a Gaussian distribution. The Elmore delay model is used for the wire delay computation based on the following equation (similar to [20]):

$$\begin{aligned} d_{int} = d_{int}^0 &+ \sum_{i\in\Gamma_g}[\frac{\partial d}{\partial L_g^i}]\triangle L_g^i + \sum_{i\in\Gamma_g}[\frac{\partial d}{\partial W_g^i}]\triangle W_g^i \\ &+ \sum_{i\in\Gamma_{int}}[\frac{\partial d}{\partial T_{int}^i}]\triangle T_{int}^i \end{aligned}$$

where d_{int}^0 is the expected value of wire delay. Γ_g and Γ_{int} are the set of grids where all the receivers reside and the interconnect tree traverses, respectively. $\triangle L_g^i$, $\triangle W_g^i$, and $\triangle T_{int}^i$

are random variables representing the variation over the expected value of transistor length, transistor width, and metal thickness, respectively (similar to [20]). The differentiations are derived based on the transistor and wire delay model from [95, 122, 103]. A 10% variation in each process parameter term is assumed in this study.

The principal component analysis technique (PCA), similar to [20], is used to derive the first-order form for arrival and required time delay distribution. The basic idea of PCA is to classify the input coefficients into orthogonal terms so that each coefficient term is uncorrelated. Reconvergent correlation can be efficiently handled by PCA. A grid hierarchical model is used for spatial correlation similar to [20]. If two gates are located near each other, they are more correlated than when they are far apart.

Four operations involved during statistical sequential arrival time and statistical required time computation are maximum, minimum, addition, and subtraction. The addition and subtraction of two Gaussian distributions result in another Gaussian distribution. The coefficient of each term in the first-order model can be added and subtracted directly. The maximum and minimum functions require the tightness probability calculation from [125], which is derived from [29, 16]. Based on this model and the assumption that the maximum and/or minimum of two Gaussian distributions result in a new Gaussian distribution, the coefficient results can be expressed as the summation of the product between the input distributions and the tightness probabilities.

8.2.2 Statistical Retiming based Timing Analysis

Similar to retiming-based timing analysis (RTA) [36], which is based on the Bellman-Ford longest path computation, statistical retiming-based timing analysis (SRTA) can be computed by using the k-statistical Bellman-Ford algorithm(kSBF) (see section 8.1). The statistical required time and the statistical arrival time are used to compute the statistical slack. The statistical slack is then used to identify the criticality of the cells and the nets. Unlike in the deterministic case, the statistical sequential required time cannot be computed at the same time as the statistical sequential arrival time because the distribution of the arrival time at the sink is not yet known until the statistical sequential arrival times of all

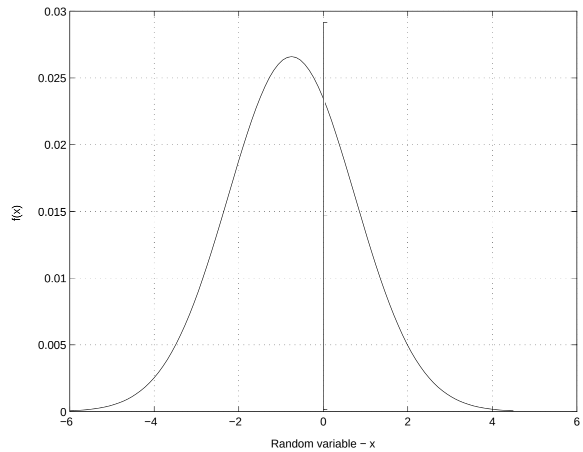

Figure 63: Positive cycle

nodes are computed. Note that the statistical sequential required time uses the minimum operation instead of the maximum operation.

Another difference from the kSBF algorithm is the early exit condition. In deterministic retiming-based timing analysis, the algorithm can perform early exit when the arrival time at the sink node exceeds the target clock period. In particular, if the graph has a positive cycle, performing the longest path computation over that cycle can cause the arrival time of the sink node to exceed the target clock period. This still holds in the statistical case, that is, if the expectation of the summation of the gate and wire delay over a cycle is positive, then the arrival time of the sink can exceed the target clock period. This condition can be used for the algorithm to terminate early. However, one has to be aware that if the expectation of the summation of the gate and wire delay over a cycle is negative, it does not mean that there is no positive cycle, as shown in Figure 63. A cycle could be negative in terms of the mean value, but there is a high probability that a positive cycle exists. Also note that a strict condition (mean plus three times standard deviation less than zero) should be used for verifying a positive mean in a statistical case. In other words, the expectation of the summation along a cycle should be less than some positive ϵ value to ensure that the early exit condition comes from positive mean cycles, not from an error from approximation.

8.2.3 Bounds on Target Clock Period

In this subsection, a theoretical result on the bounds of the target clock period, ϕ, which will be useful in the binary search procedure is provided. Recall that the target clock period is set to the smallest value for which the graph $G = (V, E)$ has no positive cycle, and the arrival time of the sink node, $a[t]$, is no larger than ϕ. Let the delay of the i^{th} directed simple $s - t$ path in G be represented by $l_i = \psi_i - \kappa_i \phi$, where ψ_i denotes the sum of the gate and wire delays along path i, and κ_i denotes the number of flip-flops in path i. Let C denote the number of directed cycles in G, and $\zeta_j = \xi_j - \sigma_j \phi$ denote the total delay of the j^{th} directed cycle, where ξ_j denotes the sum of the gate and wire delays along cycle j. σ_j denotes the number of flip-flops in cycle j. Now ϕ is the smallest number that satisfies

$$\phi \geq a[t] = \max_{i=1,\ldots,K} \{\psi_i - \kappa_i \phi\} \tag{69}$$

$$\zeta_j = \xi_j - \sigma_j \phi \leq 0 \qquad\qquad j = 1\ldots, C. \tag{70}$$

Equivalently, the target clock period is given by

$$\phi = \max \left[\max_{i=1\ldots,K} \left\{ \frac{\psi_i}{\kappa_i + 1} \right\}, \max_{j=1\ldots,C} \left\{ \frac{\xi_j}{\sigma_j} \right\} \right]. \tag{71}$$

Recall that each gate and wire delay is a random variable, and hence, ψ_i and ξ_j, which are the sums of the gate and wire delays, are also random variables. Let $\phi_d^l, \phi_d^m, \phi_d^u$ denote the values of ϕ obtained from (71) when all the gate and wire delays are replaced by their lower bounds (best case), means (average case), and upper bounds (worst case), respectively. It is obvious that ϕ, which is a random variable, is in $[\phi_d^l, \phi_d^u]$ with probability 1. Moreover, as will be shown in the theorem below, the mean of ϕ is bounded below by ϕ_d^m.

Theorem 5 *Let* $\mathbf{E}[\phi]$ *be the mean (expected value) of* ϕ*. Then,* $\phi_d^l \leq \phi \leq \phi_d^u$ *with probability 1, and* $\phi_d^m \leq \mathbf{E}[\phi]$.

Proof: Only the mean case requires a proof. Equation (71) implies

$$\phi \geq \frac{\psi_i}{\kappa_i + 1}, \quad i = 1, \ldots, K \qquad \phi \geq \frac{\xi_j}{\sigma_j}, \quad j = 1, \ldots, C$$

Since the ψ_i and ξ_j are the sum of the gate and wire delays, and κ_i and σ_j are constants, the expected values of the right-hand terms of the inequalities above can be obtained by replacing all the gate and wire delays by their means. As a result, taking the expectation on both sides of the inequalities shows that the mean of ϕ is greater than each right-hand term under a deterministic average case. This implies that the mean of ϕ is greater than the maximum of all such terms, which is ϕ_d^m. ■

Note that the bounds $\phi_d^l, \phi_d^m, \phi_d^u$ can be obtained by solving deterministic longest path problems.

8.2.4 Mincut-based Constructive Placement

The SRTA algorithm is integrated into a mincut-based global placer to optimize statistical longest paths in sequential circuits. The placer performs multi-level bipartitioning recursively until the desired number of partitions is obtained. Then, statistical retiming-based timing analysis is used to compute statistical critical paths that consider retiming. The placer then tries to place these paths into a single partition. Unlike the deterministic Bellman-Ford, which can stop before $|V| - 1$ iterations if there is no update in the graph, the kSBF algorithm requires $k+1$ iterations before it can terminate. In addition, the kSBF requires a maximum function computation and the arrival time distribution propagation along the circuit and hence it tends to be much slower compared with the deterministic approach. To alleviate this problem, tighter upper and lower bounds on the target clock period are used to accelerate the binary search process in Theorem 5. The feasible clock period of the circuit has to be between the mean of the deterministic RTA and the worst case of the deterministic RTA.

8.2.5 Retiming Delay Distribution

Unlike the statistical longest path, which can report the delay of the sink node as the output delay of the graph, calculating the retiming delay distribution is more complex. Note that the combinational delay distribution can be computed as the maximum distribution of all the primary output nodes. A heuristic algorithm to calculate the retiming delay distribution is proposed. Given a sequential circuit, the retiming delay distribution is a function of the longest path distribution of the graph and the maximum mean cycle. Given an acyclic graph, the retiming delay distribution is the delay distribution of the sink node. On the

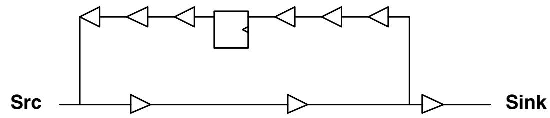

Figure 64: Impact of cycle on retiming delay distribution

other hand, if the graph has critical cycles with the cycle delay close to zero in the worst case, the maximum mean cycles can have an impact on the retiming delay distribution similar to the deterministic case [60].

However, identifying the statistical maximum mean cycle itself is not trivial and warrants further research. A heuristic to identify the distribution of the cycle is used. Note that if the cycle is critical, the worst case delay distribution of the gate and wire delays along that cycle can closely approach zero. The maximum mean cycle can be approximated by shifting the distribution of the cycle to the value of the target clock period. Consider the deterministic case example shown in Figure 64. Suppose the value of the gate delay is one, and the wire delay is zero. Based on retiming-based timing analysis, the delay of the sink node is three. However, the feasible clock period is eight because of the critical cycle. The retiming delay distribution can be approximated by taking the maximum function between the arrival time of the sink node and the distribution of the cycle. For example, if the cycle has a large negative delay value, (after shifting the cycle the delay distribution will be far less than the target clock period). Hence, this cycle will not have much impact during the computation of the retiming delay distribution. On the other hand, if this cycle is critical, its distribution will have larger impact during the computation of the retiming delay distribution.

8.3 Experimental Results

The SRTA algorithms are implemented in C++/STL, compiled with gcc v3.2.2, and run on a Pentium IV 2.4 GHz machine. The benchmark set shown in Table 17 consists of six big circuits from the ISCAS89 and five big circuits from the ITC99 suites. $k + 1$ is the number of iterations required by the k-statistical Bellman-Ford (kSBF) algorithm. BW represents the number of backward edges in the graph.

Table 17: Benchmark circuit characteristics

ckt	gate	PI	PO	FF	$k+1$	BW
s5378	2828	36	49	163	76	95
s9234	5597	36	39	211	239	354
s13207	8027	31	121	669	510	637
s15850	9786	14	87	597	495	699
s38417	22397	28	106	1636	1444	1660
s38584	19407	12	278	1452	1860	2054
b14o	5401	32	299	245	451	616
b15o	7092	37	519	449	988	1408
b20o	11979	32	512	490	1486	2197
b21o	12156	32	512	490	1511	2209
b22o	17351	32	725	703	1870	2770

Table 18 shows a comparison of the results obtained by using the Monte Carlo simulation, the modified Bellman-Ford algorithm with the error bound (eSBF), and the modified Bellman-Ford with $k+1$ iterations (kSBF) on 8x8 dimension. The Monte Carlo simulation is performed using 10,000 samples. The expectation (mean) and standard deviation (sigma) of the retiming delay distribution are reported. Note that both the eSBF and the kSBF provide results close to the Monte Carlo simulation results, especially in terms of the mean value. The kSBF provides more accurate results than the eSBF because, as pointed out in Section 8.1, the eSBF can ignore some paths during its computation. Note that it is possible that the eSBF outperforms the kSBF since the kSBF may require more iterations than necessary. Because of the error arising from the analytical model, the higher the number of iterations, the more errors can be accumulated. However, for most of the cases, the kSBF is more accurate that the eSBF. Both the eSBF and the kSBF substantially outperform the Monte Carlo simulation in terms of runtime. The runtime reported in all tables is the average runtime. The kSBF, however, requires longer runtime than the eSBF because it requires a higher number of iterations than the eSBF.

Figure 65 shows the comparison in terms of delay distribution among the Monte Carlo simulation, the eSBF, and the kSBF on s5378 benchmark. The solid, dotted, and dashed lines represent the Monte Carlo simulation, the eSBF, and the kSBF, respectively. Results show that the kSBF can provide a distribution similar to that of the Monte Carlo simulation.

Table 18: The comparison between the deterministic Monte Carlo simulation, the eSBF, and the kSBF on 8x8 dimension

ckt	monte		eSBF		kSBF	
	mean	std.dev.	mean	std.dev.	mean	std.dev.
s5378	179.65	6.27	163.29	5.49	179.47	6.51
s9234	229.24	16.18	164.58	7.14	225.531	10.5073
s13207	304.93	13.27	279.47	8.72	305.7588	13.0487
s15850	363.16	15.83	317.32	7.57	363.373	15.7213
s38417	187.11	4.53	184.95	5.28	189	8.4122
s38584	437.02	19.41	399.62	10.45	436.7471	19.3315
b14o	161.19	10.70	117.42	5.37	160.4306	5.3107
b15o	247.00	10.00	212.72	18.45	247.8771	10.2265
b20o	259.68	9.03	229.16	6.12	267.126	9.3349
b21o	267.57	6.18	240.01	4.08	267.984	9.2448
b22o	286.63	18.92	300.49	25.17	320.6154	15.2599
runtime	21 days		2.7 hours		3.7 hours	

In this case, it shows that the kSBF is better since the eSBF stops early and can ignore some paths.

8.4 Summary

In this chapter, a SBF algorithm is proposed to compute the longest path length distribution for directed graphs with cycles. The SBF algorithm is used in Statistical Retiming-based Timing Analysis (SRTA). It is used to check for the feasibility of the given target clock period distribution for retiming. The Monte Carlo simulation validates the accuracy of the SRTA algorithm. SRTA is used to guide a global placement with retiming to optimize statistical longest paths in sequential circuits.

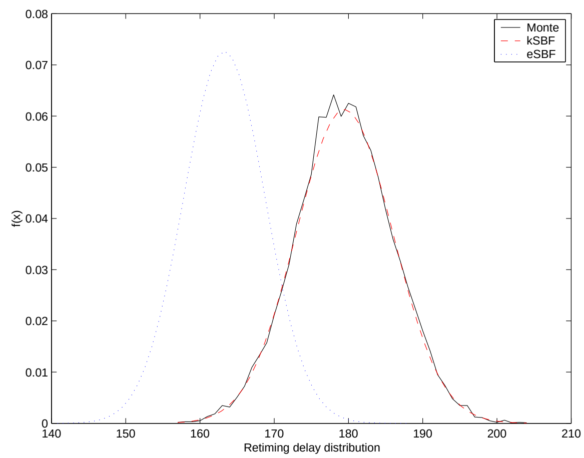

Figure 65: The Distribution Comparison among the Monte Carlo simulation (solid), the eSBF (dotted), the kSBF(dashed)

REFERENCES

[1] Ababei, C., Selvakkumaran, N., Bazargan, K., and Karypis, G., "Multi-objective circuit partitioning for cut size and path-based delay minimization," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 181–185, 2002.

[2] Acar, E., Nassif, S. R., Liu, Y., and Pileggi, L., "Assessment of true worst case circuit performance under interconnect parameter variations," in *International Symposium on Quality in Electronic Design*, pp. 431–436, 2001.

[3] Agarwal, A., Blaauw, D., and Zolotov, V., "Timing analysis for intra-die process variations with spatial correlations," in *Proceedings of the International Conference on Computer-Aided Design*, 2003.

[4] Agarwal, A., Zolotov, V., and Blaauw, D., "Statistical timing analysis using bounds and selective enumeration," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2003.

[5] Agarwal, V., Hrishikesh, M. S., Keckler, S. W., and Burger, D., "Clock Rate versus IPC: The End of the Road for Conventional Microarchitectures," in *ISCA*, 2000.

[6] Alpert, C. J., Chan, T. F., Kahng, A. B., Markov, I. L., and Mulet, P., "Faster minimization of linear wirelength for global placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1998.

[7] Augsburger, S. and Nikolic, B., "Reducing Power with Dual Supply, Dual Threshold and Transistor Sizing," in *Proc. IEEE Int. Conf. on Computer Design*, 2002.

[8] Austin, T. M., "Simplescalar tool suite." http:/www.simplescalar.com.

[9] Baer, J.-L. and Wang, W.-H., "On the inclusion properties for multi-level cache hierarchies," in *Proceedings of the 15th Annual International Symposium on Computer architecture*, pp. 73–80, IEEE Computer Society Press, 1988.

[10] Bhardwaj, S., Vrudhula, S., and Blaauw, D., "Tau: Timing analysis under uncertainty," in *Proceedings of the International Conference on Computer-Aided Design*, 2003.

[11] Billingsley, P., *Probability and Measure.* John Wiley & Sons, 1995.

[12] Bowman, K., Duvall, S., and Meindl, J., "Impact of die-to-die and within-die paraemter fluctuations on the maximum clock frequency distribution," in *Proc. ISSCC*, pp. 278–279, 2001.

[13] Breuer, M. A., "Class of Min-cut Placement Algorithms," in *DAC97*, pp. 284–290, 1997.

[14] Brodersen, R., Horowitz, M., Markovic, D., Nikolic, B., and Stojanovic, V., "Methods for True Power Minimization," in *Proceedings of the International Conference on Computer-Aided Design*, 2002.

[15] Brooks, D., Tiwari, V., and Martonosi, M., "Wattch: a framework for architectural-level power analysis and optimizations," in *Proceedings of the 27th annual international symposium on Computer architecture*, 2000.

[16] Cain, M., "The moment-generating function of the minimum of bivariate normal random variables," in *The American Statistican*, 1994.

[17] Caldwell, A. E., Kahng, A. B., and Markov, I. L., "Can recursive bisection alone produce routable placements?," in *Proceedings of the 37th Conference on Design Automation*, pp. 477–482, 2000.

[18] Chabini, N., Chabini, I., Aboulhamid, E. M., and Savaria, Y., " Unification of Basic Retiming and Supply Voltage Scaling to Minimize Dynamic Power Consumption for Synchronous Digital Designs ," in *Proc. Great Lakes Symposum on VLSI*, pp. 221–224, 2003.

[19] Chabini, N. and Wolf, W., "Reducing Dynamic Power Consumption in Synchronous Sequential Digital Designs Using Retiming and Supply Voltage Scaling," *IEEE Transactions on VLSI Systems*, vol. 12, pp. 573–589, June 2004.

[20] Chang, H. and Sapatnekar, S., "Statistical timing analysis considering spatial correlations using a single pert-like traversal," in *Proceedings of the International Conference on Computer-Aided Design*, 2003.

[21] Chang, J. and Pedram, M., "Energy minimization using multiple supply voltages," *IEEE Transactions on VLSI Systems*, 1997.

[22] Chao, K.-Y. and Wong, D. F., "Low power considerations in floorplan design," in *International Workshop on Logic Power Design*, 1994.

[23] Chao, M. C.-T., Wang., L.-C., Cheng, K.-T., and Kundu, S., "Static statistical timing analysis for latch-based pipeline designs," in *Proceedings of the International Conference on Computer-Aided Design*, 2004.

[24] Chen, C., Srivastava, A., and Sarrafzadeh, M., "On gate level power optimization using dual-supply voltages," *IEEE Transactions on VLSI Systems*, 2001.

[25] Chen, C., Srivastava, A., and Sarrafzadeh, M., " On Gate Level Power Optimization Using Dual-Supply Voltages ," *IEEE Transactions on VLSI Systems*, vol. 9, pp. 616–629, October 2001.

[26] Chen, G. and Sapatnekar, S., "Partition-driven standard cell thermal placement," in *Proc. Int. Symp. on Physical Design*, 2003.

[27] Chen, R. and Zhou, H., "Clock schedule verification under process variations," in *Proceedings of the International Conference on Computer-Aided Design*, 2004.

[28] Chishti, Z., Powell, M. D., and Vijaykumar, T. N., "Distance associativity for high-performance energy-efficient non-uniform cache architectures," in *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture*, p. 55, 2003.

[29] Clark, C., "The greatest of a finite set of random variables," in *Operations Research*, 1961.

[30] Cong, J., Fan, Y., Han, G., Yang, X., and Zhang, Z., "Architectural Synthesis Integrated with Global Placement for Multi-Cycle Communication," in *ICCAD*, 2003.

[31] Cong, J. and Koh, C.-K., "Simultaneouls driver and wire sizing for performance and power optimization," in *Proceedings of the International Conference on Computer-Aided Design*, 1994.

[32] Cong, J. and Lim, S. K., "Multiway partitioning wih pairwise movement," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 512–516, 1998.

[33] Cong, J. and Lim, S. K., "Physical planning with retiming," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 2–7, 2000.

[34] Cong, J. and Lim, S. K., "Edge separability based circuit clustering with application to circuit partitioning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 346–357, August 2003.

[35] Cong, J. and Yuan, X., "Multilevel global placement with retiming," in *Proceedings of the 40th Conference on Design Automation*, pp. 208–213, 2003.

[36] Cong, J. and Lim, S. K., "Retiming-based timing analysis with an application to mincut-based global placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 12, pp. 1684–1692, 2004.

[37] Devgan, A. and Kashyap, C., "Block-based static timing analysis with uncertainty," in *Proceedings of the International Conference on Computer-Aided Design*, 2003.

[38] Dhillon, Y. S., Diril, A. U., Chatterjee, A., and Lee, H.-H. S., " Algorithm for achieving minimum energy consumption in CMOS circuits using multiple supply and threshold voltages at the module level ," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 693–700, 2003.

[39] di Torino, P., "ITC benchmark." http://www.cad.polito.it/tools/itc99.html, 1999.

[40] Durrett, R., *Probability: Theory and Examples.* Duxbury Press, 1995.

[41] Eble, J. C., De, V. K., Wills, D. S., and Meindl, J. D., "A Generic System Simulator (GENESYS) for ASIC Technology and Architecture Beyond 2001," in *Int'l ASIC Conference*, 1996.

[42] Eisenmann, H. and Johannes, F. M., "Generic global placement and floorplanning," in *Proceedings of the 35th Conference on Design Automation*, pp. 269–274, 1998.

[43] Ekpanyapong, M., Minz, J., Watewai, T., Lee, H.-H. S., and Lim, S. K., "Profile-guided microarchitectural floorplanning for deep submicron processor design," in *Proceedings of the 41st Conference on Design Automation*, 2004.

[44] Faraboschi, P., Brown, G., Fisher, J. A., Desoli, G., and Homewood, F., "Lx: a technology platform for customizable vliw embedded processing," in *Proceedings of the 27th annual international symposium on Computer architecture*, pp. 203–213, 2000.

[45] Farhang-Boroujeny, B., *Adaptive Filters — Theory and Applications*. John Wiley and Sons, 1998.

[46] Fiduccia, C. and Mattheyses, R., "A linear time heuristic for improving network partitions," in *Proc. ACM Design Automation Conf.*, pp. 175–181, 1982.

[47] Ghose, K. and Kamble, M. B., "Reducing Power in Superscalar Processor Caches using Subbanking, Multiple Line Buffers and Bit-Line Segmentation," in *Proceedings of the 1999 International Symposium on Low Power Electronics and Design*, 1999.

[48] Gieseke, B. A., Allmon, R. L., Bailey, D. W., Benschneider, B. J., Britton, S. M., Clouser, J. D., III, H. R. F., Farrell, J. A., Gowan, M. K., Houghton, C. L., Keller, J. B., Lee, T. H., Leibholz, D., Lowell, S. C., Matson, M. D., Matthew, R. J., Peng, V., Quinn, M. D., Priore, D. A., Smith, M. J., , and Wilcox, K. E., "A 600 MHz Superscalar RISC Microprocessor with Out-Of-Order Execution," in *Digest of Technical Papers International Solid-State Circuits Conference*, pp. 176–177, Feb. 1997.

[49] Givargis, T., Vahid, F., and Henkel, J., "System-level exploration for pareto-optimal configurations in parameterized system-on-a-chip (december 2002)," *IEEE Transsactions on VLSI System*, vol. 10, no. 4, pp. 416–422, 2002.

[50] Glaskowsky, P. N., "Pentium 4 (Partially) Previewed," *Microprocessor Report*, vol. 14, pp. 11–13, August 2000.

[51] GLPK, "GLPK (GNU linear programming) kit."

[52] Gosti, W., Narayan, A., Brayton, R., and SangiovanniVincentelli, A., "Wireplanning in Logic Synthesis," in *ICCAD*, 1998.

[53] Hamada, M. and Ootaguro, Y., "Utilizing Surplus Timing for Power Reduction," in *Proc. IEEE Custom Integrated Circuits Conference*, 2001.

[54] Hinton, G., Sager, D., Upton, M., Boggs, D., Carmean, D., Kyker, A., and Roussel, P., "The Microarchitecture of the Pentium 4 Processor," *Intel Technology Journal*, Q1 2001.

[55] Hitchcock, R., "Timing verification and the timing analysis program," in *Proc. ACM Design Automation Conf.*, 1982.

[56] Ho, R., Mai, K. W., and Horowitz., M. A., "The future of wires," *Proceedings of IEEE*, 2001.

[57] Huang, D. and Kahng, A. B., "Partitioning-based Standard-cell Global Placement with an Exact Objective.," in *ISPD97*, pp. 18–25, 1997.

[58] Huh, J., Burger, D., and Keckler, S. W., "Exploring the design space of future cmps," Sept. 2001.

[59] Hung, W., Xie, Y., Vijaykrishnan, N., Kandemir, M., Irwin, M., and Tsai, Y., "Total Power Optimization through Simultaneously Multiple-VDD Multiple-VTH Assignment and Device Sizing with Stack Forcing," in *Proc. Int. Symp. on Low Power Electronics and Design*, 2004.

[60] Hurst, A. P., Chong, P., and Kuehlmann, A., "Physical placement driven by sequential timing analysis," in *Proceedings of the International Conference on Computer-Aided Design*, 2004.

[61] Ishihara, F., Sheikh, F., and Nikolic, B., "Level conversion for dual-supply systems," in *Proc. Int. Symp. on Low Power Electronics and Design*, 2003.

[62] Jyu, H.-F., "Statistical timing analysis of combinational logic," in *IEEE Transactions on VLSI Systems*, pp. 126–137, 1993.

[63] Karnik, T., Ye, Y., Tschanz, J., Wei, L., Burns, S., Govindarajulu, V., De, V., and Borkar, S., "Total Power Optimization by Simultaneous Dual-Vt Allocation and Device Sizing in High Performance Microprocessors," in *Proc. ACM Design Automation Conf.*, 2002.

[64] Karypis, G., Aggarwal, R., Kumar, V., and Shekhar, S., "Multilevel hypergraph partitioning: Application in vlsi domain," in *Proceedings of the 34th Conference on Design Automation*, pp. 526–529, 1997.

[65] Khandelwal, V., Davoodi, A., and Srivastava, A., "Efficient statistical timing analysis through error budgeting," in *Proceedings of the International Conference on Computer-Aided Design*, 2004.

[66] Kim, C., Burger, D., and Keckler, S. W., "An Adaptive, Non-uniform Cache Structure for Wire-delay Dominated On-chip Caches," in *Proceedings of the 10th Symposium on Architectural Support for Programming Languages and Operating Systems*, pp. 211–222, 2002.

[67] Kin, J., Gupta, M., and Mangione-Smith, W. H., "Filtering Memory References to Increase Energy Efficiency," *IEEE Transactions on Computers*, vol. Vol. 49, No. 1, January 2000.

[68] Kleinhans, J. M., Sigl, G., Johannes, F. M., and Antreich, K. J., "GORDIAN : VLSI placement by quadratic programming and slicing optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 356–365, 1991.

[69] Kong, T., "A novel net weighting algorithm for timing-driven placement," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 172–176, 2002.

[70] Kulkarni, S. and Sylvester, D., "New level converters and level converting logic circuits for multi-VDD low power design," 2003.

[71] L. Zhang, Y. H. and Charlie, C. C., "Statistical timing analysis in sequential circuit for on-chip global interconnect pipelining," in *Proc. ACM Design Automation Conf.*, 2004.

[72] Lalgudi, K. and Papaefthymiou, M., "Fixed-phase retiming for low power," in *Proc. Int. Symp. on Low Power Electronics and Design*, 1996.

[73] Le, J., Li, X., and Pileggi, L. T., "Stac: Statistical timing analysis with correlation," in *Proc. ACM Design Automation Conf.*, 2004.

[74] Lee, H.-H. S. and Tyson, G. S., "Region-based Caching: an Energy Efficient Memory Architecture for Embedded Processors," in *In Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, 2000.

[75] Leiserson, C. E. and Saxe, J. B., "Retiming synchronous circuitry," in *Algorithmica*, pp. 5–35, 1991.

[76] Liao, W. and He, L., "Full-Chip Interconnect Power Estimation and Simulation Considering Concurrent Repeater and Flip-flop Insertion," in *ICCAD*, 2003.

[77] Liu, Y., Pileggi, L. T., and Strojwas, A. J., "Model order reduction of rc(l) interconnect including variational analysis," in *Proceedings of the 36th Conference on Design Automation*, pp. 201–206, 1999.

[78] Long, C., Simonson, L., Liao, W., and He, L., "Floorplanning optimization with trajectory piecewise-linear model for pipelined interconnects," in *Proc. ACM Design Automation Conf.*, 2004.

[79] McFarling, S., "Combining branch predictors," in *Digital Western Research Lab Technical Report TN-36*, 1993.

[80] Mehrotra, V., Nassif, S., Boning, D., and Chung, J., "Modeling the effects of manufacturing variation on high-speed microprocessor interconnect performance," in *IEDM Tech. Dig.*, pp. 767–770, 1998.

[81] Monteiro, J., Devadas, S., and Ghosh, A., "Retiming sequential circuits for low power," in *Proceedings of the International Conference on Computer-Aided Design*, 1993.

[82] Murata, H., Fujiyoshi, K., Nakatake, S., and Kajitani, Y., "Rectangle-packing-based module placement," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 472–479, 1996.

[83] Nakatake, S., Fujiyoshi, K., Murata, H., and Kajitani, Y., "Module placement on BSG-structure and IC layout applications," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 484–493, 1996.

[84] Nassif, S. R., "Modeling and analysis of manufacturing variations," in *Proc. CICC*, pp. 223–228, 2001.

[85] Neumann, I. and Kunz, W., "Placement driven retiming with a coupled edge timing model," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 95–102, 2001.

[86] Neumann, I. and Kunz, W., "Tight coupling of timing-driven placement and retiming," in *Proc. IEEE Int. Conf. on Computer Architecture*, pp. 351–354, 2001.

[87] Nguyen, D., Davar, A., Orshansky, M., Chinnery, D., Thompson, B., and Keutzer, K., "Minimization of Dynamic and Static Power Through Joint Assignment of Threshold Voltages and Sizing Optimization," in *Proc. Int. Symp. on Low Power Electronics and Design*, 2003.

[88] Nose, K. and Sakurai, T., "Optimization of Vdd and Vth for low-power and high-speed applications," in *Proc. Asia and South Pacific Design Automation Conf.*, 2000.

[89] of Technology, E. U., "LP_solve." ftp:/ftp.es.ele.tue.nl/pub/lp_solve/.

[90] Okada, K., Yamaoka, K., and Onodera, H., "A statistical gate-delay model considering intra-gate variability," in *Proceedings of the International Conference on Computer-Aided Design*, 2003.

[91] Orshansky, M. and Bandyopadhyay, A., "Fast statistical timing analysis handling arbitrary delay correlations," in *Proc. ACM Design Automation Conf.*, 2004.

[92] Palesi, M. and Givargis, T., "Multi-objective design space exploration using genetic algorithms," in *Proceedings of the tenth International Symposium on Hardware/Softw are Codesign*, pp. 67–72, 2002.

[93] Pan, P., "Continuous retiming: Algorithms and application," in *Proc. IEEE Int. Conf. on Computer Design*, pp. 116–121, 1997.

[94] Pant, P., Roy, R., and Chatterjee, A., "Dual-Threshold Voltage Assignment with Transistor Sizing for Low Power CMOS Circuits," *IEEE Transactions on VLSI Systems*, 2001.

[95] Rabaey, J., Chandrakasan, A., and Nikolic, B., *Digital Integrated Circuits*. Prentice Hall Electronics, 2003.

[96] Rebaudengo, M. and Reorda, M., "Gallo: a genetic algorithm for floorplan area optimization," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 943–951, 1996.

[97] Roy, K., Wei, L., and Chen, Z., " Multiple-Vdd & multiple Vth CMOS (MVCMOS) for low power applications ," in *International Symposium on Circuits and Systems*, pp. 366–370, 1999.

[98] Sankaralingam, K., Singh, V. A., Keckler, S. W., and Buger, D., "Routed Inter-ALU Networks for ILP Scalability and Performance," in *ICCD*, 2003.

[99] Schreiber, R., Aditya, S., Mahlke, S., Kathail, V., Rau, B. R., Cronquist, D., and Sivaraman, M., "Pico-npa: High-level synthesis of nonprogrammable hardware accelerators," *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, vol. 31, no. 2, pp. 127–142, 2002.

[100] Sheikh, F., Kuehlmann, A., and Keutzer, K., " Minimum-power retiming for dual-supply CMOS circuits ," in *Proceedings of the 8th ACM/IEEE Workshop on Timing issues in the specification and synthesis of digital systems*, pp. 43–49, 2002.

[101] Shivakumar, P. and Jouppi, N. P., "CACTI 3.0: An Integrated Cache Timing, Power, and Area Model," Tech. Rep. 2001.2, HP Western Research Labs, 2001.

[102] SIA, "National Techonology Roadmap for Semiconductors," 2001.

[103] SIA, "National Techonology Roadmap for Semiconductors," 2003.

[104] Singh, D. P. and Brown, S. D., "Integrated retiming and placement for field programmable gate arrays," in *Proc. Int. Symp. on Field Programmable Gate Arrays*, pp. 67–76, 2002.

[105] Sirichotiyakul, S., Edwards, T., Oh, C., Zuo, J., Dharchoudhury, A., Panda, R., and Blaauw, D., "Stand-by Power Minimization through Simultaneous Threshold Voltage Selection and Circuit Sizing," in *Proc. ACM Design Automation Conf.*, 1999.

[106] Sohi, G. and Vajapeyam, S., "Instruction Issue Logic for High-Performance Interruptable Pipelined Processors," *Proceedings of the 14th Annual International Symposium on Computer Architecture*, 1987.

[107] Srivastava, A. and Sylvester, D., "Minimizing Total Power by Simultaneous Vdd/Vth Assignment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2004.

[108] Srivastava, A., Sylvester, D., and Blaauw, D., "Power minimization using simultaneous gate sizing, dual-Vdd and dual-Vth assignment," in *Proc. ACM Design Automation Conf.*, 2004.

[109] Stockmeyer, L., "Optimal orientations of cells in slicing floorplan designs," in *Proc. Information and Control*, pp. 91–101, 1983.

[110] Su, C.-L. and Despain, A. M., "Cache Design Trade-off for Power and Performance Optimization: A Case Study," in *Proceedings of the International Symposium on Low Power Design*, 1995.

[111] Sultania, A., Sylvester, D., and Sapatnekar, S., "Tradeoffs between Gate Oxide Leakage and Delay for Dual Tox Circuits," in *Proc. ACM Design Automation Conf.*, 2004.

[112] Sun, W. J. and Sechen, C., "Efficient and effective placement for very large circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 349–359, 1995.

[113] Sundararaja, V. and Parhi, K. K., " Synthesis of low power CMOS VLSI circuits using dual supply voltages ," in *Proc. ACM Design Automation Conf.*, pp. 72–75, 1999.

[114] Sutanthavibul, S., Shragowitz, E., and Rosen, J. B., "An analytical approach to floorplan design and optimization," in *Proceedings of the 27th Conference on Design Automation*, pp. 187–192, 1991.

[115] Swartz, W. and Sechen, C., "Timing driven placement for large standard cell circuits," in *Proceedings of the 32rd Conference on Design Automation*, pp. 211–215, 1995.

[116] Tien, T., Su, H. P., and Tsay, Y., "Integrating logic retiming and register placement," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 136–139, 1998.

[117] Tsai, C. and Kang, S., "Cell-level placement for improving substrate thermal distribution," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2000.

[118] University, N. C. S., "ISCAS benchmark." http://www.cbl.ncsu.edu, 1989.

[119] Usami, K. and Horowitz, M., " Clustered Voltage Scaling Technique for Low-Power Design ," in *Proc. Int. Symp. on Low Power Electronics and Design*, pp. 3–9, 1995.

[120] Usami, K., Igarashi, M., Ishikawa, T., Kanazawa, M., Takahashi, M., Hamada, M., Arakida, H., Terazawa, T., and Kuroda, T., "Design methodology of ultra low-power MPEG4 codec core exploiting voltage scaling techniques," in *Proc. ACM Design Automation Conf.*, 1998.

[121] Usami, K., Igarashi, M., Minami, F., Ishikawa, T., Kanzawa, M., Ichida, M., and Nogami, K., " Automated low-power technique exploiting multiple supply voltages applied to media processor ," *Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 463–472, 1998.

[122] Uyemura, J., *CMOS Logic Circuit Design.* Kluwer Academic Publishers, 2002.

[123] Vaishnav, H. and Pedram, M., "Pcube: a performance driven placment algorithm for low-power designs," in *Proceedings on European Design Automation Conference*, 1993.

[124] Vaishnav, H. and Pedram, M., "Efficient statistical timing analysis through error budgeting," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 799–812, 1999.

[125] Visweswariah, C., Ravindran, K., Kalafala, K., Walker, S., and Narayan, S., "First-order incremental block-based statistical timing analysis," in *Proc. ACM Design Automation Conf.*, 2004.

[126] Vygen, J., "Algorithms for large-scale flat placement," in *Proceedings of the 34th Conference on Design Automation*, pp. 746–751, 1997.

[127] Wang, Q. and Vrudhula, S., "Algorithms for minimizing standby power in deep submicron, dual-Vt CMOS circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2002.

[128] Weiss, D., Wuu, J., and Chin, V., "The on-chip 3-mb subarray-based third-level cache on an itanium microprocessor," vol. Vol. 37, No. 11, pp. 1523–1529, 2002.

[129] Wong, D. and Liu, C. L., "A new algorithm for floorplan design," in *Proceedings of the 23th Conference on Design Automation*, pp. 101–107, 1986.

[130] Wong, D. and Liu, C. L., "Floorplan design for rectangular and l-shaped modules," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 101–107, 1987.

[131] in *www.tensilica.com*.

[132] Yeh, C., Chang, M., Change, S., and Jone, W., " Gate-level design exploiting dual supply voltages for power-driven applications ," in *Proc. ACM Design Automation Conf.*, pp. 68–71, 1999.

[133] Young, F. Y., Chu, C. C. N., Luk, W. S., and Wong, Y. C., "Floorplan area minization using lagrangian relaxation," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 687–692, 2001.

[134] Zhang, R., Roy, K., Koh, C.-K., and Janes, D. B., "Stochastic interconnect modeling, power trends, and performance characterization of 3-dimensional circuits," *IEEE Trans. on Electron Devices*, vol. 4, 2001.