ILP-based Supply and Threshold Voltage Assignment For Total Power Minimization

Mongkol Ekpanyapong, Pinar Korkmaz, and Sung Kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of Technology
Email: limsk@ece.gatech.edu

Abstract

In this paper we present an ILP-based method to simultaneously assign supply and threshold voltages to individual gates for dynamic and leakage power minimization. In our three-step approach, low power min-flipflop (FF) retiming is first performed to reduce the clock period while taking the FF delay/power into consideration. Next, the subsequent voltage assignment formulated in ILP makes the best possible supply/threshold voltage assignment under the given clock period constraint set by the retiming. Finally, a post-process further refines the voltage assignment solution by exploiting the remaining timing slack in the circuit. Related experiments show that the min-FF retiming plus simultaneous Vdd/Vth assignment approach outperforms the existing max-FF retiming plus Vdd-only assignment approach.

I. Introduction

OVER the last decade, IC power management has moved from a third-order to a first-order concern for chip designers, especially those designing ASICs and SOCs for portable-system applications. The low power research community has been actively proposing a huge volume of solutions during the last decade. Among the most successful ones at the circuit-level are supply voltage (Vdd) scaling, threshold voltage (Vth) scaling, gate-oxide (Tox) scaling, gate-sizing, retiming, and any combination of these methods. A majority of the existing works can be categorized into (i) Vdd scaling [1], [2], [3], [4], (ii) Vth scaling [5], (iii) simultaneous Vdd/Vth scaling [6], [7], [8], [9], (iv) Vth scaling and sizing [10], [11], [12], [13], (v) simultaneous Vdd/Vth scaling and gate sizing [14], [15], [16], [17], [18], (vi) simultaneous Vth/Tox scaling and state assignment [19], (vii) retiming [20], [21], and (viii) Vdd scaling and retiming [22], [23], [24]. In addition, various level converter design and usage are studied to support the low-Vdd to high-Vdd conversion in Vdd scaling method [25], [26].

We present the first work that performs retiming and simultaneous supply/threshold voltage assignment for total power reduction. The advantage of simultaneous Vdd/Vth assignment over Vdd-only has been demonstrated in [9]. On the other hand, retiming [27] is used to reduce dynamic power, where flip-flops (FFs) are repositioned to stop logic glitches from being propagated [21]. In addition, FFs can be used to enable low-to-high supply voltage transition, thereby reducing the need for separate level converters. The state-of-the-art in combining retiming and voltage assignment (Vdd-only) is by Chabini and Wolf [24], where they proposed a two-step approach that performs retiming and Vdd assignment sequentially.[1] We improve this work in the following ways:

- The authors [24] performed max-area retiming to increase the number of gates off timing critical path, which are ideal candidates for voltage assignment. We show that this approach in fact increases the FF count and thus the total power consumed by the FFs. Thus, we suggest min-FF retiming as a better choice.
- The authors [24] formulated the supply voltage assignment problem using integer linear programming (ILP) approach. Our simultaneous supply and threshold voltage assignment problem is also formulated as ILP, but we employ various LP-relaxation techniques to reduce the overall runtime by a few orders of magnitude.
- We show that min-FF retiming, while it reduces the critical path delay as well as total power consumed by the FFs, may reduce the total slack in the circuit and thus limit the subsequent voltage assignment. However, we show that the impact of the min-FF retiming on timing slack is minimal.
- Related experiments show that the min-FF retiming plus simultaneous Vdd/Vth assignment approach outperforms the existing max-FF retiming plus Vdd-only assignment approach [24] in terms of total power reduction.

We employ a three-step approach: retiming, voltage assignment, and post refinement step. A low power retiming is first performed to reduce the clock period while taking the FF delay/power into consideration. Next, the subsequent voltage assignment makes the best possible supply/threshold voltage assignment while satisfying the timing constraints set by the prior retiming step. We formulate the voltage assignment in ILP, relax it to LP, solve the LP in an iterative fashion, and apply various heuristics to convert the continuous LP solutions to integer solutions. Finally, a post refinement step further refines the voltage assignment solution by exploiting the remaining timing slack in the circuit. Related experiments show that our LP-based method named RVA (Retiming-based Voltage Assignment) algorithm provides results that are very close to the

[1] An ILP-based *simultaneous* retiming and supply voltage assignment has been attempted [23], where retiming as well as Vdd assignment are formulated as a single ILP, but the runtime was prohibitive even for very small circuits. Thus, the follow-up work employed a two-step approach [24].

original ILP formulation but at a fraction of runtime. In addition, the solution quality remains almost the same before and after the continuous-to-integer conversion.

The remainder of this paper is organized is as follows. Section II discusses issues related to gate-level voltage assignment. Section III presents low power retiming. Section IV presents our voltage assignment method. The experimental results are shown in Section V. We conclude the paper in Section VI.

II. Implementation Issues

Multiple supply voltage states can be achieved by running multiple power supply lines into the circuit. However, multiple supply lines can make routing process complicated [28]. Thus, there exists a tradeoff between lower energy dissipation and higher routing cost. To reduce the complexity of routing when multiple supply voltages are used in physical layout, gates with the same supply voltage are placed in a cluster [1]. This is especially true for a standard-cell design since the gates in a standard-cell design are arranged in rows, and their power lines are connected directly. However, this clustered-level supply voltage assignment is usually restrictive and generates inferior results compared to individual gate-level assignment.

Several on-chip voltage regulation techniques are proposed [29], [30], [31] to overcome the routing problem, which locally generates the low voltage power supply rails from the given higher voltage power supply rails without requiring any external components. Note that the power reduction by using this on-chip voltage regulator is not as effective as running multiple supply lines because of the power loss in the DC series path of the voltage regulator. The combination of both techniques can be used for power minimization with routing resources as the constraints. Recently, several circuit techniques are proposed [32], [33] to eliminate the need for additional level shifter used in dual supply CMOS circuit design. In [32], the authors use a second threshold voltage in the PMOS transistors of the high voltage gates driven by low voltage gates, thereby providing them with built-in level-shifting capability. These modified gates have no energy or area penalties and only a slight delay penalty over the regular high voltage gates. The idea in [33] is that if the voltage difference between driver and load is less than some specific value, there is no need for level converter insertion.

Four well known techniques for multiple threshold voltage scaling [34] are ion implantation, oxide thickness (T_{ox}) scaling, channel length (L_c) scaling, and changing body or back gate voltage. Ion implantation is achieved by using extra mask technique. The T_{ox}/L_c scaling technique assigns unique oxide thickness and channel length to each transistor. Changing body or back gate voltage is used to subsequently modify the threshold voltage for bulk silicon devices. This technique allows threshold voltage to be changed after fabrication and is also called adaptive body bias technique. Note that the ion implantation and T_{ox}/L_c scaling methods suffer from process variations in deep submicron technologies. Note that the masks for an additional higher Vth are expensive, which makes use of LCs preferable to using high Vth/high Vdd gates. This motivates why our work focuses on using LCs and LCFFs.

III. Low Power Retiming

A. Preliminaries

The synchronous sequential circuit is modeled with a directed graph $G = (V, E, d, w)$, where V is the set of gates, and E is the set of directed edges connecting gates. Edge $e_{i,j}$ represents a connection from gate i to gate j. $d(i)$ is the delay of gate i and $w(e(i,j))$ is the number of FFs on edge $e_{i,j}$.[2] Let $P(i,j)$ denote a directed path from gate i to gate j, and $w(P(i,j)) = \sum_{e \in P} w(e(i,j))$ denotes the total weight of the edges along $P(i,j)$. Let $d(P(i,j))$ denote the total delay of the nodes along $P(i,j)$. The original retiming paper [27] introduces the following two matrices: (i) $W(u,v)$ denotes $\min\{w(P(u,v)) | \forall u, v \in V\}$, which is the minimum weight value among all paths that connect u and v, and (ii) $D(u,v)$ denotes $\max\{d(P(u,v)) | w(P(u,v)) = W(u,v), \forall u, v \in V\}$, which is the maximum delay value among all paths with total weight of $W(u,v)$.

Let T be a target clock period.[3] Let $r(u)$ represent the number of FFs moved from all fan-out edges of node u to all fan-in edges of u. Retiming assigns an integer $r(u)$ to each node $u \in V$ such that the following constraints are met: (i) $r(u) - r(v) \leq w(e_{u,v}), \forall e_{u,v} \in E$, (ii) $r(u) - r(v) \leq W(u,v) - 1, \forall (u,v) \in V$ such that $D(u,v) > T$, (iii) the clock period after the retiming is equal to or less than T.

B. ILP based Formulation

Let $FI(v)$ and $FO(v)$ be the number of fan-in and fan-out of node v. An ILP-based low power retiming is formulated as follows:

$$\text{Minimize} \sum_{v \in V} (\{FI(v) - FO(v)\} \cdot r(v)),\ \forall v \in V \tag{1}$$

[2] Consideration of interconnect delay and power is discussed in Section IV-E.

[3] We perform binary search to find the minimum clock period. The retiming algorithm is used to check the feasibility of a given clock period in this case.

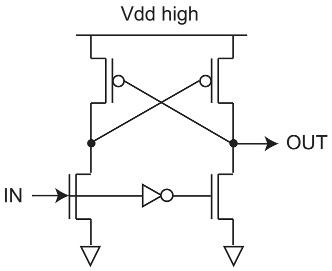

Fig. 1. Illustration of level converter

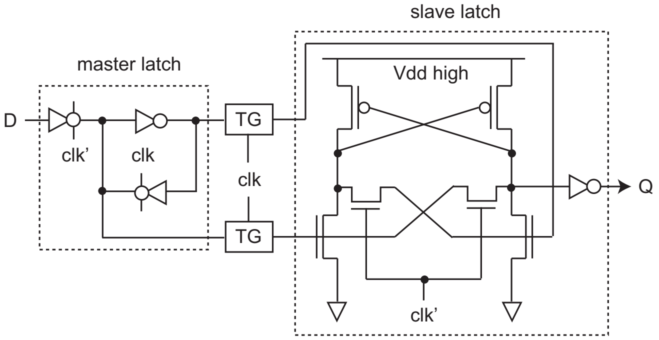

Fig. 2. Illustration of level converter FF

Subject to:

$$r(u) - r(v) \leq w(e_{u,v}),\ \forall e_{u,v} \in E \tag{2}$$

$$r(u) - r(v) \leq W(u,v) - 1,\ \forall D(u,v) > T,\ \forall u,v \in V \tag{3}$$

The objective of the mathematical formulation is to minimize the total number of FFs under the clock period constraint. This is done by minimizing the total edge weight of the graph after retiming. If $FI(v) < FO(v)$, then the total number of FFs is reduced if $r(v) > 0$. Thus, the objective function tries to assign a valid $r(v) > 0$ for a node v with $FI(v) < FO(v)$. On the other hand, if $FI(v) > FO(v)$, then the total number of FF is reduced if $r(v) < 0$. Then the objective function tries to assign a valid $r(v) < 0$ for a node v with $FI(v) > FO(v)$. Constraint (2) states that the number of FFs on each edge after retiming cannot be negative. Constraint (3) states that there exists at least one FF on any path with delay more than T.

In [24], the authors suggest that the total number of edges that contain FFs is *maximized* in their retiming formulation for dynamic power reduction. The motivation is to increase the number of nodes off timing critical paths, which are ideal candidates for supply voltage assignment. This approach, however, tends to increase the FF count as well as the power consumed by the FFs. Table IV in Section V shows that the min-FF retiming indeed produces better total power reduction compared to the max-FF retiming. Note that the min-FF retiming, while it reduces the critical path delay as well as total power consumed by the FFs, may reduce the total slack in the circuit and thus limit the subsequent voltage assignment. However, Table III shows that the impact of the min-FF retiming on timing slack is minimal.

IV. Voltage Assignment Algorithm

A. ILP-based Voltage Assignment

The second step of our approach is to perform dual supply and threshold voltage assignment so that the total power (= dynamic plus leakage) consumed by the gates and level converters (LC) is minimized.[4] One issue with Vdd assignment is that the low-to-high Vdd conversion needs a special method to guarantee the reliable computation. There exist two ways to support this conversion [25], [26]. The first is to use a separate level converter (LC), which can be inserted anywhere in the circuit to raise the low-Vdd input voltage back to the high-Vdd level. An illustration is shown in Figure 1. The second is to use a FF that can handle the conversion as well, which is named the level conversion FF (LCFF). Figure 2 shows an illustration of LCFF proposed in [28]. In this paper we use both LCFFs and LCs so that LCs are used only on zero-weight edges (= edges with no FFs). Since both LCFF and LC cause additional delay and power, voltage assignment has to be done carefully to suppress the related delay/power overhead.

[4] The minimization of FF power consumption is addressed during the min-FF retiming. Since the number of FFs do not change during voltage assignment, we do not consider FF power consumption during voltage assignment.

Our initial formulation is integer linear programming-based since the voltage assignment variable for each node in the retimed graph takes one out of the following four possible states:

- state 1: high-Vdd plus low-Vth (maximum performance, maximum total power)
- state 2: low-Vdd plus low-Vth (medium performance, low dynamic power)
- state 3: high-Vdd plus high-Vth (medium performance, low leakage power)
- state 4: low-Vdd plus high-Vth (minimum performance, minimum total power).

In addition, the LC assignment variable for each edge either takes 0 (no LC) or 1 (with LC).

The following variables are used in our ILP-based voltage assignment formulation:

- $x_{v,k}$: voltage assignment variable for node v into state k ($k = 1$ corresponds to high-Vdd+low-Vth, etc).
- $m(e)$: level converter assignment on edge e, where $m(e) = 1$ means LC is used on e; $w(e) = 0$ otherwise.
- $z_{v,k}$: supply voltage level of v given that v is assigned to voltage state k.
- $p_{v,k}$: total power consumption of v given that v is assigned to voltage state k.
- $d_{v,k}$: delay of v given that v is assigned to voltage state k.
- $s(v)$: arrival time of node v.
- p_{lc}, d_{lc}: total power consumption and delay of a level converter.
- T: clock period constraint.
- D: difference between high Vdd and low Vdd.

Our ILP-based dual supply/threshold voltage assignment for total power reduction under timing constraint is formulated as follows:

$$\text{Minimize } (\sum_{v \in V} \sum_{k=1}^{4} p_{v,k} \cdot x_{v,k}) + (\sum_{e \in E} p_{lc} \cdot m(e)) \tag{4}$$

Subject to:

$$\sum_{k=1}^{4} x_{v,k} = 1, \ \forall v \in V \tag{5}$$

Timing constraints:

$$\sum_{k=1}^{4} d_{v,k} \cdot x_{v,k} + s(v) \leq T, \ \forall v \in V \tag{6}$$

$$\sum_{k=1}^{4} d_{u,k} \cdot x_{u,k} + d_{lc} \cdot m(e) + s(u) \leq s(v), \ \forall e_{u,v} \in E \tag{7}$$

$$s(v) \geq 0, \ \forall v \in V \tag{8}$$

Level converter (LC) constraints:

$$\sum_{i=1}^{4} z_{u,i} \cdot x_{u,i} - \sum_{j=1}^{4} z_{v,j} \cdot x_{v,j} + Dm(e) \geq 0, \ \forall e_{u,v} \in E \tag{9}$$

Integer constraints:

$$x_{v,k} \in \{0, 1\}, \ \forall v \in V \tag{10}$$

$$m(e) \in \{0, 1\}, \ \forall e \in E \tag{11}$$

The objective of ILP is to minimize the total power consumption on all gates and level converters used. Constraints (5) and (10) state that each gate can be assigned to only one voltage state. Constraint (6) guarantees that the arrival time of each node combined with its delay is always less than the target clock period. Constraint (7) states that the arrival time of node v has to be greater than the summation of the arrival time of node u, the delay of node u, and the delay of level converter inserted on $e_{u,v}$. Constraint (9) states that if a low Vdd gate u drives a high Vdd gate v, a level converter is inserted onto $e_{u,v}$.

B. Linear Programming Relaxation

Our related experiment shown in Section V indicates that the computational effort to solve the ILP-based voltage assignment quickly becomes prohibitive as the size of the circuit increases. In this section, we propose a method to relax the ILP formulation into LP to overcome this limitation. We first solve the LP-relaxed version of the original ILP problem, which requires a few orders of magnitude smaller runtime. Next, we convert the non-integral LP solution into integral ILP solution while satisfying the level conversion and clock period constraint. The objective of our LP remains the same: minimization of total power consumed by the gates and level converters. One of the biggest challenges is the continuous (LP) to integral (ILP) conversion

```
LP-based Voltage Assignment Algorithm
input: retimed graph $G(V, E)$
output: dual Vdd/Vth assignment and LC insertion

// initial solution
1.  set $m(e) = 0$ for all edges and solve LP;
2.  voltage_mapping;
3.  $best$ = compute total power;
// main loop
4.  $itr = gain = 0$;
5.  while ($gain \geq gain\_limit$ and $itr < max\_itr$)
6.      compute new $m_{th}$;
7.      for (each edge $e \in E$)
8.          if ($m(e) > m_{th}$) then $m(e) = 1$;
9.          else $m(e) = 0$;
10.     solve LP;
11.     if (timing is met)
12.         voltage_mapping;
13.         $curr$ = compute total power;
14.         update $best$;
15.     $itr$++;
```

Fig. 3. Linear Programming relaxation algorithm to solve the ILP-based voltage assignment problem. "gain" denotes the total power saving.

of the voltage assignment (= $x_{v,k}$) and level converter assignment (= $m(e)$) variables. Our basic approach is to iteratively search for the best possible $m(e)$ assignment while using $x_{v,k}$ conversion algorithm to guide the search process.

Our LP formulation uses the same objective and constraints as the original ILP formulation, i.e., we minimize Equation (4) under the constraints (5) to (9). Instead of (10) and (11), however, we use the following non-integral constraints:

$$0 \leq x_{v,k} \leq 1, \ \forall v \in V \tag{12}$$

$$0 \leq m(e) \leq 1, \ \forall e \in E \tag{13}$$

Figure 3 shows our LP-based voltage assignment algorithm. Our basic approach is to first map $m(e)$ values into binary and use them to map $x_{v,k}$ values into binary. We then repeat this process with new $m(e)$ values until there is no further reduction on the total power. More specifically, we use a threshold value m_{th} to first map $m(e)$ into binary values (lines 7-9). We then solve the LP problem based on these binary $m(e)$ values and see if the timing constraints are met (lines 10-11). If so, we use a heuristic algorithm named voltage_mapping discussed in the next section to map the continuous $x_{v,k}$ values to binary (lines 12-14).[5] We perform a gain-based gradient search to obtain a new m_{th} value (line 6) and repeat the whole process and see if the total power is further minimized under the new LC assignment. This search continues until the gain is not significant or the number of iterations has exceeded a certain limit (line 5).

We obtain the baseline solution by setting $m(e) = 0$, solving LP, and performing the voltage mapping (lines 1-3). Note that fixing $m(e) = 0$ for all edges means we do not allow any LC to be inserted after the voltage assignment. In other words, the voltage assignment is severely restricted such that there should be no edge $e_{u,v}$ that connects a low Vdd node u to a high Vdd node v unless $w(e) > 0$, i.e. a FF exists on e. Nonetheless, it is still possible to reduce the total power under this restriction, and the final result becomes our baseline solution. We perform gradient search to obtain a new target threshold value m_{th} (initial m_{th} value is 0.5), where the total power reduction during the last two iterations are used to compute a new target. Note that the power gain is not linearly dependent on m_{th}. It is possible to obtain more power reduction with higher and/or lower m_{th} value. In case of a high m_{th} value, the number of LCs added is small, thereby reducing the power consumed by LCs. However, this limits the voltage assignment opportunity. In case of a low m_{th} value, however, the larger number of LCs added increases the power consumed by LCs but allows more rigorous voltage assignment.

C. Voltage Mapping

The main objective of our voltage mapping stage is to map the continuous voltage assignment variables $x_{v,k}$ resulting from our LP formulation to binary values. There exist two major constraints during this mapping: LC (level converter) and timing constraints. Since we have performed LC insertion before calling the voltage mapping step, the supply voltage assignment has to honor the existing LCs, i.e., there should always be low-Vdd to high-Vdd transition on each edge e with LC as expressed

[5] Note that it is still possible for the LP to obtain non-feasible solutions when too many LCs are inserted along the critical paths. In this case, voltage assignment may not be able to fix all timing violations.

```
Voltage Mapping Algorithm
input: LP-based voltage assignment with LC inserted
output: ILP-based voltage assignment with reduced LC set
1.  $T$ = topological ordering of gates;
2.  assign low-Vdd+high-Vth to all PIs;
3.  while ($T$ is not empty)
4.      $v = T$.pop;
5.      $dly(v) = \sum_{k=1}^{4} x_{v,k} \cdot d_{v,k}$;
6.      $vdd(v) = x_{v,1} + x_{v,3}$;
7.      $v \leftarrow$ Vdd-L+Vth-H;
// Vdd mapping
8.      if ($\exists u \in FI(v) | u =$ Vdd-L and $m(e_{u,v}) = 1$)
9.          $v \leftarrow$ Vdd-H;
10.     if ($vdd(v) > 0$)
11.         $v \leftarrow$ Vdd-H;
// LC removal
12.     if ($\exists u \in FI(v) | u =$ Vdd-H & $m(e_{u,v}) = 1$ or
        $u =$ Vdd-L & $m(e_{u,v}) = 1$ and $v =$Vdd-L)
13.         $m(e_{u,v}) \leftarrow 0$
// Vth mapping
14.     if ($v =$ Vdd-H & $dly(v) <$ delay(Vdd-H+Vth-H))
15.         $v \leftarrow$ Vth-L;
16.     if ($v =$ Vdd-L & $dly(v) <$ delay(Vdd-L+Vth-H))
17.         $v \leftarrow$ Vth-L;
```

Fig. 4. Voltage mapping algorithm under LC and timing constraints. $k = 1$ and $k = 3$ denote the high Vdd state in line 6.

in Equations (9) and (11). In addition, the voltage mapping should be done in such a way that no node after the voltage mapping should violate the clock period and arrival time constraints as expressed in Equations (6), (7), and (8). Since the voltage mapping step picks only one of four continuous assignment variables ($x_{v,1}$, $x_{v,2}$, $x_{v,3}$, $x_{v,4}$) and makes it 1 while fixing others to 0 for each node v, Equations (5) and (10) are also satisfied.

Figure 4 shows our voltage mapping algorithm. Since the goal is to reduce the total power under LC and timing constraints, more low-Vdd and high-Vth nodes means more power reduction as long as these constraints are not violated. Note that a simple maximum function may not guarantee the LC and timing constraints. For example, if $x_{v,1} = 0.2$, $x_{v,2} = 0.2$, $x_{v,3} = 0.4$, and $x_{v,4} = 0.2$, then this "maximum" scheme assigns high-Vdd plus high-Vth ($k = 3$) to v. In our algorithm, we visit each node in a topological order so that the voltage mapping for all fan-in nodes is done when visiting a new node (line 1). The PIs are initialized to low-Vdd+high-Vth (line 2). For each node in a topological order, we first compute $dly(v) = \sum_{k=1}^{4} x_{v,k} \cdot d_{v,k}$ and $vdd(v) = x_{v,1} + x_{v,3}$ (lines 5-6). $dly(v)$ denotes the delay of node v based on the continuous voltage assignment, and $vdd(v)$ denotes the sum of high-Vdd related continuous variables.

Our approach is to decide the best possible voltage mapping for the given node v based on the four possible scenarios shown in Figure 5. We start with the minimum total power configuration for each node, i.e., low-Vdd+high-Vth (line 7). We then decide whether we must raise the Vdd (lines 8-11) or lower the Vth (lines 14-17) based on the LC and timing constraints. During the Vdd mapping step, we first see for a given node v if there is any fan-in node u with low Vdd assigned and $e_{u,v}$ contains an LC. If so, a high-Vdd has to be assigned to v to satisfy the LC constraint (lines 8-9). Next, if $vdd(v) > 0$, the previous linear programming partially assigned high-Vdd to v, and raising v to high-Vdd will never violate timing constraints (lines 10-11).

At this point, it is important to note that some LCs become unnecessary during the PI-to-PO Vdd mapping process such as case 5, 7, and 8 in Figure 5. Thus, our LC removal step (lines 12-13) deletes these unnecessary LCs if (i) a high-Vdd node drives a low or high-Vdd node while using an LC (case 5 and 7), or (ii) a low-Vdd node drives another low-Vdd node while using an LC (case 8). Since LC removal never increases the overall delay, the timing constraint is never violated. During the subsequent Vth mapping, our goal is to see if the initial high-Vth has to be adjusted due to timing constraints—if $dly(v)$ lies in between the delay of a high-Vth gate and a low-Vth gate, low-Vth assignment will guarantee to satisfy the timing constraint at the expense of slight leakage power increase. The runtime of our voltage mapping algorithm is $O(|V| + |E|)$, where $|E|$ is number of edges, since it involves topological sorting.

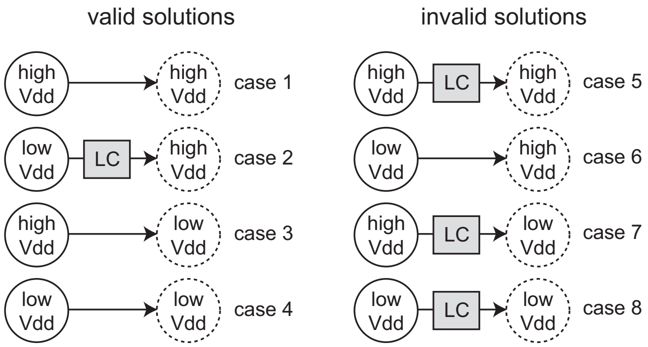

Fig. 5. 8 possible supply voltage assignment for dotted nodes. The invalid solutions are either non-optimal (= additional LC only increases total power) or violate LC constraint, and thus LP will never generate them.

```
Post Refinement
input: retimed and voltage-scaled solution
output: refined voltage assignment solution
// clustering
1.  perform static timing analysis;
2.  mark all nodes with positive timing slack;
3.  form clusters among marked nodes;
4.  sort clusters based on its size;
// main loop
5.  for (each cluster $C$)
6.      while (there is power reduction)
7.          for (each node $v \in C$)
8.              power_gain($v, slk(v), C$);
9.          $z$ = max power gain node;
10.         commit voltage change for $z$;
11.         update slack for downstream nodes of $z$;
```

Fig. 6. Our cluster-based post refinement algorithm that performs voltage assignment under LC and timing constraints. Each cluster contains a set of reachable nodes with positive timing slack.

D. Post Refinement

The last step of our algorithm is the post refinement, where an additional voltage assignment is applied to the solution we obtained from the previous steps, namely, retiming and ILP/LP voltage assignment. The primary concern during voltage mapping discussed in Section IV-C is to satisfy the LC and timing constraints. Thus, our focus is to accept voltage mapping that will never violate the timing constraint for each node, which results in a delay reduction for each node in most cases. The change in the delay of a node affects the delay of all of its downstream nodes in a directed graph and may allow additional power reduction among them. Thus, a positive timing slack $slk(v)$ (= required time minus arrival time from static timing analysis) resulting from our conservative voltage mapping needs to be propagated downwards to correctly reflect the slack change globally. However, our voltage mapping does not perform static timing analysis (= timing slack re-computation) upon the voltage mapping of each node due to its prohibitive runtime, which may hide some power reduction opportunity. Thus, the voltage mapping based on the initial timing slack is a primary source of non-optimality. In addition, our LP formulation discussed in Section IV-B may assign $m(e)$ values that are not close to 0 or 1 for potentially many edges. Thus, relying on a single threshold value to decide which edge gets LC or not for *all* edges is another source of non-optimality.

Figure 6 shows our post refinement algorithm. The basic idea is to identify the nodes with positive timing slack and try to reduce their total power consumption by additional voltage assignment under timing and LC constraints. This time, however, we examine the impact of the proposed voltage assignment of each node on *all* affected nodes. We first perform clustering based on timing slack, where each cluster contains a set of reachable nodes with positive slack (lines 1-4). In this case, we visit the largest cluster first (line 4) since our exploration is limited to the nodes inside each cluster and thus the more (and earlier) the nodes examined to see the impact of voltage assignment the better. We visit each cluster (line 5) and compute the total power gain for each node in the cluster (lines 7-8). During the power gain computation of each node v, we compute the power reduction for v as well as all of its predecessors inside the cluster using our recursive algorithm power_gain shown in Figure 7 (to be discussed later). We then select the node that results in the maximum power reduction and commit the voltage change (lines 9-10). Lastly, we update the timing slack for all downstream nodes of the max-gain node (line 11). We continue

```
power_gain $(v, dly, C)$
input: a node $v \in C$ and timing slack $dly$
output: maximum power saving for $v$
1. mark $v$ visited;
2. $x$ = current voltage state of $v$;
// find best new voltage state
3. for (each voltage state $i \neq x$)
4.     $\Delta p$ = power reduction from state $x \rightarrow i$;
5.     $\Delta d$ = delay increase from state $x \rightarrow i$;
6.     if ($\Delta p > 0$ and $slk(v) > \Delta d + dly$)
7.         mark $i$ feasible;
8. $y$ = feasible voltage state with max $\Delta p$;
9. $dly' = dly + \Delta d$ based on $x \rightarrow y$;
10. $tot\_gain = \Delta p$ based on $x \rightarrow y$;
// recursive call
11. for (each non-visited fan-in $u \in C$)
12.     if($slk(u) > dly'$)
13.         $gain$ = power_gain($u, dly', C$);
14.         $tot\_gain = tot\_gain + gain$;
15. return $tot\_gain$;
```

Fig. 7. A recursive algorithm that computes the total power gain of a given node and all of its predecessors from voltage assignment refinement.

to target the same cluster until there is no further power gain (line 6). There exist $O(n)$ clusters in the worst, and each cluster performs $O(n \log n)$ static timing analysis K times, where K is the maximum size among the clusters. Thus, the worst-case complexity of our post refinement algorithm is $O(n^2 \log n)$. However, the number of clusters is usually much smaller than the number of nodes, and K is a small integer. Thus, the practical runtime is $O(n \log n)$.

Figure 7 shows our recursive algorithm that computes the total power gain of a given node and all its predecessors inside the given cluster. The voltage assignment and thus the increase in the delay of a node v reduces the timing slack of many of its downstream nodes. Thus, it is unlikely that there exists any power saving opportunity via voltage assignment among the downstream nodes. The upstream nodes of v, however, are not affected by the change in the delay of v. Thus, we limit our exploration to v and its predecessors to examine the impact of voltage assignment v. In addition, the reason we limit our search to the nodes inside the given cluster is because it is not possible for the zero-slack nodes outside the cluster to accommodate the delay increase without timing violation. For a given node v, we compute the power saving and delay increase for each candidate power state (lines 3-7). Among the feasible power states, we pick the one with the maximum total power reduction (lines 8-10). We visit the predecessors and keep track of the power gain from them (line 11-14). Finally, we return the total gain (line 15) as the final output.

Note that the computation of Δp (line 4) and Δd (line 5) considers the impact of additional LC insertion. For instance, if the Vdd is currently set to high for a given node u and one of its fanout v is set to high-Vdd while $m(e_{u,v}) = 0$, Δp for high-to-low Vdd adjustment for u should not only include the power saving from Vdd assignment but also the power increase from the LC that has to be inserted. In addition, Δd should include the delay increase from high-to-low Vdd adjustment as well as the LC insertion. Moreover, this delay change from Vdd adjustment further affects the subsequent Vth assignment and its corresponding leakage power. A similar argument applies when we examine the impact of related LC removal from a low-to-high Vdd adjustment on dynamic/leakage/delay tradeoff.

E. Wire Delay Consideration

Our discussion so far has been based on retiming and voltage assignment for unplaced netlist. Thus, our formulation has focused on gate/FF power saving. This is reasonable because the interconnect length is not known without placement information. In addition, our related experiments presented in Section V show that the power saving based on gate voltage assignment is still significant. However, our algorithms can be easily extended to consider placed netlists. In this case, the retiming needs to consider wire delay impact during retiming calculation. In addition, our voltage assignment should include wire delay impact during the arrival time update. Since the wirelength does not change during these steps, we compute the wire delay values once before our algorithms and use these static values throughout the algorithms.[6]

First, we add delay to each wire, denoted $d(i, j)$, and include it in path delay computation, i.e., $d(P(i, j))$ is the sum of gate as well as wire delay along the path $P(i, j)$. This is then used in the delay matrix $D(i, j)$ introduced in Section III-B so

[6] We note that the power consumed by interconnect is proportional to wirelength as well as the switching activities of the driving gates. The integration of such placement technique for wire power saving is beyond the scope of this paper.

TABLE I

BENCHMARK CIRCUIT CHARACTERISTICS BEFORE RETIMING

ckt	#gate	#PI	#PO	#FF
s641	379	35	42	19
s713	393	35	42	19
s820	289	18	24	5
s832	287	18	24	5
s838	446	34	33	32
s1196	529	14	32	18
s1238	508	14	32	18
s1488	653	8	25	6
s1494	647	8	25	6

TABLE II

DELAY (IN ps), DYNAMIC POWER (IN nW), AND LEAKAGE POWER (IN nW) OF THE GATES, LC, AND LCFF.

config	delay	dynamic	leakage
130nm			
High-Vdd/Low-Vth gate	31.75	60.18	7.96
Low-Vdd/Low-Vth gate	46.22	27.31	4.08
High-Vdd/High-Vth gate	43.33	54.92	0.05
Low-Vdd/High-Vth gate	81.68	24.44	0.03
level conversion FF	276.62	238.58	23.98
level converter	57.41	262.29	51.38
90nm			
High-Vdd/Low-Vth gate	22.52	48.23	20.16
Low-Vdd/Low-Vth gate	36.14	14.54	4.48
High-Vdd/High-Vth gate	26.79	27.66	1.23
Low-Vdd/High-Vth gate	54.33	9.96	0.26
level converter FF	253.38	209.82	126.77
level conversion	40.88	251.96	208.46

that the min-FF retiming becomes interconnect-aware. Given a placement of gates and FF, the min-FF retiming repositions FF so that the number of FFs is minimized under clock period constraint. Second, Equation (7) is changed as follows to reflect the wire delay during voltage assignment:

$$\sum_{k=1}^{4} d_{u,k} \cdot x_{u,k} + d_{lc} \cdot m(e) + d(u,v) + s(u) \leq s(v), \ \forall e_{u,v} \in E$$

We assume that the wire delay values stay constant during voltage mapping and post refinement phase. This assumption is reasonable since the majority of nets remain untouched except for the ones losing and gaining LCs. In this case, the wire delay change is minimal since LC removal and insertion from the related routing update is minimal. Our related results on retiming and voltage assignment for placed netlists indicate that the total power saving is even more significant with wire delay consideration.

V. EXPERIMENTAL RESULTS

A. Experimental Setting

Our algorithm named Retiming-based Voltage Assignment (RVA) is implemented in C++/STL and run on a Pentium IV 2.8 GHz machine. The solutions to the LPs were found using the Gnu Linear Programming Kit's [35] version 4.5. Our benchmark set consists of nine sequential circuits from ISCAS89 benchmark [36] circuits. The benchmark characteristics before retiming are summarized in Table I. #gate represents the total number of gates, #PI and #PO represent the total number of primary inputs and primary outputs, respectively, and #FF represents the total number of flip-flops. The benchmark circuits are mapped to inverter, 2 to 5-input AND/OR, and FFs.

The delay, dynamic, and leakage power consumption of the gates, FFs, and LCs are computed using HPSICE. The simulations are performed using BSIM3 model parameters for the 130nm process [37] and 90nm process [38]. For the 130nm process, Vdd high/low is set to 1.2V/0.8V while Vth high/low is set to 0.42V/0.24V. For the 90nm process, Vdd high/low is set to 1.0V/0.6V. Vth high/low is set to 0.25V/0.15V. The values of delay, dynamic power, and leakage power of gate, LC, and FFs are shown in Table II.[7] We assume 20% average switching activities for the gates. The switching activities are randomly assigned.[8]

[7] All results except for the last one (= Table VIII) are based on 130nm.

[8] Switching activity as well as the size of the gates have considerable impact on the power consumption. Consideration of these factors in determining the Vdd/Vth assignment, however, is out of the scope of this paper.

TABLE III

IMPACT OF MIN-FF RETIMING ON TOTAL TIMING SLACK

ckt	before	after
s641	60068	52280
s713	63589	54693
s820	6660	6512
s832	11180	7804
s838	16870	16870
s1196	35092	35092
s1238	29947	29947
s1488	19025	19024
s1494	16262	14433

TABLE IV

IMPACT OF RETIMING OBJECTIVE ON POWER MINIMIZATION. WE REPORT THE TOTAL POWER CONSUMED BY THE GATES/LCS (= GL) AS WELL AS THE GATES/LCS PLUS FFS (= GLF).

	max-FF retiming		min-FF retiming	
ckt	GL	GLF	GL	GLF
s641	64.02	133.38	66.59	75.66
s713	71.10	144.36	71.52	80.59
s820	77.00	186.60	78.85	162.83
s832	74.75	185.82	86.74	173.10
s838	90.23	212.81	78.15	159.75
s1196	115.73	291.26	125.21	139.52
s1238	119.24	300.11	131.57	146.36
s1488	152.96	344.98	156.82	264.18
s1494	152.42	347.45	158.29	266.61
GL	1.00		1.04	
GLF		1.00		0.69

B. *Impact of Retiming*

In Table III, we show the total timing slack among all nodes before and after the min-FF retiming. The purpose is to investigate the impact of retiming on the subsequent voltage assignment. The nodes with larger timing slack, i.e., the nodes off timing critical paths, are the prime target for voltage assignment. We observe that the impact of the min-FF retiming on the timing slack is minimal, suggesting that min-FF retiming is not interfering with the subsequent voltage assignment. Moreover, min-FF retiming helps reduce the total power by minimizing the power consumed by FFs as shown in Table IV (to be discussed). The runtime for retiming ranged from a few seconds to one minute.

In Table IV, we show the impact of retiming objective (max-FF vs min-FF) on total power minimization. Our LP-based voltage assignment (both Vdd and Vth) is performed after the retiming step. We report the total power consumed by the gates/LC (= GL) as well as by the gates/LC/FF (= GLF). We first observe that the GLF values are significantly higher than GL values regardless of the retiming objective. This indicates that the FF power must be considered during the computation and optimization of total power consumption. Next, we observe that the GL result is slightly better with max-FF retiming. This is possible since max-FF objective may help reduce the number of LCs need to be inserted. However, when the power consumed by the FFs is considered, i.e., GLF result, min-FF obtains significantly better results (31% on average). This strongly supports our claim that min-FF objective is a better choice for the total power minimization.

C. *Voltage Assignment Results*

Table V shows the total number of nodes under each voltage configuration. We also report the number of LCs used. We first observe that a significant portion of the gates is assigned high-Vdd/high-Vth. These gates are often used to reduce the leakage power while meeting the timing constraints. The low-Vdd/low-Vth gates also provide the same kind of effect as high-Vdd/high-Vth gates. However, the Vdd assignment has more impact on the delay increase than Vth assignment, which is why low-Vdd/low-Vth gates are not used as often as high-Vdd/high-Vth gates due to level converter requirement. Next, the usage of high-Vdd/low-Vth (maximum power, minimum delay) is inevitable for timing critical nodes due to the timing constraints. We observe that the usage of high-Vdd/low-Vth gates increases as the circuit size increases. The minimum power configuration (= low-Vdd/high-Vth) is used heavily for almost all circuits to reduce the total power consumption. It is interesting to note that the demand for high-Vdd/low-Vth and LC becomes higher as the size of the circuits increases.

Table VI shows the breakdown of total power into leakage (for all gates), dynamic (for all gates), LC power (dynamic+leakage) and FF power (dynamic+leakage). We note that a significant portion of the total power is consumed by the FFs and LCs consistently. In addition, the dynamic power is still higher than leakage for 130nm technology.

TABLE V

Voltage assignment breakdown

ckt	Vdd-L Vth-H	Vdd-L Vth-L	Vdd-H Vth-H	Vdd-H Vth-L	LC
s641	111	0	244	84	24
s713	103	6	262	82	27
s820	101	30	109	88	48
s832	56	45	93	132	37
s838	177	30	187	89	61
s1196	89	10	268	192	37
s1238	89	11	227	211	41
s1488	181	67	154	280	85
s1494	202	40	152	282	87

TABLE VI

Power consumption breakdown

ckt	dynamic	leakage	LC	FF	total
s641	47.24	6.73	12.62	9.07	75.66
s713	50.30	7.02	14.20	9.07	80.59
s820	47.02	6.58	25.25	83.98	162.83
s832	56.54	10.73	19.46	86.37	173.10
s838	37.89	8.17	32.09	81.60	159.75
s1196	90.95	14.79	19.46	14.32	139.52
s1238	93.60	16.40	21.57	14.79	146.36
s1488	87.41	24.70	44.71	107.36	264.18
s1494	90.04	22.49	45.76	108.32	266.61

D. Comparison with Existing Works

Table VII and Table VIII show total power comparison among the following voltage assignment methods on 130nm process technology and 90 process technology:

- UPP: all gates are assigned high-Vdd and low-Vth. This voltage assignment corresponds to the maximum possible total power consumption.
- LOW: all gates are assigned low-Vdd and high-Vth. This voltage assignment corresponds to the minimum possible total power consumption. Note that timing constraint may be violated here.
- CVS: we report the well-known *Clustered Voltage Scaling* results [1].
- ECVS: we report the *Extended Clustered Voltage Scaling* results [39].
- LX: LX is our RVA algorithm without post refinement presented in Section IV-D
- LP: the solution to our LP formulation shown in Section IV-B without voltage mapping. This method provides optimal results but assigns non-integer values in $[0, 1]$ to the LC and voltage assignment variables. This LP assignment is not usable but provides a useful baseline to evaluate our voltage mapping heuristic presented in Section IV-C.
- RVA: our RVA algorithm solves the LP above and maps the non-integer assignment values to integer using our voltage mapping heuristic and post refinement.

We summarize our observations here:

- We note that there exist a significant room for total power improvement from gate-level voltage assignment, which is evident from UPP and LOW columns.
- Comparison of LP vs RVA reveals the effectiveness of our voltage mapping and post refinement heuristics. We observe that our RVA results are within 9% to the optimal LP results.
- Comparison of LX vs RVA reveals the effectiveness of our post refinement heuristics. We observe that our RVA results are 8% better than LX for 130nm process technology and 3% better for the 90nm process technology.
- RVA outperforms CVS/ECVS by 22% on average for 130nm process technology and 28% on average for 90nm process technology. Note that our comparison to CVS/ECVS is not fair since CVS/ECVS groups the gates with the same voltage into the same cluster. The advantage of CVS/ECVS is easier power/ground routing since the gates with same voltage tend to be placed nearby. However, our experiment shows that CVS/ECVS cannot exploit the maximum energy savings possible with dual supply voltages. This in turn means that there exists power saving vs. congestion tradeoff.

VI. Conclusions

This paper presented an ILP-based method to simultaneously assign supply and threshold voltages to individual gates for dynamic and leakage power minimization. Our method consists of three steps: low power retiming, ILP-based voltage assignment, and post refinement. We relax the ILP formulation into LP, solve the LP in an iterative manner, and perform several heuristics to convert LP solutions back to ILP. The related experiments show that we obtain solutions that are very close to pure ILP approach within a fraction of runtime while outperforming several well-known methods. Our ongoing work

TABLE VII

Comparison among various voltage assignment algorithms in 130nm process technology. "dly" denotes the critical path delay after retiming.

ckt	dly	UPP	LOW	CVS	ECVS	LX	RVA	LP
s641	360	118.63	39.50	106.09	106.08	77.06	75.66	75.04
s713	371	126.01	42.06	111.59	111.59	83.29	80.59	77.16
s820	84	207.94	123.82	182.40	182.31	179.35	162.83	118.71
s832	74	211.61	126.79	177.35	177.05	181.02	173.10	134.63
s838	105	219.10	121.96	187.40	185.31	170.83	159.75	124.41
s1196	169	185.31	65.86	174.42	174.42	145.86	139.52	90.78
s1238	154	187.96	67.89	177.53	177.53	151.61	146.36	99.85
s1488	108	334.13	177.61	280.44	274.64	278.72	264.18	224.18
s1494	104	335.32	178.85	287.07	283.91	283.32	266.61	225.29
RATIO	-	1.00	0.47	0.88	0.88	0.75	0.69	0.60

TABLE VIII

Comparison among various voltage assignment algorithms in 90nm process technology. "dly" denotes the critical path delay after retiming.

ckt	dly	UPP	LOW	CVS	ECVS	LX	RVA	LP
s641	326	161.11	26.29	136.44	136.44	73.97	70.11	69.59
s713	355	168.94	27.37	146.52	146.52	76.60	74.69	71.78
s820	87	263.04	135.54	206.18	195.25	200.03	191.94	142.89
s832	90	267.15	139.15	220.96	217.05	198.52	191.03	160.52
s838	115	291.89	132.79	237.15	233.83	192.58	171.31	132.80
s1196	146	234.28	42.61	211.88	211.88	139.35	134.07	103.89
s1238	163	233.81	43.86	212.46	212.46	129.79	122.91	96.04
s1488	116	427.20	181.81	311.71	301.32	310.64	293.95	257.36
s1494	116	427.91	183.26	351.31	346.18	317.75	295.40	242.00
RATIO	-	1.00	0.34	0.83	0.82	0.63	0.60	0.5

includes an extension of our work to consider interconnect power reduction via an integration with power-aware placement. In addition, the combination of our work with other well-known circuit-level power reduction schemes such as gate sizing, state assignment, and gate oxide assignment will achieve even more total power reduction.

References

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

[2] C. Chen, A. Srivastava, and M. Sarrafzadeh, "On Gate Level Power Optimization Using Dual-Supply Voltages," *IEEE Trans. on VLSI Systems*, vol. 9, no. 5, pp. 616–629, October 2001.

[3] J. Chang and M. Pedram, "Energy minimization using multiple supply voltages," *IEEE Trans. on VLSI Systems*, vol. 5, no. 4, pp. 436–443, 1997.

[4] C. Chen, A. Srivastava, and M. Sarrafzadeh, "On gate level power optimization using dual-supply voltages," *IEEE Trans. on VLSI Systems*, vol. 9, no. 5, pp. 616–629, 2001.

[5] Q. Wang and S. Vrudhula, "Algorithms for minimizing standby power in deep submicron, dual-Vt CMOS circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 3, pp. 306–318, 2002.

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

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

[8] Y. S. D. et al, "Algorithm for achieving minimum energy consumption in CMOS circuits using multiple supply and threshold voltages at the module level," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2003, pp. 693–700.

[9] A. Srivastava and D. Sylvester, "Minimizing Total Power by Simultaneous Vdd/Vth Assignment," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 665–677, 2004.

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

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

[12] P. Pant, R. Roy, and A. Chatterjee, "Dual-Threshold Voltage Assignment with Transistor Sizing for Low Power CMOS Circuits," *IEEE Trans. on VLSI Systems*, vol. 9, no. 2, pp. 390–394, 2001.

[13] D. Nguyen, A. Davar, M. Orshansky, D. Chinnery, B. Thompson, and K. Keutzer, "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.

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

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

[16] R. Brodersen, M. Horowitz, D. Markovic, B. Nikolic, and V. Stojanovic, "Methods for True Power Minimization," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2002.

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

[18] W. Hung, Y. Xie, N. Vijaykrishnan, M. Kandemir, M. Irwin, and Y. Tsai, "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.

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

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

[21] J. Monteiro, S. Devadas, and A. Ghosh, “Retiming sequential circuits for low power,” in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1993.

[22] F. Sheikh, A. Kuehlmann, and K. Keutzer, “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*, 2002, pp. 43–49.

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

[24] N. Chabini and W. Wolf, “Reducing Dynamic Power Consumption in Synchronous Sequential Digital Designs Using Retiming and Supply Voltage Scaling,” *IEEE Trans. on VLSI Systems*, vol. 12, no. 6, pp. 573–589, June 2004.

[25] S. Kulkarni and D. Sylvester, “New level converters and level converting logic circuits for multi-VDD low power design,” in *Proc. IEEE Int. SOC Conf.*, 2003.

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

[27] C. E. Leiserson and J. B. Saxe, “Retiming synchronous circuitry,” *Algorithmica*, pp. 5–35, 1991.

[28] K. Usami and M. Igarashi, “Low-power design methodology and applications utilizing dual supply voltages,” in *Proc. Asia and South Pacific Design Automation Conf.*, 2000.

[29] L. Carley, A. Aggarwal, and R. Krishnamurthy, “Decreasing Low-Voltage Manufacturing-Induced Delay Variations with Adaptive Mixed-Voltage-Swing Circuits,” in *Proc. Int. Symp. on Low Power Electronics and Design*, 1998.

[30] L. Carley and A. Aggarwal, “A completely On-Chip Voltage Regulation Technique for Low Power Digital Circuits,” in *Proc. Int. Symp. on Low Power Electronics and Design*, 1999.

[31] N. Dragone, A. Aggarwal, and L. Carley, “An Adaptive On-Chip Voltage Regulation Technique fow Low-Power Applications,” in *Proc. Int. Symp. on Low Power Electronics and Design*, 2000.

[32] A. Diril, Y. Dhillon, A. Chatterjee, and A. Singh, “Level-Shifter Free Design of Low Power Dual Supply Voltage CMOS Circuits Using Dual Threshold Voltages,” *IEEE Trans. on VLSI Systems*, 2005.

[33] Y. Yeh, S. Kuo, and J. Jou, “Converter-free multiple-voltage scaling techniques for low-power CMOS digital design,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2001.

[34] L. Wei, K. Roy, and V. De, “Low Voltage Low Power CMOS Designs Techniques for Deep-Submicron ICs,” in *Intl. Conf. on VLSI Design*, 2000.

[35] GLPK, “GLPK (GNU linear programming) kit.” [Online]. Available: http://www.gnu.org/software/glpk/glpk.html

[36] ISCAS89, “The ISCAS 1989 benchmark suite.” [Online]. Available: http://www.cbl.ncsu.edu

[37] T. M. Service.[Online], http://www.mosis.org.

[38] P. T. M. [Online], http://www.eas.asu.edu/ ptm.

[39] K. U. et al., “Design methodology of ultra low-power MPEG4 codec core exploiting voltage scaling techniques,” in *Proc. ACM Design Automation Conf.*, 1998, pp. 483–488.