The Epsilon-Approximation To Discrete VT Assignment For Leakage Power Minimization

Yujia Feng
Dept. of Electrical and Computer Engineering
Michigan Technological University
Houghton, Michigan 49931
yujia@mtu.edu

Shiyan Hu
Dept. of Electrical and Computer Engineering
Michigan Technological University
Houghton, Michigan 49931
shiyan@mtu.edu

ABSTRACT

As VLSI technology reaches $45nm$ technology node, leakage power optimization has become a major design challenge. Threshold voltage (vt) assignment has been extensively studied, due to its effectiveness in leakage power reduction. In contrast to the efficiently solvable continuous vt assignment problem, the discrete vt assignment problem is known to be NP-hard. All of the existing techniques are heuristics without performance guarantee due to the NP-hardness nature of the problem. It is still not known whether there is any rigorous approximation algorithm for the discrete vt assignment problem.

In this paper, the first ϵ-approximation algorithm is designed for the discrete vt assignment problem. The algorithm can ϵ-approximate the optimal vt assignment solution in $O(mn^3 \log n \log \frac{1}{\epsilon} + \frac{mn^3 \log^2 \frac{1}{\epsilon}}{\epsilon^2})$ time, where n is the size of the combinational circuit and m is the number of available threshold voltages per gate. It is based on an advanced potential function technique and an efficient dual decision core query technique. Our experiments on ISCAS'85 benchmark circuits demonstrate that the new algorithm always returns a solution with error bounded by ϵ even compared to the lower bound of the optimal solution. On average, it can approximate the optimal solution with 2.8% additional leakage power running in 51.3 seconds, while the integer linear programming technique is computationally prohibitive. Our algorithm also significantly outperforms the heuristic in [1] by 16.5% leakage power saving with similar runtime. This clearly demonstrates the practicality of the proposed ϵ-approximation algorithm for the vt assignment problem.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids - Placement and Routing; J.6 [**Computer-aided Engineering**]: Computer-aided Design

General Terms

Algorithms, Performance, Design

ICCAD'09, November 2–5, 2009, San Jose, California, USA.

Keywords

ϵ-Approximation, NP-Complete, VT Assignment, Leakage Power

1. INTRODUCTION

As VLSI technology reaches 45nm technology node and beyond, leakage power optimization has become a major design challenge. In practice, multi-threshold devices [2] are heavily deployed due to their effectiveness in trading off timing and leakage power by utilizing the exponential dependance between threshold voltage (vt) and leakage. In general, low vt is applied to the timing critical paths to keep the circuit performance while high vt is applied to the timing non-critical paths for reducing the leakage power consumption. In old technologies, dual-vt devices are popular, while in certain $45nm$ and $65nm$ technologies, more threshold voltages are available due to their increasing importance.

Vt assignment problem has been extensively studied in the literature such as [1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Some works focus on vt assignment with continuous threshold voltage values. However, in reality, only a small set of discrete threshold voltages are available. For those focused on discrete vt assignment problem, they are all heuristics without performance guarantee. This is not surprising as the problem is NP-hard [3]. For example, some works [9, 10] formulate the problem as mathematical (linear or geometric) programming problems followed by rounding the solution to the available discrete threshold voltage. The work [8] formulates the vt assignment problem as an integer linear program (ILP) and relies on a heuristic to solve it.

Despite extensive attempts made on the discrete vt assignment problem, whether there is any ϵ-approximation algorithm still remains unknown. This makes the discrete vt assignment problem not well understood. In this paper, we propose the first ϵ-approximation algorithm for the NP-hard timing constrained minimum leakage power vt assignment problem. The main contribution of this paper is summarized as follows.

- An algorithm is designed to ϵ-approximate the optimal discrete vt assignment solution in $O(mn^3 \log n \log \frac{1}{\epsilon} + \frac{mn^3 \log^2 \frac{1}{\epsilon}}{\epsilon^2})$ time, where n is the number of edges in the circuit and m is the number of available threshold voltages per gate.
- Our algorithm is the first ϵ-approximation algorithm on the NP-hard timing constrained minimum leakage power vt assignment problem.

- The algorithm involves an advanced potential function based integer linear programming technique and a dual binary-search-style decision core query technique.

- The new ϵ-approximation algorithm is highly practical. The experiments on ISCAS'85 benchmark circuits demonstrate that the new algorithm always returns a solution with error bounded by ϵ even compared to the lower bound of the optimal solution. On average, our algorithm can approximate the optimal solution within 2.8% additional leakage power and run in 51.3 seconds, while the integer linear programming technique is computationally prohibitive for these circuits. This significantly outperforms the vt assignment heuristic in [1] by 16.5% leakage power saving with similar runtime.

Finally, it is worth noting that a multitude of works [5, 9, 10, 8, 12] focus on simultaneous gate sizing and vt assignment, which are desired in the early and middle design stages in a physical synthesis flow. In these stages, gate sizing is quite effective for improving circuit timing/power, and thus simultaneous gate sizing and vt assignment can significantly improve the design quality. In contrast, in the very late design stage, applying gate sizing often necessitates the incremental placement which may hurt the turn-around-time. In addition, the utilization of such optimizations should have been stretched to the maximum extent during the previous design stages. It is the critical time to apply the vt assignment since it will not impact the placement while still enabling the significant improvement in timing.

The rest of the paper is organized as follows: Section 2 formulates the discrete vt assignment problem. Section 3 describes the algorithmic flow for the proposed ϵ-approximation to the discrete vt assignment problem. Section 4 describes the integer linear programming formulation for the discrete vt assignment problem. Section 5 describes the proposed ϵ-approximation algorithm in details. Section 6 presents the experimental results with analysis. A summary of work is given in Section 7.

2. PRELIMINARIES

In the discrete vt assignment problem, a combinational circuit is given where each gate size is fixed. At each gate, a set of threshold voltages are available. Gates with different threshold voltages lead to different tradeoff between circuit timing and power. The vt assignment problem is to determine the threshold voltage at each gate such that the timing satisfies the target while the leakage power is minimized.

Following many vt assignment works in the literature [1, 3, 8], the dynamic power of a gate is defined as $C_L Vdd^2 f$ where C_L is the load capacitance, Vdd is the supply voltage and f is the switching frequency. Vdd and f will not be changed by vt assignment. More importantly, C_L will not be changed as well since the interconnect capacitance and resistance certainly do not change and tuning the threshold voltage of a gate will not impact the gate capacitance. This is true for all the gates in the circuits and thus the load capacitance of any gate is fixed. Consequently, the dynamic power is not changed with threshold voltage [8].

Thus, the major power optimization target is set for leakage power in this paper. To demonstrate the effectiveness of the proposed algorithmic framework, the delay and leakage power models in [8] are adopted. However, our algorithm is not limited to this model. More accurate and more complicated model can be used as well. For example, one can use the piece-wise linear model in [11]. For a gate g, the leakage power can be computed as $c_g I_g(vt) Vdd$ [8] where c_g is a constant at a gate g. $I_g(vt)$ refers to the leakage current and is a function of the threshold voltage vt. On the other hand, the delay of a gate can be modeled as $d_g(vt) = a_g(vt) + b_g(vt) C_L$ where $a_g(vt)$ and $b_g(vt)$ are the functions of vt [8]. Given a combinational circuit where all gate sizes are fixed, the delay of a gate with each fixed vt assignment can be computed offline since the load capacitance C_L (containing the capacitances of both the interconnects and gates driven by the gate) does not change with vt assignment. Similarly, the leakage power of a gate with each fixed vt assignment can also be computed offline. These are critically different from the gate sizing problem since the gate delay and dynamic power cannot be computed offline there as they depend on downstream gate sizes.

Formally, the vt assignment problem can be formulated as follows. A combinational circuit C with n edges is given as a directed acyclic graph (DAG) where some gate nodes are primary input gates or primary output gates. In addition, the gate sizes are fixed but the gate threshold voltage can be chosen from a set of up to m available threshold voltages for each gate. For example, in certain $65nm$ technology, $m = 4$. For a gate g with a threshold voltage vt, its delay is $d(g, vt)$ and its leakage power is $p(g, vt)$. Note that they are computed offline as discussed above. Denote by $fin(g)$ the set of fan-in gates of the gate g. Let $PriIn(C)$ denote the primary input gates of C and $PriOut(C)$ denote the primary output gates of C, respectively. Denote by $VT(g)$ the set of available threshold voltages for gate g and by $vt(g)$ the assigned threshold voltage for gate g. Thus, $p(g, vt(g))$ refers to the leakage power of gate with the assignment $vt(g)$. Denote by $D(g)$ the maximum arrival time at the input of a gate g from any primary input gate. In this paper, we also call the leakage power the *cost*. Let $c_{i,j}$ denote the delay between two connecting gates g_i, g_j. It remains as a constant during vt optimization.

The *timing constrained minimum leakage power vt assignment* problem is to compute a vt assignment solution at each gate in the combinational circuit such that the total leakage power is minimized and the maximum circuit delay is no greater than a timing constraint T. Delay propagation for the combinational circuit is the same as that in the traditional gate sizing problem [14]. That is, for the nodes along a single branch, the overall delay is computed as the sum of the delay at each node. Given a node with multiple input branches, the delay at the node's input is the maximum delay among all input branches. It is well known that the vt assignment problem can be formulated to a mathematical programming problem as follows.

$$\begin{array}{ll} min & \sum_{i=1}^{n} p(g_i, vt(g_i)) \\ s.t. & D(g_i) + d(g_i, vt(g_i)) + c_{i,j} \leq D(g_j), \forall g_i \in fin(g_j) \\ & D(g_i) \leq T, \forall g_i \in PriOut(C) \\ & D(g_i) \geq 0, \forall g_i \in PriIn(C) \\ & vt(g_i) \in VT(g_i), \forall g_i. \end{array} \quad (1)$$

The timing constrained minimum leakage power problem is known to be NP-complete [3] by reducing it to the circuit implementation problem [15].

3. THE ALGORITHMIC FLOW

The proposed ϵ-approximation algorithm for discrete vt assignment is constructed using several advanced algorithmic techniques including potential function based approximation and dual binary-search-style decision core queries.

The discrete vt assignment problem in Eqn. (1) is first transformed to a cost (leakage power) minimization integer linear programming (ILP) problem in Eqn. (2). The ILP problem is then relaxed to a linear programming (LP) problem. In contrast to most approaches in the literature, the LP is not solved by existing linear programming solver. Rather, it is solved in a combinatorial way with bounded error. This also allows us to design an ϵ-approximation to the original vt assignment ILP problem.

In the approximation algorithm, the cost minimization ILP problem in Eqn. (2) will be reduced to a decision problem which says that given a cost budget, whether there is a solution satisfying the timing constraint within the cost budget. We call the algorithm to solve such a decision problem a *decision core*.

Given such a decision core, the overall algorithm is not hard to design. One can set an upper bound and a lower bound on the optimal leakage power consumption and perform a binary search within these bounds. Suppose that the ratio between the initial upper and lower bound is B. This will lead to $O(\log B/\epsilon)$ iterations. In this work, a dual binary-search-style decision core query procedure is used to improve the number of iterations to $O(\log\log B + \log 1/\epsilon)$, which is based on a logarithmic binary search proposed in [17].

4. ILP FORMULATION

The first step in our ϵ-approximation is to transform the discrete vt assignment problem into a cost minimization Integer Linear Programming (ILP) problem. Note that such a formulation has been designed for many times in the literature such as [8, 7]. However, none of them can solve the ILP in polynomial time with ϵ-bounded error, which is the main target of this paper.

For each gate node in the combinational circuit, associate with each available gate threshold voltage a *binary variable*. Suppose that there are m available threshold voltages for gate g. For j-th threshold voltage where $1 \le j \le m$, set a binary variable $x_{g,j}$ corresponding to it. $x_{g,j} = 1$ means that j-th threshold voltage is used and $x_{g,j} = 0$ means that j-th threshold voltage is not used. Since there is exactly one threshold voltage picked at each gate node in a vt assignment solution, $\sum_j x_{g,j} = 1$ for each gate node g. For j-th threshold voltage of gate node g, denote by $d_{g,j}$ the gate delay and $p_{g,j}$ the gate leakage power. As mentioned in Section 2, these values can be easily computed offline. To compute the gate delay, since all x are binary variables, $\sum_j d_{g_i,j}x_{g_i,j}$ gives the delay for gate g_i since only one of $x_{g_i,j}$ will be 1 and all other $x_{g_i,\cdot}$ will be 0. Similarly, the total leakage power of a solution is easy to compute, which is $\sum_{g_i} x_{g_i,j} \cdot p_{g_i,j}$. The integer linear programming formulation for the discrete vt assignment problem is as follows.

$$\begin{array}{ll} \min & \sum_{g_i} x_{g_i,j} \cdot p_{g_i,j} \\ s.t. & \\ & D(g_i) + c_{i,j} + \sum_j d_{g_i,j}x_{g_i,j} \le D(g_j), \forall g_i \in fin(g_j) \\ & D(g_i) \le T, \forall g_i \in PriOut(C) \\ & D(g_i) \ge 0, \forall g_i \in PriIn(C) \\ & \sum_j x_{g,j} = 1, \forall g, \\ & x_{g,j} \in \{0,1\}, \forall x_{g,j}. \end{array} \quad (2)$$

5. EPSILON-APPROXIMATION

5.1 LP Relaxation

To design an ϵ-approximation algorithm to the ILP in Eqn. (2), first relax the last set of constraints to form an LP as follows.

$$\begin{array}{ll} \min & \sum_{g_i} x_{g_i,j} \cdot p_{g_i,j} \\ s.t. & \\ & D(g_i) + c_{i,j} + \sum_j d_{g_i,j}x_{g_i,j} \le D(g_j), \forall g_i \in fin(g_j) \\ & D(g_i) \le T, \forall g_i \in PriOut(C) \\ & D(g_i) \ge 0, \forall g_i \in PriIn(C) \\ & \sum_j x_{g,j} = 1, \forall g \\ & x_{g,j} \in [0,1], \forall x_{g,j}. \end{array} \quad (3)$$

Denote by H the hypercube defined by the last constraint.

It is well known that the linear programming problem can be solved in polynomial time using ellipsoidal method or interior point method. However, this will not allow us to solve or approximately solve the original integer linear program. Therefore, an advanced algorithm proposed in [16] is used in this paper to approximate the optimal solution of the LP since it will allow us to compute an ϵ-approximation to our vt assignment ILP formulation. The approach is a potential function based technique and it [16] requires that the linear program is a convex min-max resource-sharing LP. Formally, a convex min-max resource-sharing LP is defined as a linear program with minimization objective in the form as Eqn. (4) where e is the vector of all ones and f is a non-negative linear convex function defined on a non-empty convex compact set L [16]

$$\min\{\lambda | f(x) \le \lambda e, x \in L\}. \quad (4)$$

Note that the formulation in Eqn. (3) is not a convex min-max resource-sharing LP. However, one can still transform it into a a convex min-max resource-sharing LP by exploring the specific structure of the discrete vt assignment problem. For this, one can treat $\mathbf{x}$ as variables and $D(g_i)$ as constants. This is valid since $D(g_i)$ can be easily computed by evaluating the delay for the vt assignment solution when $\mathbf{x}$ is given. The details are elaborated as follows.

5.2 Decision Problem

We first transform the optimization problem in Eqn. (3) into a decision problem. For this, the objective is cast into a constraint as $\sum_{g_i} x_{g_i,j} \cdot p_{g_i,j} \le P$ for a leakage power value P in the LP, i.e., to decide whether there is a feasible vt assignment solution such that the total leakage power is no greater than P. We have

$$\begin{array}{ll}\min & 1\\ s.t. & \\ & \sum_{g_i} x_{g_i,j} \cdot p_{g_i,j} \le P,\\ & D(g_i) + c_{i,j} + \sum_j d_{g_i,j} x_{g_i,j} \le D(g_j), \forall g_i \in fin(g_j)\\ & D(g_i) \le T, \forall g_i \in PriOut(C)\\ & D(g_i) \ge 0, \forall g_i \in PriIn(C)\\ & \sum_j x_{g,j} = 1, \forall g\\ & x_{g,j} \in [0,1], \forall x_{g,j}.\end{array} \quad (5)$$

Instead of directly solving this decision LP, we will transform it into a new optimization problem and approximate it. Subtracting some LHS (Left-Hand Side) by some RHS (Right-Hand Side) and adding 1 to both LHS and RHS, we get

$$\begin{array}{ll}\min & 1\\ s.t. & \\ & \sum_{g_i} x_{g_i,j} \cdot p_{g_i,j} - P + 1 \le 1,\\ & D(g_i) + c_{i,j} + \sum_j d_{g_i,j} x_{g_i,j} - D(g_j) + 1 \le 1, \forall g_i \in fin(g_j)\\ & D(g_i) - T + 1 \le 1, \forall g_i \in PriOut(C)\\ & D(g_i) \ge 0, \forall g_i \in PriIn(C)\\ & \sum_j x_{g,j} = 1, \forall g\\ & x_{g,j} \in [0,1], \forall x_{g,j}.\end{array} \quad (6)$$

Note that we will now treat only $\mathbf{x}$ as variables and all other symbols are constants. This is valid since once $\mathbf{x}$ are known, D can be easily computed by evaluating the delay at each gate node in the combinational circuit. λ is introduced to relax the inequality constraints in the above LP to obtain a new optimization problem as follows.

$$\begin{array}{ll}\min & \lambda\\ s.t. & \\ & \sum_{g_i} x_{g_i,j} \cdot p_{g_i,j} - P + 1 \le \lambda,\\ & D(g_i) + c_{i,j} + \sum_j d_{g_i,j} x_{g_i,j} - D(g_j) + 1 \le \lambda, \forall g_i \in fin(g_j)\\ & D(g_i) - T + 1 \le \lambda, \forall g_i \in PriOut(C)\\ & D(g_i) \ge 0, \forall g_i \in PriIn(C)\\ & \sum_j x_{g,j} = 1, \forall g,\\ & x_{g,j} \in [0,1], \forall x_{g,j}.\end{array} \quad (7)$$

To solve the decision problem in Eqn. (6), we need to know whether the minimum λ in Eqn. (7) can be ≤ 1. If this happens, then $\sum_{g_i} x_{g_i,j} \cdot p_{g_i,j} - P \le 0$ and thus the total cost is no greater than P. Similarly, $D(g_i) - T \le 0$, and thus the timing constraint is satisfied. Since all D are obtained by evaluating the timing in the combinational circuit, there is at least one $D(g_i) + c_{i,j} + \sum_j d_{g_i,j} x_{g_i,j} - D(g_j)$ will evaluate to 0. This means that at least one of $D(g_i) + c_{i,j} + \sum_j d_{g_i,j} x_{g_i,j} - D(g_j) + 1$ cannot be smaller than 1. Thus, $\lambda \ge 1$. Note that this LP will not be solved directly by existing linear programming solver, rather, it will be approximately solved by an advanced combinatorial algorithm proposed in [16] in mathematical programming literature. This will also allow us to approximately solve the ILP problem formed by replacing the last constraint in Eqn. (7) with $x_{g,j} \in \{0,1\}, \forall x_{g,j}$, which is the base of our decision core.

5.3 Potential Function

At a high level, the algorithm proceeds as follows. We maintain a set of parameters, namely, vector $\mathbf{x}$, vector $\mathbf{q}$ and scalar θ. Start with an initial $\mathbf{q}, \theta$ and compute the solution $\mathbf{x}$ by minimizing $\mathbf{q}^T f(\mathbf{x})$. Using this $\mathbf{x}$, new $\mathbf{q}, \theta$ are computed. This process is iterated until the θ is close to the optimal $\lambda(\mathbf{x})$ over all $\mathbf{x} \in H$. There is an algorithm proposed in [16] which runs in only $O(J \ln J + \frac{J \ln \frac{1}{\epsilon}}{\epsilon^2})$ iterations in order to obtain an ϵ-approximation to the vt assignment LP, where J is the number of inequality constraints in the LP formulation in Eqn. (7).

We first describe an important concept called *potential function* in the algorithm. Given J inequality constraints in the LP, denote by $f_j(\mathbf{x})$ the LHS of each inequality constraint $j = 1, 2, \ldots, J$ in the LP. Denote by $\lambda(\mathbf{x})$ the objective value in Eqn. (7) with $\mathbf{x}$. Suppose that we want to approximate the solution with the approximation error bounded by ϵ. The potential function defined in [16] is

$$\phi_\epsilon(\mathbf{x}) = \min_{\lambda(\mathbf{x}) < \theta < \infty} (\ln \theta - \tfrac{\epsilon}{J} \textstyle\sum_{j=1}^{J} \ln(\theta - f_j(\mathbf{x}))), \quad (8)$$

where $\mathbf{x}$ are the variables in Eqn. (7) and θ is an additional variable introduced for performing approximation. It can be shown that $\phi_\epsilon(\mathbf{x})$ is achieved when [16]

$$\tfrac{\epsilon}{J} \textstyle\sum_{j=1}^{J} \frac{\theta}{\theta - f_j(\mathbf{x})} = 1. \quad (9)$$

Since $\frac{\epsilon}{J} \sum_{j=1}^{J} \frac{\theta}{\theta - f_j(\mathbf{x})}$ is strictly decreasing for $\theta > \lambda(\mathbf{x})$, θ can be computed by finding the unique root in that range. [16] shows that θ is an ϵ-approximation to the optimal value of objective $\lambda(\mathbf{x})$ since [16]

$$\frac{\lambda(\mathbf{x})}{1 - \epsilon/J} \le \theta(\mathbf{x}) \le \frac{\lambda(\mathbf{x})}{1 - \epsilon}, \forall \mathbf{x} \in H.$$

As a result, it can be shown that the potential function $\phi_\epsilon(\mathbf{x})$ is an approximation to $\ln \lambda(\mathbf{x})$. Further, in each iteration of updating $\mathbf{x}$, θ, and $\mathbf{q}$, $\phi_\epsilon(\mathbf{x})$ is always reduced by a constant. Consequently, $\phi_\epsilon(\mathbf{x})$ will converge to the logarithmic of the optimal λ, and θ will converge to the optimal value. In addition, this can be achieved in $O(J \ln J + \frac{J \ln \frac{1}{\epsilon}}{\epsilon^2})$ iterations. Refer to [16] for the proof.

5.4 Approximation

For any approximation error bound ϵ and $\mathbf{x}$, one can always compute θ by Eqn. (9) and $\phi_\epsilon(\mathbf{x})$ by Eqn. (8) to approximate $\lambda(\mathbf{x})$. However, we need to compute the optimal $\lambda(\mathbf{x})$ over all possible $\mathbf{x} \in H$ (recall that hypercube H is defined by the last constraint in Eqn. (3)). Thus, we need to find a good $\mathbf{x}$. For this, a vector $\mathbf{q} = \{q_1, q_2, \ldots, q_J\}$ is associated with inequality constraints in Eqn (7) as in [16] where

$$q_j(\mathbf{x}) = \tfrac{\epsilon}{J} \tfrac{\theta}{\theta - f_j(\mathbf{x})}. \quad (10)$$

Given an θ, it is easy to compute all q_j for $j = 1, 2, \ldots, J$. The new $\mathbf{x}$ is computed such that $\mathbf{q}^T f(\mathbf{x})$ is minimized. In contrast to [16] where this minimization problem is approximated, in our context, the minimum can be computed by exploring the specific structure of Eqn. (7).

Note that our problem is defined on the hypercube H, which is a convex region. This means that for any positive constants $\mathbf{q}$, the minimum value of $\mathbf{q}^T f(\mathbf{x})$ is achieved only at some vertices in the hypercube. That is, for each variable $x_{g_i,j}$, the optimal value is achieved when it is either 0 or 1. As a result, the optimal solution is an integer solution. Thus, the approximation algorithm is also an approximation to the ILP defined in Eqn. (2). To compute the minimum value, one first computes the coefficient for each $x_{g_i,j}$. For each g_i, find the $x_{g_i,j}$ with the smallest coefficient and set it to 1 and others to 0. Denote the above algorithm by $MINPF(\mathbf{q})$. This procedure returns a solution where all $\mathbf{x}$

are (binary) integers, which is needed in the decision core in Section 5.5. Given J constraints, n edges (which upper bounds the number of gates) and m threshold voltages at each gate, $MINPF(\mathbf{q})$ takes $O(Jmn)$ time.

Note that instead of directly approximating the solution with ϵ, we gradually reduce the approximation bound in an iterative manner to obtain a smooth approximation. In s-th iteration, denote by σ_s the approximation bound and set $t = \sigma_s/6$ [16]. Note that ϵ is replaced with t in Eqn. (9), Eqn. (10), and Eqn. (13).

Finally, we are to describe the two stopping criteria, proposed in [16], for the above iterative algorithm. The correlation between the current $\mathbf{x}$, denoted by $\widehat{\mathbf{x}}$, and the $\mathbf{x}$, computed in the previous iteration is defined as follows.

$$v(\mathbf{x}, \widehat{\mathbf{x}}) = \frac{q^T f(\mathbf{x}) - q^T f(\widehat{\mathbf{x}})}{q^T f(\mathbf{x}) + q^T f(\widehat{\mathbf{x}})}. \tag{11}$$

Clearly, the smaller v, the closer $\mathbf{x}$ is to $\widehat{\mathbf{x}}$. This means that no much improvement has been made. In this case, we conclude that the solution already converges to the optimal solution. Precisely, the iteration stops when $v \leq \sigma/6$ where σ is the current approximation error bound (gradually approaching ϵ). The other stopping criterion to further speedup the algorithm in [16] is as follows. Denote by w_s in an iteration s as

$$\begin{array}{ll} w_s & = \frac{1+\sigma_1}{(1+\sigma_1/3)J}, s = 1 \\ w_s & = \frac{1+\sigma_s}{1+2\sigma_s}, otherwise. \end{array} \tag{12}$$

This criterion is $\lambda(\mathbf{x}) \leq w_s \lambda(\mathbf{x}_{s-1})$. It means that if the objective value does not change significantly, the solution also converges to the optimal solution. Define τ as

$$\tau = \frac{\epsilon\theta}{2J(p^T f(\mathbf{x}) + p^T f(\widehat{\mathbf{x}}))}. \tag{13}$$

$\mathbf{x}$ will be updated as $(1-\tau)\mathbf{x} + \tau\widehat{\mathbf{x}}$ [16]. This $\mathbf{x}$ is used to approximate the non-integer minimum in the LP. However, since $\mathbf{x}$ and $\widehat{\mathbf{x}}$ are very close to each other during iterations, it suffices to return the integer solution $\widehat{\mathbf{x}}$ as the approximate solution.

5.5 Decision Core

We have obtained an ϵ-approximation algorithm for solving Eqn. (7). It also allows us to solve the ILP problem corresponding to it (formed by replacing the last constraint in Eqn. (7) with $x_{g,j} \in \{0,1\}, \forall x_{g,j}$) since the algorithm always computes a binary integer solution. The original vt assignment LP in Eqn. (3) can be solved as follows. Denote by P^* the optimal solution to the vt assignment ILP in Eqn. (2). Set an upper bound $\overline{P}$ and a lower bound $\underline{P}$ on P^*. Denote by B the ratio between the upper and lower bounds. One can perform a binary search within these bounds, which needs in $O(\log B/\epsilon)$ iterations.

In this work, we are to perform a *dual binary-search-style decision core query*. The first step is to perform a logarithmic scale binary search within these bounds which will need only $O(\log\log B + \log 1/\epsilon)$ iterations. This logarithmic scale binary search is proposed in [17] for a restricted shortest path problem and is adopted in [18] for a layer assignment problem. For this, set $P = \sqrt{\overline{P} \cdot \underline{P}}$, and solve the above minimization problem in Eqn. (7) with this P. There are three cases. (1) $\lambda = 1$. Since all x are binary numbers as computed above (by $MINPF$ procedure in Section 5.4), $\lambda = 1$ means that $\sum_{g_i} x_{g_i,j} \cdot p_{g_i,j} - P + 1 \leq 1$, $D(g_i) - T + 1 \leq 1$, and $D(g_i) + c_{i,j} + \sum_j d_{g_i,j} x_{g_i,j} - D(g_j) + 1 \leq 1$. Thus, all the constraints are satisfied. It means that there is a vt assignment solution with the cost P satisfying T. (2) $\lambda > 1 + \epsilon$. In this case, it means that there exists a constraint which cannot be evaluated to 1 in the optimal solution. Thus, there is no vt assignment solution with the cost P satisfying T. (3) $1 < \lambda \leq 1 + \epsilon$. In this case, whether there is a feasible solution cannot be decided since the above algorithm only approximately solves the decision problem in Eqn. (7). We make a conservative decision, i.e., there is no vt assignment solution with the cost P satisfying T.

For Case (1), set the upper bound $\overline{P} = P$ and for Case (2) and (3), set the lower bound $\underline{P} = P$. The above process is then repeated until the ratio between the upper and the lower bounds is below 2. Since B is initially set to $\frac{\overline{P}}{\underline{P}}$ and $P = \sqrt{\overline{P} \cdot \underline{P}}$, it is easy to see that in either case of (upper or lower) bound updates, the ratio between new upper and lower bound is reduced $\sqrt{B}$ [17]. In general, after b iterations, the ratio is reduced to $B^{\frac{1}{2^b}}$. Set $B^{\frac{1}{2^b}} \leq 2$. We have $b = O(\log\log B)$. Thus, after $O(\log\log B)$ iterations, the ratio between the upper bound $\overline{P}$ and the lower bound $\underline{P}$ is no greater than 2.

With this better starting point, we then perform the standard binary search within the new bounds. For this, we set $P = \frac{\overline{P}+\underline{P}}{2}$ and perform the above process until the ratio between the bounds is below $(1+\epsilon)$. We then return the solution corresponding to the upper bound $\overline{P}$ as the final solution. It is easy to see that the total number of iterations in this (second) binary search is bounded by $O(\log 2/\epsilon)$ since the ratio between the initial upper and lower bounds is bounded by 2. Note that in this work, we assume that the initial ratio between the upper and lower bounds B is bounded by $O(m)$, which is the number of gate threshold voltage at each gate. Thus, the above algorithm needs $O(\log\log m + \log 1/\epsilon)$ iterations.

5.6 Time Complexity

In querying the decision core, each time a cost value P is specified. For each P, there are $O(J \ln J + \frac{J \ln \frac{1}{\epsilon}}{\epsilon^2})$ iterations. In each iteration, the most time consuming part is the query of $MINPF$, which takes $O(Jmn)$ time. To bound J, since the the combinational circuit has n edges, $J = O(n)$. Thus, for each cost value P, the runtime is $O((J \ln J + \frac{J \ln \frac{1}{\epsilon}}{\epsilon^2}) \cdot Jmn) = O(mn^3 \log n + \frac{mn^3 \log \frac{1}{\epsilon}}{\epsilon^2})$. Since the dual binary-search-style decision core query calls the decision core on $O(\log\log m + \log 1/\epsilon)$ different cost values P, the total runtime is

$$\begin{array}{ll} O & (mn^3 \log n \log\log m + \frac{mn^3 \log\log m \log \frac{1}{\epsilon}}{\epsilon^2} \\ & + mn^3 \log n \log \frac{1}{\epsilon} + \frac{mn^3 \log^2 \frac{1}{\epsilon}}{\epsilon^2}). \end{array} \tag{14}$$

Since for small ϵ, $\log\log m < \log \frac{1}{\epsilon}$, the above bound can be simplified to

$$O(mn^3 \log n \log \tfrac{1}{\epsilon} + \tfrac{mn^3 \log^2 \frac{1}{\epsilon}}{\epsilon^2}). \tag{15}$$

In summary, we have shown that an ϵ-approximation to the delay constrained minimum leakage power vt assignment problem can be computed in $O(mn^3 \log n \log \frac{1}{\epsilon} + \frac{mn^3 \log^2 \frac{1}{\epsilon}}{\epsilon^2})$ time for any $0 < \epsilon < 1$, where n is the number of edges in the combinational circuit and m is the number of threshold

voltage at each gate.

6. EXPERIMENTAL RESULTS

The proposed ϵ-approximation algorithm for the discrete vt assignment problem is implemented in C++ and tested on a Pentium IV machine with 2.4GHz CPU and 4GB memory. The experiments are performed to ISCAS'85 benchmark circuits. A $65nm$ gate library is used where each gate type has 4 available threshold voltages. The circuits are sized with all gate implementations fixed at lowest vt by a gate sizing algorithm [19] and are placed by mPL [20]. For the results in Table 2, the timings obtained in the above are relaxed (by about 10%) and vt assignment is performed.

We first compare the new algorithm to the integer linear programming (ILP) approach as in Eqn. (2). Such kind of ILP techniques have been designed for many times in the literature such as [8, 7]. Refer to Table 1 for the results on the smallest circuit C432. Note that in this experiment, we choose loose timing constraint since integer linear programming (ILP) can only be solved for small circuit with loose timing constraints but not with tight timing constraints. Consider C432 as an example. For the timing constraint set to 1500ps, it takes only 4.8 seconds to solve. For the timing constraint set to 1000ps, it takes 204.2 seconds to solve. For the timing constraint set to 800ps which is still not tight, ILP is computationally prohibitive (we stop ILP solver after 2 hours).

From Table 1, since ILP computes the optimal solution, one can see that the new ϵ-approximation algorithm always returns the solution within the target approximation ratio ϵ. In fact, the actual approximation ratio (leakage ratio) is much smaller than ϵ. For example, the solution returned by ϵ-approximation is only 2.2% off the optimal solution when $\epsilon = 5\%$. The optimal solution is the one obtained by ILP, however, it cannot run in polynomial time and is computationally prohibitive when handling large circuits or small circuits with tight timing constraints. In addition, the speedup obtained from our ϵ-approximation is large, often with orders of magnitude compared to ILP. Further, the runtime of ϵ-approximation decreases when the target approximation ratio ϵ increases, which is as desired. One can see from Figure 1 that different power and runtime tradeoff can be obtained.

Table 1: Comparison of the optimal ILP technique with our ϵ-approximation algorithm on C432 with a loose timing constraint $T = 1000$ps. For ILP solution, total power is 274 and CPU is 204.2 seconds. Leakage Ratio and Speedup are computed by comparing to ILP solution.

ϵ-approximation on C432 w/ $T = 1000$ps				
ϵ	Leakage Power	CPU(s)	Leakage Ratio	Speedup
0.05	280	2.6	2.2%	78.5×
0.10	291	2.0	6.2%	102.1×
0.20	307	1.9	12.0%	107.5×
0.30	316	1.6	15.3%	127.6×
0.40	341	1.3	24.5%	157.1×
0.50	367	1.2	33.9%	170.2×
Average				123.8×

We next demonstrate the performance of the new ϵ approximation algorithm by comparing to the heuristic in [1]

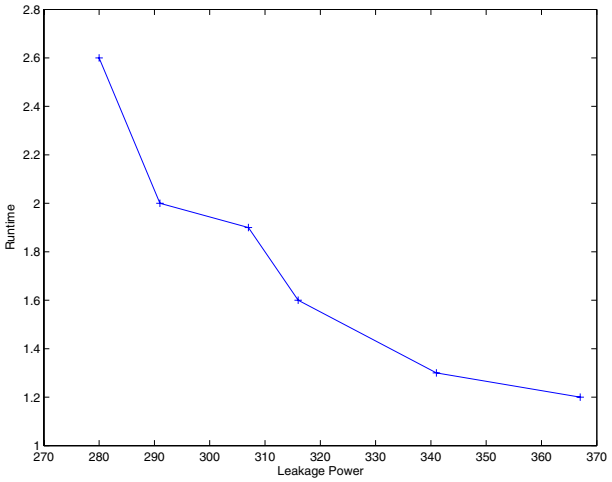

Figure 1: Leakage Power v.s. CPU time for C432.

for vt assignment. The experiments are performed on the remaining ISCAS'85 benchmark circuits with tight timing constraints. Note that tight timing constraints are important since this is why we need vt assignment for timing closure especially in late design stage. In the experiments, various ϵ has been tested. Due to the space limitation, we choose to present the results with $\epsilon = 0.05$ and $\epsilon = 0.1$. They are the desired ϵ since in practice we would like to compute the solutions close to optimum with small runtime. Refer to Table 2 for the results. We make the following observations.

- In our experiments, tight timing constraints are set on each benchmark circuits. Thus, the ILP formulated in Eqn. (2) is computationally prohibitive.

- The ϵ-approximation can compute the solution efficiently. Our algorithm can finish in several minutes for handling the largest ISCAS'85 benchmark circuit. Compared to the heuristic in [1], our ϵ-approximation algorithm significantly saves the leakage power. For example, when $\epsilon = 0.05$, the average leakage power saving can be is 16.5%. The runtime of our algorithm is similar to the [1].

- Since ILP is computational prohibitive, ϵ-approximation is compared to linear programming (LP) in Eqn. (3) which relaxes the integer constraints of the ILP in Eqn. (2). Although LP approach can be efficiently computed by linear programming solver, it cannot generate feasible discrete vt assignment solutions since the solution has fractional vt assignment. We can use the LP solution serves as a lower bound for the optimal integer linear programming solution. It is clear that the results of ϵ-approximation are always within the guaranteed approximation error bound even compared to the lower bound. This means that the results are also within the approximation bounds compared to the optimal discrete vt assignment solutions. For example, for $\epsilon = 0.05$, on average ϵ-approximation can approximate the optimal solution (even when lower bound on the optimal solution is considered) with the factor of 2.8% and run in 51.3 seconds.

Table 2: Comparison of [1] and ϵ-approximation on ISCAS'85 benchmark circuits with tight timing constraints. Since ILP cannot be solved in a reasonable time with tight timing constraints, the results of LP relaxation are also shown which give the lower bound for the optimal vt assignment.

Circuit	Lower Bound	Heuristic [1]		New ϵ-approximation (with $\epsilon = 5\%$)				New ϵ-approximation (with $\epsilon = 10\%$)		
	Leakage	Leakage	CPU(s)	Leakage	CPU(s)	Leakage Ratio	Leakage Saving	Leakage	CPU(s)	Leakage Ratio
C499	552	657	4.8	565	6.8	2.4%	16.3%	586	5.7	6.2%
C880	683	832	3.5	695	4.5	1.8%	19.7%	734	3.7	7.5%
C1355	721	876	4.1	748	5.2	3.7%	17.1%	780	4.0	8.2%
C1908	914	1105	10.5	934	15.4	2.2%	18.3%	975	12.4	6.7%
C2670	1165	1473	16.8	1219	20.1	4.6%	20.8%	1247	17.3	7.0%
C3540	1258	1512	23.2	1285	42.9	2.2%	17.7%	1363	35.2	8.4%
C5315	2437	2785	50.6	2480	93.7	1.8%	12.3%	2534	74.5	4.0%
C6288	4590	5278	75.0	4776	121.3	4.1%	10.5%	4842	97.4	5.5%
C7552	2872	3409	87.9	2952	152.2	2.8%	15.5%	3012	127.6	4.9%
Average			30.7		51.3	2.8%	16.5%		42.0	6.5%

- Comparing ϵ-approximation with different target approximation ratios, one clearly sees the solution quality and runtime tradeoff, which is also desired.

7. CONCLUSION

Discrete vt assignment is a critical technique in timing and leakage power optimization. Despite numerous prior studies, this NP-complete problem is still not well understood especially from theoretical point of view. This work designs the first ϵ-approximation algorithm running in $O(mn^3 \log n \log \frac{1}{\epsilon} + \frac{mn^2 \log^2 \frac{1}{\epsilon}}{\epsilon^2})$ time, where n is the size of combinational circuit and m is the number of available threshold voltages at each gate. Our experiments on ISCAS'85 benchmark circuits demonstrate that the new algorithm always returns a solution with error bounded by ϵ and can significantly outperforms the vt assignment heuristic in [1] by 16.5% leakage power saving with similar runtime.

8. REFERENCES

[1] L. Wei, Z. Chen, M. Johnson, K. Roy, and V. De, "Design and optimization of low voltage high performance dual threshold cmos circuits," *in DAC*, pp. 489–494, 1998.

[2] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-v power supply high-speed digital circuit technology withmultithreshold-voltage cmos," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 847 – 854, 1995.

[3] V. Sundararajan and K. Parhi, "Low power synthesis of dual threshold voltage cmos vlsi circuits," *in ISLPED*, pp. 139 – 144, 1999.

[4] P. Pant, R. Roy, and A. Chatterjee, "Dual-threshold voltage assignment with transistor sizing for low power cmos circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 2, pp. 390 – 394, 2001.

[5] 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 DAC*, pp. 486 – 491, 2002.

[6] D. Lee and D. Blaauw, "Static leakage reduction through simultaneous threshold voltage and state assignment," *in DAC*, pp. 191–194, 2003.

[7] D. Nguyen, A. Davare, 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 ISLPED*, pp. 158 – 163, 2003.

[8] F. Gao and J. Hayes, "Total power reduction in cmos circuits via gate sizing and multiple threshold voltages," *in DAC*, pp. 31 – 36, 2005.

[9] H. Chou, Y.-H. Wang, and C.-P. Chen, "Fast and effective gate-sizing with multiple-vt assignment using generalized lagrangian relaxation," *in ASPDAC*, pp. 381 – 386, 2005.

[10] S. Shah, A. Srivastava, D. Sharma, D. Sylvester, D. Blaauw, and V. Zolotov, "Discrete vt assignment and gate sizing using a self-snapping continuous formulation," *in DAC*, pp. 705–712, 2005.

[11] V. Khandelwal, A. Davoodi, and A. Srivastava, "Simultaneous vt selection and assignment for leakage optimization," *IEEE Transactions on VLSI Systems*, vol. 13, no. 6, pp. 762–765, 2005.

[12] T.-H. Wu, L. Xie, and A. Davoodi, "A parallel and randomized algorithm for large-scale dual-vt assignment and continuous gate sizing," *in ISLPED*, pp. 45–50, 2008.

[13] T. Karnik, S. Borkar, and V. De, "Sub-90nm technologies: challenges and opportunities for cad," *in ICCAD*, pp. 203 – 206, 2002.

[14] J.P. Fishburn and A.E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing,", *in ICCAD*, pp. 326–328, 1985.

[15] W.-N. Li, A. Lim, P. Agrawal, and S. Sahni, "On the circuit implementation problem," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 8, pp. 1147 – 1156, 1993.

[16] K. Jansen and H. Zhang, "Approximation algorithms for general packing problems and their application to the multicast congestion problem," *Mathematical Programming*, vol. 114, no. 1, pp. 183–206, 2008.

[17] R. Hassin, "Approximation schemes for the restricted shortest path problem," *Mathematics of Operations Research*, vol. 17, no. 1, pp. 36–42, 1992.

[18] S. Hu, Z. Li, and C. Alpert, "A polynomial time approximation scheme for timing constrained minimum cost layer assignment," *in ICCAD*, 2008.

[19] S. Hu, M. Ketkar, and J. Hu, "Gate Sizing For Cell Library-Based Designs", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 6, pp. 818–825, 2009.

[20] CPMO – Constrained Placement by Multilevel Optimization, *http://cadlab.cs.ucla.edu/cpmo/*