Network Flow-based Simultaneous Retiming and Slack Budgeting for Low Power Design

Bei Yu∗, Sheqin Dong∗, Yuchun Ma∗, Tao Lin∗, Yu Wang∗, Song Chen† and Satoshi Goto†
∗Department of Computer Science & Technology, Tsinghua University, Beijing, China
†Graduate School of IPS, Waseda University, Kitakyushu, Japan

Abstract—Low power design has become one of the most significant requirements when CMOS technology entered the nanometer era. Therefore, timing budget is often performed to slow down as many components as possible so that timing slacks can be applied to reduce the power consumption while maintaining the performance of the whole design. Retiming is a procedure that involves the relocation of flip-flops (FFs) across logic gates to achieve faster clocking speed. In this paper we show that the retiming and slack budgeting problem can be formulated to a convex cost dual network flow problem. Both the theoretical analysis and experimental results show the efficiency of our approach which can not only reduce power consumption but also speedup previous work.

1. INTRODUCTION

Timing constraint design and low power design have become significant requirements when the CMOS technology entered the nanometer era. On the one hand, more and more devices trend to be put in the small silicon area while at the same time the clock frequency is pushed even higher. As an effective timing optimization scheme, retiming is a procedure that involves the relocation of flip-flops (FFs) across logic gates to achieve faster clocking period. On the other hand, to tackle the tremendous growth in the design complexity, timing budgeting is performed to relax the timing constraints for as many components as possible without violating the system’s timing constraint. Therefore, both retiming and timing budget might influence the timing distribution of the design greatly.

Since Leiserson and Saxe proposed the idea of retiming in 1983 [1], it has become one of the most powerful sequential optimization techniques. In [2], the min-area retiming problem was solved by min-cost network flow algorithm. Recent publications [3] and [4] proposed a very efficient retiming algorithm for minimal period by algorithm derivation. [5] and [6] respectively presented efficient incremental algorithms for min-period retiming under setup and hold constraints, and min-area retiming under given clock period.

For timing-constrained gate-level synthesis, timing slack is an effective method for circuit’s potential performance improvement. The components with relaxed timing constraints can be further optimized to improve system’s area, power dissipation, or other design quality metrics. The slack budgeting problem has been studied well. Some of the previous slack budgeting approaches are suboptimal heuristics such as Zero-Slack Algorithm (ZSA) [7]. [8][9] formulated the slack budgeting problem as Maximum-Independent-Set (MIS) on sensitive transitive closure graph. In [10] and [11], authors proposed combinatorial methods based on net flow approach to handle the slack budget problem.

Fig. 1: Relocate FFs to increase potential slack without violating timing constraint. (a) No potential slack in this circuit. (b) moving the FF from edge de to edge cd, the potential slack can be increased from 0 to 3.
the potential slack by retiming for synchronous sequential circuit. They proposed a reasonable algorithm flow, however, their solution quality suffers in two aspects. First, there was no guarantee that the algorithm will get optimal solution because iterative strategy is easily trapped in local optimum. Besides, the slack budget problem was translated to a Maximal Independent Set (MIS) problem, which is a NP-hard problem.

[19] showed that for an Integer Linear Programming (ILP) with separable convex objective functions and special form of constraints, it can be viewed as convex cost dual network flow problem and solved in polynomial time. This model has been adopted in various works, such as buffer insertion [20], multi-voltage supply [21][22], clock skew scheduling [23] and slack budgeting [17].

In this paper we first formulate retiming and slack budgeting problem as an Integer Linear Programming (ILP) problem. Since ILP has been listed to be one of the known NP-hard problems, we then show how to transform this problem to the convex cost dual network flow problem with just a little loss of optimality. Experimental results show that our algorithm can not only reduce power consumption, increase total slack constraints, but also effectively speedup previous work.

The remainder of this paper is organized as follows. Section 2 defines the simultaneous slack budget and retiming problem. Since ILP has been listed to be one of the known NP-hard problems, we then show how to transform this problem to the convex cost dual network flow problem. The clock period is given as $T$. For each vertex, three non-negative labels, $a_i/\gamma_i/s_i$, represent the latest arrival time, require time, and slack of vertex $i$, respectively. Then slack $s_i$ is calculated as follows:

$$\begin{align*}
\{ a_i &= d_i \quad \text{if} \quad w(k,i) > 0 \quad \text{or} \quad i \in PI \\
\{ a_i &= max_j(a_j + d_j) \quad \forall j \in FI(i) \\
\{ \gamma_i &= T \quad \text{if} \quad w(k,i) > 0 \quad \text{or} \quad i \in PO \\
\{ \gamma_i &= min_j(\gamma_j - d_j) \quad \forall j \in FI(i)
\end{align*}$$

where $PI$ is set of all primary inputs and $PO$ is set of all primary outputs. $FI(i)$ and $PO(i)$ represent the incoming and outgoing gates to gate $i$ respectively. Then slack $s_i$ is then calculated by

$$s_i = \gamma_i - a_i$$

A retiming of a circuit $G$ is an integer-valued vertex-labeling $r$, which represent how many FFs are moved from the outgoing edges to the incoming edges of each vertex. Thus the number of FFs on edge $(i,j)$ with label $r$ is formulated as follow:

$$w_{i,j} + r_j - r_i$$

### Definition 1: Power Slack Curve
Each gate $i$ is given $k$ discrete slack levels, and the power-slab tradeoff is represented by $\{(s_i^1, P(s_i^1)), \ldots, (s_i^k, P(s_i^k))\}$. In the Power Slack Curve, each point is connected to its neighboring point(s) by a linear segment.

Based on the relationship between power reduction and slack provided by [15], we assume Power Slack Curve is a convex decreasing function.

### Definition 2: Simultaneous Slack Budget and Retiming Problem
Given a directed graph $G = (V, E, d, w)$ representing a synchronous sequential circuit, and period constraint $T$, we want to find FFs reallocation represented by $r$, such that the power consumption obtained by slack budgeting is minimized under the period constraint.

According to the above definitions and notations, the simultaneous slack budget and retiming problem can be easily formulated into the following mathematical program:

$$\begin{align*}
\min & \sum_{i \in V} P(s_i) \\
\text{s.t.} & (1) - (3) \\
& r_j - r_i \geq -w_{i,j} \quad \forall (i,j) \in E \\
& s_i \in \{ s_i^1, \ldots, s_i^k \} \quad \forall i \in V \\
& a_i \leq T \quad \forall i \in V
\end{align*}$$

### 3. Methodology

#### 3.1. MILP Formulation

The MILP formulation for retiming synchronous circuits is originally presented in [1] to minimize clock period. The clock period $\Phi(G) \leq T$ if and only if there exists an assignment of real values $a_i$ and an integer value $r_i$ to each vertex $i \in V$ such that the following conditions are satisfied:

$$\begin{align*}
a_i &\geq d_i + s_i \quad \forall i \in V \\
a_i &\leq T \quad \forall i \in V \\
r_i - r_j &\leq w_{i,j} \quad \forall (i,j) \in E \\
a_j &\geq a_i + d_i + s_i \quad \text{if} \quad r_i - r_j = w_{i,j}
\end{align*}$$

Suppose $R_i = r_i + a_i/T$, then $a_i = T \cdot R_i - T \cdot r_i$. The problem can be formulated as (II).

$$\begin{align*}
\min & \sum_{i \in V} P(\bar{s}_i) \\
\text{s.t.} & (I) \text{a} \\
& \bar{R}_i - \bar{r}_i \geq \bar{s}_i \quad \forall i \in V \\
& \bar{R}_i - \bar{r}_i \leq T \quad \forall i \in V \\
& \bar{r}_j - \bar{r}_i \geq -T \cdot w_{i,j} \quad \forall (i,j) \in E \\
& 0 \leq \bar{R}_i, \bar{r}_i \leq \bar{N}_{ff} \quad \forall (i,j) \in E \\
& \bar{s}_i \in \{ \bar{s}_1^1, \ldots, \bar{s}_1^k \} \quad \forall i \in V \\
& 0 \leq \bar{s}_i \leq T \quad \forall i \in V \\
& \bar{R}_j - \bar{r}_i \geq t_{ij} \quad \forall (i,j) \in E \\
& t_{ij} = \bar{s}_i + d_i \quad (j = 1, \ldots, k).\end{align*}$$

where $\bar{N}_{ff} = N_{ff} \cdot T$, $\bar{s}_i = d_i + s_i$, $\bar{r}_i = r_i \cdot T$ and $\bar{R}_i = R_i \cdot T$.
This problem can be solved by common ILP solver. However, computationally ILP is one of the most difficult combinatorial optimization problems and the runtime is unacceptable even if the problem size is small. In the following subsections, we will explain how to transform this problem to a convex cost dual network flow problem.

3.2. Formulation Simplification

Constraint (IIIh) make problem (II) too complex to solve by network flow-based algorithm. First we consider a more simple formulation (III), which removes constraint (IIIh). To compensate the lose of accuracy, we add penalty function $P(t_{ij})$ in objective function.

\[
\begin{align*}
\min & \sum_{i \in V} P(\bar{s}_i) + \sum_{(i,j) \in E} P(t_{ij}) \\
\text{s.t.} & \quad (IIa) - (IIg) \\
& \quad t_{ij} \geq -T \cdot w_{ij}, \quad \forall (i,j) \in E
\end{align*}
\]

where $P(t_{ij}) = P(\bar{s}_j)/k$, and $k$ is a coefficient. Here we set $k = \sum (1 - w_{ij})^4$.

Given solution of problem (III) $\bar{s}_i (i = 1, \ldots, m)$ and $t_{ij} (\forall (i,j) \in E)$, we propose a heuristic method to generate solution of problem (II).

\[t_{ij} \geq \bar{s}_j - T \cdot w_{ij} \Rightarrow \bar{s}_j = \min(t_{ij} + T \cdot w_{ij}), \forall i \in FI(j)\]

We denote the $\bar{s}_j$ got in (8) as $\bar{s}_j(\Omega)$ and $\bar{s}_j$ got from problem (III) as $\bar{s}_j(\Theta)$, then we can get $\bar{s}_j$ in problem (II) as follows:

\[
\bar{s}_j = \min[\bar{s}_j(\Omega), \bar{s}_j(\Theta)] = \min[\min(t_{ij} + T \cdot w_{ij}), \bar{s}_j(\Theta)], \forall i \in FI(j)\]

By now we have build the connection between solution of problem (II) and problem (III). After we calculate the solution of (III), we can then get the solution of (II). In the next subsection, we will prove problem (III) can be transformed to convex cost dual network flow problem.

3.3. Remove Redundant Constraint

In this subsection we will prove that without loss of optimality, problem (III) can remove constraint $\bar{R}_i - \bar{r}_i \leq T$.

Let $s_i^*$ denote the value of $s_i$ for which $P(s_i)$ is minimum. In case there are multiple values for which $P(s_i)$ is minimum, the minimum value will be chosen. Let us define the function $Q(s_i)$ in the following manner:

\[
Q(s_i) = \begin{cases} 
P(s_i) & \text{if } s_i \leq s_i^* \\
\bar{P}(s_i) & \text{if } s_i > s_i^*
\end{cases}
\]

Now consider the following problem (III′), which replaces (IIa) and (IIb) by $\bar{R}_i - \bar{r}_i = \bar{s}_i$;

\[
\begin{align*}
\min & \sum_{i \in V} Q(\bar{s}_i) + \sum_{(i,j) \in E} P(t_{ij}) \\
\text{s.t.} & \quad (IIc) - (IIg) \\
& \quad \bar{R}_i - \bar{r}_i = \bar{s}_i, \quad \forall i \in V \\
& \quad t_{ij} \geq -T \cdot w_{ij}, \quad \forall (i,j) \in E
\end{align*}
\]

1We suppose for each $(i,j) \in E$, $w_{ij}$ is $0 - 1$ variable.

![Fig. 2: (a)The DAG $G$ representing a synchronous sequential circuit. (b)The transformed DAG $G$ of $G$.](image)

**Theorem 1:** For every optimal solution $(\hat{R}, \hat{r}, \hat{s})$ of problem (III), there is an optimal solution $(\bar{R}, \bar{r}, \bar{s})$ of problem (III′), and the converse also holds.

*Proof:* Consider an optimal solution $(\hat{R}, \hat{r}, \hat{s})$ of (III), we show how to construct an optimal solution $(\bar{R}, \bar{r}, \bar{s})$ of (III′) with the same cost. There are two cases to consider:

**Case 1:** $\bar{R}_i - \bar{r}_i \geq s_i^*$. It follows from (IIa) and the convexity of $P(s_i)$ that $\bar{s}_i = s_i^*$. In this case, we set $\bar{s} = \bar{R}_i - \bar{r}_i$. It follows from (10) that $P(\bar{s}_i) = Q(\bar{s}_i)$.

**Case 2:** $\bar{R}_i - \bar{r}_i < s_i^*$. Similar to case 1, we can get $\bar{s}_i = \bar{R}_i - \bar{r}_i$. In this case, we set $\bar{s}_i = \bar{R}_i - \bar{r}_i$. It follows from (10) that $P(\bar{s}_i) = Q(\bar{s}_i)$.

Similarly, it can be shown that if $(\hat{R}, \hat{r}, \hat{s})$ is an optimal solution of (III′), then the solution $(\bar{R}, \bar{r}, \bar{s})$ constructed in the following manner is an optimal solution of (III): $\bar{s}_i = \max\{s_i^*, \bar{s}_i\}$.

**Theorem 2:** The constraint $\bar{R}_i - \bar{r}_i \leq T$ in problem (III) can be removed.

*Proof:* By Theorem 1, we can transform each constraint in (IIa) to an equality constraint. In other words, $\bar{R}_i - \bar{r}_i = \bar{s}_i$. Because constraint (IIf) $0 \leq \bar{s}_i \leq T$, $\bar{R}_i - \bar{r}_i \leq T$. So we can remove constraint $\bar{R}_i - \bar{r}_i \leq T$.

3.4. Transformation to Primal Network Flow Problem

To further simplify problem (III), we transform $G(V,E)$ into $G(V,E')$ in such a way that each vertex $i \in V$ is split into two vertex $\bar{r}_i$ and $\bar{R}_i$. So constraints (IIa) (IIg) and (IIc) can be transformed to the connection relationship in $E$. $\hat{V} = \{\bar{r}_1, \bar{R}_1, \ldots, \bar{r}_m, \bar{R}_m\}$. $E = E_1 \cup E_2 \cup E_3$, where $E_1$ include edges $(\bar{r}_i, \bar{R}_i)$, $E_2$ include edges $(\bar{R}_i, \bar{R}_j)$ and edges $(\bar{r}_i, \bar{r}_j)$ belong to $E_3$. Fig. (2a) illustrates a simple DAG $G$ representing a synchronous sequential circuit, and the transformed DAG $G$ of $G$ is illustrated in Fig. (2b).

Now the problem formulation can be simplified as follows:

\[
\begin{align*}
\min & \sum_{(i,j) \in E} P(s_{ij}) \\
\text{s.t.} & \quad (IVa) \\
& \quad 0 \leq \mu_i \leq \bar{N}_{ff} \quad \forall i \in \hat{V} \\
& \quad l_{ij} \leq s_{ij} \leq u_{ij} \quad \forall (i,j) \in \hat{E}
\end{align*}
\]
where $s_{ij}$ represents slack assigned to edge from node $i$ to $j$. For each edge $e(i,j) \in E_1$, if $i = \bar{i}_p$ and $j = \bar{r}_p$, then $s_{ij} = \bar{s}_p$, and $l_{ij} = \bar{s}_p^2$. For each edge $e(i,j) \in E_2$, $s_{ij} = \bar{s}_j - T \cdot w_{ij}$, then $l_{ij} = \bar{s}_j^2 - T \cdot w_{ij}$ and $u_{ij} = \bar{s}_j^2 - T \cdot w_{ij}$. For each edge $e(i,j) \in E_3$, $l_{ij} = -1 \cdot w_{ij}$ and $u_{ij} = \bar{N}_{ij}$. An example Power-Slack Curve of an edge in $E_1 \cup E_2$ and that of an edge in $E_3$ are illustrated in Fig. (3a) and Fig. (3b), respectively.

We then further eliminate constraints (IVb) and (IVc). First of all, $P(s_{ij})$ can be modified to eliminate the bounds on $\bar{s}_i$ as follows.

$$P(s_{ij}) = \begin{cases} P(u_{ij}) + M(s_{ij} - u_{ij}) & \bar{s}_i > u_{ij} \\ P(s_{ij}) & 0 \leq \bar{s}_i \leq T \\ P(l_{ij}) - M(s_{ij} - l_{ij}) & \bar{s}_i < l_{ij} \end{cases} \quad (11)$$

where $M$ is a sufficiently large number such that $P(s_{ij})$ is still a convex function.

Similarly, the bounds on $\mu_i$ can also be eliminated by adding into objective a convex cost function $B(\mu_i)$ defined as follows.

$$B(\mu_i) = \begin{cases} M \cdot (\mu_i - \bar{N}_{ij}) & \mu_i > \bar{N}_{ij} \\ 0 & 0 \leq \mu_i \leq \bar{N}_{ij} \\ -M \cdot \mu_i & \mu_i < 0 \end{cases} \quad (12)$$

After the above simplifications, problem (IV) can be transformed to problem (V):

$$\min \sum_{(i,j) \in E} P(s_{ij}) + \sum_{i \in V} B(\mu_i) \quad (V)$$

s.t. $\mu_j - \mu_i \geq s_{ij} \quad \forall (i,j) \in \bar{E}$

3.5. Problem Transformation by Lagrangian Relaxation

Using Lagrangian relaxation to eliminate constraint in problem (V), get the Lagrangian sub-problem:

$$L(\bar{x}) = \sum_{e(i,j) \in \bar{E}} P(s_{ij}) + \sum_{i \in V} B(\mu_i) \quad (13)$$

It is easy to show that

$$\sum_{e(i,j) \in \bar{E}} (u_{ij} - u_{ij}) x_{ij} = \sum_{i \in V} x_{0i} \times \mu_i \quad (14)$$

where

$$x_{0i} = \sum_{j : e(i,j) \in \bar{E}} x_{ij} - \sum_{j : e(j,i) \in \bar{E}} x_{ji}, \forall i \in V \quad (15)$$

Lagrangian subproblem (13) can be restated as follows:

$$L(\bar{x}) = \min \sum_{e(i,j) \in \bar{E}} [P(s_{ij}) + x_{ij} s_{ij}] + \sum_{i \in V} [B(\mu_i) + x_{0i} \mu_i] \quad (16)$$

A start node $v_0$ is added to $\bar{V}$, $v_0$ interconnects all other nodes in $\bar{V}$. We set $s_{0i} = \mu_i$, $l_{0i} = 0$, $u_{0i} = \bar{N}_{ij}$. So $V = \{v_0\} \cup \bar{V}$. The new edges are denoted as $E_4$, $E = \bar{E} \cup E_4$. The Power-Slack curve of an edge $e(i,j) \in E_4$ is illustrated in Fig. (3c). So we can transform $L(\bar{x})$ as formulation (17).

$$L(\bar{x}) = \min \sum_{e(i,j) \in \bar{E}} [P(s_{ij}) + x_{ij} s_{ij}] \quad (17)$$

s.t. \(\sum_{j : e(i,j) \in \bar{E}} x_{ij} - \sum_{j : e(j,i) \in \bar{E}} x_{ji} = 0 \quad \forall i \in V\)

$$x_{ij} \geq 0 \quad \forall (i,j) \in E_1 \cup E_2 \cup E_3$$

3.6. Convex Cost-scaling Approach

We define function $H_i(x_{ij})$ for each $e(i,j) \in E$ as follows:

$$H_i(x_{ij}) = \min_{x_{ij}} \{P_i(s_{ij}) + x_{ij} s_{ij}\} \quad (18)$$

For the $e(i,j) \in E_1$, because the function $H_i(x_{ij})$ is a piecewise linear concave function of $x_{ij}$, and $\forall e(i,j) \in E_1$, then $H_i(x_{ij})$ is described in the following manner [19]:

$$H_i(x_{ij}) = \begin{cases} P_i(s_{ij}^k) + s_{ij}^k x_{ij} & 0 \leq x_{ij} \leq b_{ij}(k) \\ \ldots \\ P_i(s_{ij}^q) + s_{ij}^q x_{ij} & b_{ij}(q-1) \leq x_{ij} \leq b_{ij}(q) \\ \ldots \\ P_i(s_{ij}^q) + s_{ij}^q x_{ij} & b_{ij}(q-1) \leq x_{ij} \leq b_{ij}(q) \end{cases}$$

where $b_{ij}(q) = \frac{P_i(s_{ij}^q) - P_i(s_{ij}^{q-1})}{s_{ij}^q - s_{ij}^{q-1}}$.

For the $e(i,j) \in E_2$, similar to $E_2$, then $H_i(x_{ij}) = \begin{cases} P_i(t_{ij}^1) + t_{ij}^1 x_{ij} & 0 \leq x_{ij} \leq b_{ij}(k) \\ \ldots \\ P_i(t_{ij}^q) + t_{ij}^q x_{ij} & b_{ij}(q-1) \leq x_{ij} \leq b_{ij}(q) \\ \ldots \\ P_i(t_{ij}^q) + t_{ij}^q x_{ij} & b_{ij}(q-1) \leq x_{ij} \leq b_{ij}(q) \end{cases}$
where \( b_{ij}(q) = \frac{P_i(q s_{ij}^{r(q)}) - P_i(q s_{ij}^{l(q)})}{s_{ij}^{r(q)} - s_{ij}^{l(q)}} \), and \( s_{ij}^q = s_{ij}^q - T \cdot w_{ij} \).

For the \( e(i,j) \in E_3 \), because \( P_i(s_{ij}) = 0 \),
\[
H_{ij}(x_{ij}) = \min_{s_{ij}} (s_{ij} x_{ij}) = - T \cdot w_{ij} \cdot x_{ij}, \quad x_{ij} \geq 0
\]

For the \( e(i,j) \in E_4 \), the variable \( x_{ij} \) is not a Lagrangian multiplier, and it is bounded by \(-M \leq x_{ij} \leq M\).
\[
H_{ij}(x_{ij}) = \left\{ \begin{array}{ll}
0 & 0 \leq x_{ij} \leq M \\
N_{ij} & M \leq x_{ij} \leq 0
\end{array} \right.
\]

Note that these functions \( H_{ij}(x_{ij}) \) are all concave. We define \( C_{ij}(x_{ij}) = -H_{ij}(x_{ij}) \), so that \( C_{ij}(x_{ij}) \) is a piecewise linear convex function. Then we can subsequently propose problem (VI) as follows:

\[
L(\bar{x}) = \min \sum_{e(i,j) \in E} C_{ij}(x_{ij}) \quad (VI)
\]
\[
s.t. \sum_{j:e(i,j) \in E} x_{ij} - \sum_{j:e(i,j) \in E} x_{ji} = 0 \quad \forall i \in V
\]
\[
0 \leq x_{ij} \leq M \quad \forall (i,j) \in E_1 \cup E_2 \cup E_3
\]
\[
-M \leq x_{ij} \leq M \quad \forall (i,j) \in E_4
\]

To transform the problem into a minimum cost flow problem, we construct an expanded network \( G' = (V', E') \). There are four kinds of edges to consider:

- \( e(i,j) \) in \( E_1 \): we introduce \( k \) edges in \( G' \), and the costs of these edges are: \(-s_{ij}^k, -s_{ij}^{k-1}, \ldots, -s_{ij}^1\); upper capacities: \( b_{ij}(k), b_{ij}(k-1) - b_{ij}(k), b_{ij}(k-2) - b_{ij}(k-1), \ldots M - b_{ij}(2) \), where \( M \) is a huge coefficient; lower capacities are all 0.
- \( e(i,j) \) in \( E_2 \): we introduce \( k \) edges in \( G' \), and the costs of these edges are: \(-t_{ij}^k, -t_{ij}^{k-1}, \ldots, -t_{ij}^1\); upper capacities: \( b_{ij}(k), b_{ij}(k-1) - b_{ij}(k), b_{ij}(k-2) - b_{ij}(k-1), \ldots M - b_{ij}(2) \), where \( M \) is a huge coefficient; lower capacities are all 0.
- \( e(i,j) \) in \( E_3 \): cost, lower and upper capacity is \((c \cdot w_{ij}, 0, M)\).
- \( e(i,j) \) in \( E_4 \): two edges are introduced in \( G' \), one with cost, lower and upper capacity as \((N_{ij}, -M, 0)\), another is \((0, 0, M)\).

Using the cost-scaling algorithm [24], we can solve the minimum cost flow problem in \( G' \). For the given optimal flow \( x^* \), we construct residual network \( G(x^*) \) and solve a shortest path problem to determine shortest path distance \( d(i) \) from node \( s \) to every other node. By implying that \( \mu(i) = d(i) \) and \( s_{ij} = \mu(i) - \mu(j) \) for each \( e(i,j) \in E_1 \cup E_2 \), we can finally solve problem (III).

## 4. Experimental Results

We implemented our algorithm in the C++ programming language and executed on a Linux machine with eight 3.0GHz CPU and 6GB Memory. 19 cases from the ISCAS89 benchmarks are tested, and the name, number of gates, number of signal passes, the maximum number of gate output/inputs, and the minimum period for each case are given in Table I. We used four discrete slack levels for each gate as \( \{0, 10, 20, 33\} \). Energy consumption of the gates with slack level scaling were found from model in [15].

In the experiments, a min-period retiming algorithm [4] is first employed to generate the minimum clock period \( T \), which is listed in the 2nd column of TABLE II. Liu et al.’s [18] algorithm was implemented for comparison. Note that algorithm [18] cannot directly solve discrete slack budgeting problem, because if sensitive transitive closure graph is used, the timing constraints might be violated after slack budgeting [8]. Therefore we use a transitive closure graph instead of sensitive transitive closure graph here. To evaluate the accuracy of our algorithm, the ILP for achieving the optimal solution were also implemented using an open source ILP solver CBC [25].

Table II shows comparisons among optimal ILP, algorithm in [18] and our algorithm. The column Power Consumption gives actual power consumption of each circuit and less value means more power can be reduced. Comparing with optimal solution, our algorithm increases 29% power consumption while [18] increases 52%. Column Total Slack gives the sum of each gate’s slack. Comparing with optimal solution, our algorithm loses 16% of slacks while [18] loses 31%. Note that power consumption is not proportional to the slack amount. As for benchmark s27.test, [18] and optimal ILP get equal slack amount, but their power consumption is different. Column Runtime compares the run time of each algorithm. From the results we can find that although optimal ILP can get optimal solution, its runtime sometimes is unacceptable. Comparing with [18], our algorithm can not only generate better design results, but also get nearly 500× speedup.

## 5. Conclusion

In this paper we have showed that the retiming and slack budgeting problem can be simultaneously solved by formulating the problem to a convex cost dual network flow problem. Both the theoretical analysis and experimental results show the
efficiency of our approach which can not only reduce power consumption but also speedup previous work.

REFERENCES


---

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>T</th>
<th>Power Consumption</th>
<th>Total Slacks</th>
<th>Runtime(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s27.test</td>
<td>20</td>
<td>800 824 850</td>
<td>0.02</td>
<td>0.00</td>
</tr>
<tr>
<td>s208.1.test</td>
<td>28</td>
<td>3542 1198 4772</td>
<td>0.39</td>
<td>0.44 0.06</td>
</tr>
<tr>
<td>s298.test</td>
<td>24</td>
<td>6498 8888 8010</td>
<td>0.78</td>
<td>0.69 0.07</td>
</tr>
<tr>
<td>s382.test</td>
<td>44</td>
<td>6456 9038 9958</td>
<td>&gt;1000</td>
<td>10.56 0.12</td>
</tr>
<tr>
<td>s386.test</td>
<td>64</td>
<td>8836 12870 9564</td>
<td>4.58</td>
<td>1.03 0.1</td>
</tr>
<tr>
<td>s344.test</td>
<td>46</td>
<td>9876 11848 9894</td>
<td>0.82</td>
<td>2.53 0.09</td>
</tr>
<tr>
<td>s349.test</td>
<td>46</td>
<td>9938 12472 9894</td>
<td>0.79</td>
<td>4.49 0.11</td>
</tr>
<tr>
<td>s444.test</td>
<td>46</td>
<td>8938 14032 11884</td>
<td>&gt;1000</td>
<td>12.04 0.12</td>
</tr>
<tr>
<td>s526.test</td>
<td>42</td>
<td>7602 14106 11498</td>
<td>42.57</td>
<td>1.67 0.17</td>
</tr>
<tr>
<td>s526n.test</td>
<td>42</td>
<td>7752 11734 11548</td>
<td>30.32</td>
<td>4.72 0.17</td>
</tr>
<tr>
<td>s510.test</td>
<td>42</td>
<td>13976 17492 14864</td>
<td>37.24</td>
<td>1.62 0.17</td>
</tr>
<tr>
<td>s420.1.test</td>
<td>50</td>
<td>4574 17920 9224</td>
<td>1.29</td>
<td>16.91 0.14</td>
</tr>
<tr>
<td>s832.test</td>
<td>98</td>
<td>13652 14518 16274</td>
<td>71.96</td>
<td>151.26 0.24</td>
</tr>
<tr>
<td>s820.test</td>
<td>92</td>
<td>13552 17694 16448</td>
<td>68.98</td>
<td>13.18 0.25</td>
</tr>
<tr>
<td>s641.test</td>
<td>238</td>
<td>13334 20408 14424</td>
<td>2.24</td>
<td>92.97 0.26</td>
</tr>
<tr>
<td>s713.test</td>
<td>262</td>
<td>13018 21228 14322</td>
<td>2.27</td>
<td>121.1 0.27</td>
</tr>
<tr>
<td>s838.1.test</td>
<td>80</td>
<td>6004 1889 2356</td>
<td>42.57</td>
<td>1.67 0.17</td>
</tr>
<tr>
<td>s1238.test</td>
<td>110</td>
<td>6096 10444 8208</td>
<td>2.27</td>
<td>121.1 0.27</td>
</tr>
<tr>
<td>s1488.test</td>
<td>166</td>
<td>21292 23799 27836</td>
<td>42.57</td>
<td>1.67 0.17</td>
</tr>
<tr>
<td>Avg</td>
<td>-</td>
<td>9249.3 14070 11947.9</td>
<td>5457.7</td>
<td>95.3 0.19</td>
</tr>
<tr>
<td>Diff</td>
<td>-</td>
<td>1 +52% +29%</td>
<td>1 -31%  -16%</td>
<td>1 0.002</td>
</tr>
</tbody>
</table>

TABLE II: Comparisons with Optimal ILP and Previous Work [18]