

# Physical Synthesis for FPGA Interconnect Power Reduction by Dual-Vdd Budgeting and Retiming

YU HU, YAN LIN, and LEI HE  
University of California, Los Angeles  
and  
TIM TUAN  
Xilinx Research Labs

---

Field programmable dual-Vdd interconnects are effective in reducing FPGA power. We formulate the dual-Vdd-aware slack budgeting problem as a linear program (LP) and a min-cost network flow problem, respectively. Both algorithms reduce interconnect power by 50% on average compared to single-Vdd interconnects, but the network-flow-based algorithm runs 11x faster on MCNC benchmarks. Furthermore, we develop simultaneous retiming and slack budgeting (SRSB) with flip-flop layout constraints in dual-Vdd FPGAs based on mixed integer linear programming, and speed-up the algorithm by LP relaxation and local legalization. Compared to retiming followed by slack budgeting, SRSB reduces interconnect power by up to 28.8%.

Categories and Subject Descriptors: B.7.2 [**Integrated Circuits**]: Design Aids

General Terms: Algorithms, Design

Additional Key Words and Phrases: Low power, retiming, FPGA

## ACM Reference Format:

Hu, Y., Lin, Y., He, L., and Tuan, T. 2008. Physical synthesis for FPGA interconnect power reduction by dual-Vdd budgeting and retiming. *ACM Trans. Des. Autom. Electron. Syst.* 13, 4, Article 30 (April 2008), 29 pages, DOI = 10.1145/1344418.1344426 <http://doi.acm.org/10.1145/1344418.1344426>

---

## 1. INTRODUCTION

Field programmable dual-Vdd techniques have been used for FPGA power reduction for FPGA logic blocks [Li et al. 2004a, 2004c] and interconnects

---

This article is partially supported by NSF grant CCR-0306682. Address comments to [lhe@ee.ucla.edu](mailto:lhe@ee.ucla.edu).

Authors' addresses: Y. Hu, Y. Lin, and L. He, Electrical Engineering Department, UCLA, Los Angeles, CA 90095; T. Tuan, Xilinx Research Laboratories, 2500 Logic Drive, San Jose, CA 95124-3400.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org). © 2008 ACM 1084-4309/2008/04-ART30 \$5.00 DOI 10.1145/1344418.1344426 <http://doi.acm.org/10.1145/1344418.1344426>

ACM Transactions on Design Automation of Electronic Systems, Vol. 13, No. 2, Article 30, Pub. date: April 2008.

[Gayasen et al. 2004; Li et al. 2004b]. The leading-edge commercial FPGA has also applied field programmable dual-Vdd [Altera 2006]. In this article, we study interconnect power reduction using Vdd programmable interconnects. The interconnects consist of buffered wire segments. Buffers have programmable Vdd levels. A Vdd-level converter is needed when a low-Vdd (VddL) buffer drives a high-Vdd (VddH) buffer to avoid excessive leakage. In Li et al. [2004b], a level converter is inserted in front of each interconnect buffer to provide the fine-grained Vdd programmability for interconnects. However, it has been shown in Lin and He [2006] that this fine-grained Vdd-level converter insertion may introduce large leakage and area overhead. Recently, a few approaches have been presented without directly using level converters in Vdd programmable interconnects. Anderson and Najm [2004] assume level-restore buffers, where an NMOS power transistor is used to generate the VddL level by leveraging the threshold voltage drop of the NMOS transistor when it is turned on and it avoids two power supply networks. However, the range of VddL is limited by the threshold voltage of the NMOS power transistor. The positive-feedback PMOS transistor in the level-restore buffer can be viewed as an alternative level converter which has less leakage overhead, but may cause larger delay and therefore larger short-circuit power compared to the level converter used in Li et al. [2004b]. Gayasen et al. [2004] enforce that all the routing trees driven by (driving) a logic block had the same Vdd level as the source (sink) logic block where level converters are inserted at CLB inputs (outputs). Lin et al. [2005b] assume that the level converter is not inserted in interconnects while each routing tree has one Vdd level. In this article, we assume dual-Vdd interconnects identical to those in Lin and He [2006]. There are Vdd-level converters at the inputs and outputs of logic blocks, but not between wire segments. The Vdd assignment algorithm guarantees that no VddL buffer should drive VddH buffers such that each routing tree may have two Vdd levels but without a level converter within a routing tree (see Figure 1). Furthermore, slack budgeting is formulated to distribute timing slacks to all nets such that the chip-level power is minimized.

Uniform wire length and buffer size are assumed in Lin and He [2006]. However, state-of-the-art commercial FPGAs have used wire segments of different lengths to improve performance [Xilinx 2005; Lewis et al. 2003]. For uniform wire-length FPGAs, the sizes of different interconnect switches within a routing tree are the same and so are their loading capacitance values. This enables us to budget the VddL switch number by uniformly distributing the timing slack along a source-to-sink path. Unfortunately, those assumptions are no longer true for mixed wire-length designs. The interconnect switches with different sizes are needed to drive different wire segments. We show that the lower bound estimation of power reduction in Lin and He [2006] is not valid for mixed-length interconnects, and it becomes a challenge to estimate the potential number of VddL switches based on the given timing slack for mixed wire-length designs. In this work, we present a new method for Vdd-level assignment in mixed wire-length FPGA designs based on an estimated upper bound of power reduction. Tested on both industrial designs and MCNC benchmarks, the experimental results show that the proposed upper bound of power

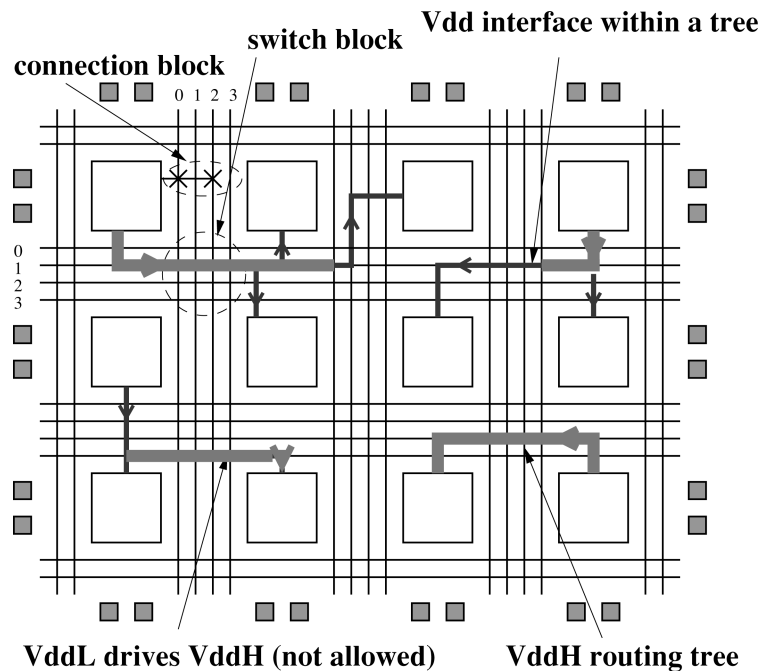


Fig. 1. Vdd programmable interconnect FPGA architecture: Dark (thin) wire segments have VddL while shaded (thick) wire segments have VddH.

reduction is highly correlated to the power reduction obtained by cycle-accurate simulation [Lin et al. 2005a]. An LP-based formulation for the dual-Vdd timing slack budgeting problem is then presented based on this power reduction estimation and a 50% power reduction is achieved on average compared to single-Vdd interconnects. Moreover, we show that the proposed upper bound of power reduction estimation enables an efficient reformulation of the dual-Vdd budgeting problem by min-cost network flow, which is about 11x faster than the LP-based one on MCNC benchmarks.

The slack budgeting in Lin and He [2006] is applied only within combinational subcircuits. By adding retiming [Leiserson and Saxe 1991] as an extra design freedom, we can simultaneously consider all combinational subcircuits in a sequential circuit to explore a larger search space for greater power reduction. The retiming procedure relocates registers to reduce the clock period, area, or power dissipation while preserving the I/O functionality of the circuit. It has been studied extensively in logic and physical synthesis for ASICs. Fischer et al. [2005], Monteriro et al. [1993], Hsu and Wang [2002], and Anan et al. [1998] consider retiming with pipelining for low-power designs. Cong et al. [1999], Cong and Lim [2000], and Cong and Yuan [2003] employ retiming in placement to minimize the clock period. Tien et al. [1998] propose an algorithm to retime circuits in the postlayout stage to speed-up ASIC designs. For FPGAs, Cong and Wu [1998a, 1998b, 1997, 1996] present several effective algorithms for technology mapping with retiming for high-performance circuits. Retiming is also

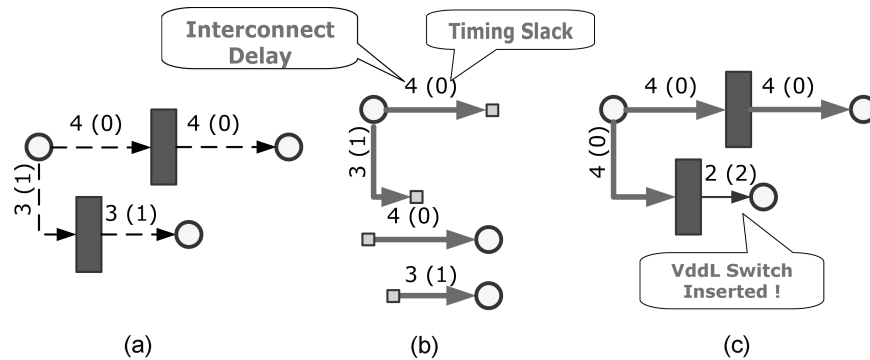


Fig. 2. Comparison of effectiveness for power reduction by the sequential approach and simultaneous approach: (a) original circuit; (b) sequential approach (retiming followed by budgeting); (c) simultaneous retiming and dual-Vdd budgeting. The number outside (inside) the brackets associated in each edge denotes the interconnect delay (timing slack) of that edge, respectively. The thick, shaded lines are interconnects driven by VddH while the thin, dark lines are interconnects driven by VddL.

combined with slack budgeting in Chabini and Chabini [2003] for low power for ASIC designs. Yeh and Marek-Sadowska [2003a, 2003b] present simultaneous slack budgeting and retiming algorithms considering area minimization for high-performance FPGAs. However, the power reduction and postlayout flip-flop binding constraints are not considered in Yeh and Marek-Sadowska [2003a, 2003b].

In this article, we present a mixed integer and linear programming (MILP)-based approach for simultaneous retiming and slack budgeting (SRSB) to further reduce interconnect power for FPGAs. The motivation of this idea is illustrated in Figure 2, where circuits in (a), (b), and (c) have the same clock period of 4 units. To change a buffer from VddH to VddL, one needs a slack of 2 units. If circuit (a) is decomposed to its combinational subcomponents (b) and dual-Vdd budgeting is performed, no extra buffer can be powered by VddL. On the other hand, one extra buffer can be powered by VddL in (c), which can be obtained from (a) by retiming under the same clock period. Therefore, SRSB is able to reduce more power than slack budgeting alone in Lin and He [2006]. To minimize the distortion of the placement and routing, the placement and flip-flop (FF) binding constraints are considered during the retiming process. Note that our retiming approach will not increase chip area, since we only use the presently available FF slots in placed FPGA circuits. Compared to the sequential approach (min-clock retiming followed by dual-Vdd budgeting), the MILP-based SRSB approach achieves 7.7% (up to 28.8%) and 3.8% (up to 7.0%) interconnect power reduction on average for MCNC and industrial circuits, respectively. To overcome the intractability of the MILP problem, an LP relaxation procedure followed by flip-flop legalization is presented with 7x (up to 30x) speedup on average. To further reduce runtime, we find a reliable indicator for the potential power reduction by SRSB, and present an efficient postlayout power-aware resynthesis CAD flow to apply SRSB only when necessary. To the best of our knowledge, this article is the first in-depth study of simultaneous retiming and

slack budgeting for dual-Vdd programmable FPGA power reduction while considering FF binding constraints.

The rest of this article is organized as follows. Section 2 introduces background and modeling. Section 3 describes a dual-Vdd slack budgeting algorithm and experimental results for mixed wire-length FPGAs. Section 4 presents the simultaneous retiming and dual-Vdd slack budgeting algorithm and experimental results. Section 5 concludes the work.

## 2. PRELIMINARIES AND MODELING

### 2.1 Preliminaries

We assume the traditional island-style routing architecture [Betz 1999] as shown in Figure 1. The logic blocks (CLBs) are surrounded by routing channels consisting of wire segments. In our experiments, every CLB consists of ten logic cells (LCs) and every LUT has four inputs, which has been shown to achieve the best power, delay, and area tradeoffs [Lin et al. 2005a]. The input and output pins of a CLB can be connected to the wire segments in the surrounding channels via a *connection block*. There is a routing switch block at each intersection of a horizontal channel and a vertical channel. An interconnect switch is either a routing switch or a connection switch. For the rest of this article, we use “switch” to represent an interconnect switch for simplicity whenever there is no ambiguity. As suggested in Lewis et al. [2003], we assume a mix of different interconnect wire lengths and use 60% length 4 wire and 40% length 8 wire. Having properly sized the interconnect switches, we use 25x and 10x switches<sup>1</sup> to drive length-8 and length-4 wires, respectively. Following Lin and He [2006], we use subset switch boxes and apply Vdd programmability to interconnects to reduce FPGA power. We use the Vdd programmable interconnect switch with the minimal number of configuration SRAM cells proposed in Lin et al. [2005a]. Two PMOS power transistors are inserted between the tristate buffer and VddH, VddL power rails, respectively. Turning off one of the power transistors, we can select a Vdd level for the routing switch. By turning off both power transistors, an unused routing switch can be power-gated. Similar to the routing switch, a programmable Vdd is also applied to the connection switch. To make the presentation simple, we summarize the notations frequently used in this article in Table I. They will be explained in detail when first used.

### 2.2 Delay Modeling with Dual-Vdd

A directed acyclic timing graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  is constructed to model the circuit for timing analysis. Vertices represent the input and output pins of basic circuit elements such as registers and LUTs. Edges are added between the inputs of combinational logic elements (e.g., LUTs) and their outputs, and between the connected pins specified by the circuit netlist. Register inputs are not joined to register outputs. Each edge is annotated with the delay required to pass through

<sup>1</sup>In our experiments, the width of the NMOS transistor for a 1x inverter is  $4\lambda$ , and a 25x (10x) switch means that the width of the last inverter is 25x (10x) of a 1x inverter.

Table I. Notations Frequently Used in This Article

$SR\mathcal{C}$	set of vertices corresponding to routing tree sources
$\mathcal{R}_i$	$i$ th routing tree in FPGA
$\mathcal{FO}_{ij}$	set of fanout switches of $j$ th switch in $\mathcal{R}_i$
$\mathcal{SL}_{ij}$	set of sinks in the fanout cone of $j$ th switch in $\mathcal{R}_i$
$a(v)$	arrival time of vertex $v$ in $\mathcal{G}$
$d(u, v)$	delay from vertex $u$ to vertex $v$ in $\mathcal{G}$
$N_r$	total number (no.) of routing trees in FPGA
$c_{ij}$	load capacitance of $j$ th switch in $\mathcal{R}_i$
$l_{ik}$	no. of switches in the path from source to $k$ th sink in $\mathcal{R}_i$
$S_{ik}$	allocated slack for $k$ th sink in $\mathcal{R}_i$
$p_{i0}$	vertex in $\mathcal{G}$ corresponding to the source of $\mathcal{R}_i$
$p_{ik}$	vertex in $\mathcal{G}$ corresponding to $k$ th sink of $\mathcal{R}_i$
$f_s(i, j)$	transition density of $j$ th switch in $\mathcal{R}_i$
$N_k(i)$	no. of sinks in $\mathcal{R}_i$
$N_s(i)$	total no. of switches in $\mathcal{R}_i$
$N_l(i)$	no. of VddL switches in $\mathcal{R}_i$
$F_n(i)$	estimated no. of VddL switches in $\mathcal{R}_i$
$w_e$	FF number in $e(i, j)$
$r_u$	retiming value in node $u$

the circuit element or routing. We use  $\mathcal{PI}$  to represent the set of primary inputs and register outputs which have no incoming edges, and  $\mathcal{PO}$  to represent the set of primary outputs and register inputs which have no outgoing edges.

The Elmore delay model is used to calculate the routing delay. Following Lin and He [2006], we define the *fanout cone* of a switch as the subtree of the routing tree rooted at the switch. Assigning VddL to a switch affects the delay from the source to all the sinks in its fanout cone, and therefore affects the delay of the corresponding edges in  $\mathcal{G}$ . To incorporate dual-Vdd into timing analysis, we use SPICE to precharacterize the intrinsic delay and effective driving resistance for a switch under VddH and VddL, respectively. Considering the VddL/VddH ratio between 0.6 ~ 0.7 suggested in Hamada [1998], we use 1.3v for VddH and 0.8v for VddL in this article at 100nm technology node, where all parameters are extracted by the Berkeley predictive device model [BPTM 2002]. As shown in Lin and He [2006], Vdd level has little impact on the input and load capacitance of a switch; such impact is ignored in this article.

### 2.3 Power Modeling with Dual-Vdd

There are three types of power source in FPGAs: switching power, short-circuit power, and static (leakage) power. The first two contribute to the dynamic power and can only occur when a signal transition happens at the gate output. The transition density of each node in the circuit is collected by a cycle-accurate FPGA power simulator by assuming random inputs at primary inputs [Lin et al. 2005b]. Although a timing change may change the transition density, our problem formulation assumes (as in Lin and He [2006]) that the transition density for an interconnect switch does not change when VddL is used. The third type of power, static power, is the power consumed when there is no signal transition for a circuit element. We assume that the unused switches are power-gated. Despite simplification such as no short-circuit power and fixed transition

density in our formulation, a cycle-accurate power simulation considering short-circuit power and accurate transition density calculation [Lin et al. 2005b] is performed to verify the experimental results in Section 3.3.

Let  $v_{ij}$  indicate the Vdd level of the  $j$ th switch in  $\mathcal{R}_i$ , as follows.

$$v_{ij} = \begin{cases} 1 & \text{if Vdd-level of the } j\text{th switch in } \mathcal{R}_i \text{ is VddH} \\ 0 & \text{if Vdd-level of the } j\text{th switch in } \mathcal{R}_i \text{ is VddL} \end{cases}$$

The interconnect power reduction  $P_r$  using programmable dual-Vdd can be expressed as

$$P_r = \sum_{i=0}^{N_r-1} \sum_{j=0}^{N_s(i)-1} (1 - v_{ij})(0.5 f_{clk} f_s(i, j) c_{ij} (VddH^2 - VddL^2) + \Delta P_s(i, j)), \quad (1)$$

which is the sum of dynamic and leakage power reduction.  $N_r$  is the total number of routing trees,  $f_s(i, j)$  is the transition density<sup>2</sup> of the  $j$ th switch in the  $i$ th routing tree  $\mathcal{R}_i$ ,  $N_s(i)$  is the number of switches in  $\mathcal{R}_i$ , and  $\Delta P_s(i, j)$  and  $c_{ij}$  are the leakage power reduction and load capacitance of each switch, respectively.

#### 2.4 Retiming with FF Constraints

In order to retime an FPGA design during postlayout optimization, we need to consider placement and FF binding constraints. In older FPGA devices, a logic cell (LC) has only one output. If a LUT drives an FF within the same LC, the combinational output of a LUT is not allowed to drive other FFs, since it cannot be seen by other FFs. In newer devices such as the Xilinx Virtex-IV [Xilinx 2005] or Altera Stratix II [Lewis et al. 2005], an LC has both the combinational output (without FF) and sequential output (with FF). If a LUT drives an FF within the same LC, the sequential output can still drive other FFs, either within the same cluster (through local routing) or in other clusters (through global routing). In other words, FFs can be cascaded. FFs within a cluster can be cascaded through a LUT (acting as a WIRE) or a MUX. Moreover, FFs within different clusters can be cascaded going through a global routing and a LUT or MUX in the other cluster. In postlayout retiming, we try to keep most layout (both placement and global routing) unchanged, therefore we allow a LUT to drive any available FFs within the same cluster, but not FFs in a different cluster.

Without loss of generality, Figure 3 shows the simplified model of the 5-input LC structure in modern FPGAs. Two outputs correspond to the normal output and sequential output. The sequential output is enabled when  $IS$  is used to feed FF directly, which enables the independent access of LUT and FF within one LC.

<sup>2</sup>In general, all interconnect switches in a routing tree have the same transition density. Nevertheless, a glitch with small duration may degrade in both amplitude and duration when propagating through logic gates due to the electrical property of those gates [Wirth et al. 2005].

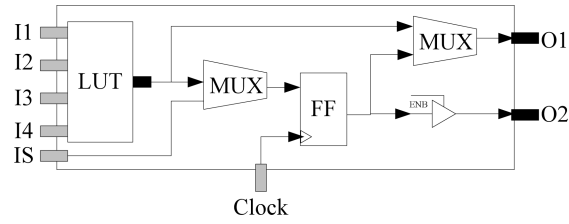


Fig. 3. A simplified model for LC in modern FPGAs.

Table II. Power Reduction for Mixed-Length FPGAs by LP-Based Dual-Vdd Timing Slack Budgeting

Benchmarks			svdd	dual-Vdd	
circuit	clb#	net#	svdd-pwr	VddL# (%)	dvdd-pwr(redu)
bigkey	294	1542	0.1553	68.0%	0.1038 (-33.1%)
clma	1358	7995	0.1554	75.3%	0.0837 (-46.2%)
diffeq	195	1291	0.0142	88.5%	0.0061 (-57.1%)
dsip	162	1139	0.1809	68.4%	0.1200 (-33.7%)
elliptic	421	2617	0.0516	91.1%	0.0221 (-57.2%)
frisc	595	3240	0.0338	98.1%	0.0143 (-57.8%)
s298	256	908	0.0613	81.6%	0.0338 (-44.9%)
s38417	847	5426	0.1704	82.7%	0.0960 (-43.7%)
s38584.1	704	4502	0.1501	95.0%	0.0645 (-57.0%)
tseng	131	918	0.0164	95.4%	0.0065 (-60.1%)
ave	469	2792	0.0939	83.5%	0.0523 (-48.5%)
ex1	28	195	0.0106	81.8%	0.0054 (-49.4%)
ex2	58	306	0.0111	92.9%	0.0048 (-56.7%)
ex3	82	653	0.0083	99.7%	0.0033 (-60.3%)
ex4	129	541	0.0736	66.2%	0.0509 (-30.7%)
ex5	163	1179	0.0319	94.2%	0.0138 (-56.8%)
ex6	228	1421	0.0137	87.5%	0.0066 (-51.9%)
ex7	271	1637	0.0170	90.7%	0.0080 (-52.7%)
ex8	284	1845	0.0483	95.9%	0.0210 (-56.4%)
ex9	293	1849	0.0268	90.7%	0.0129 (-51.7%)
ex10	371	2820	0.3635	70.0%	0.1917 (-47.2%)
ex11	409	2557	0.1451	75.9%	0.0748 (-48.4%)
ex12	584	3890	0.3833	77.8%	0.2096 (-45.3%)
ex13	630	4675	0.0703	89.8%	0.0352 (-49.9%)
ex14	1110	8240	0.2803	80.4%	0.1472 (-47.4%)
ave	331	2272	0.1060	85.3%	0.0561 (-50.4%)

## 2.5 Generic Experimental Settings

For simplicity of presentation, we summarize the common experimental settings used in this article. We conduct all experiments on the 10 biggest MCNC sequential circuits in the MCNC benchmark [Yang 1991] and 14 circuits from industrial designs.<sup>3</sup> As shown in Section 2.1, we map them to 4-input LUTs by DAOMap [Chen and Cong 2004] and pack them to CLBs with 10 LCs by TV-pack [Betz 1999]. The characters of these circuits are shown in Table II. We use mosek [Mosek 2006] to solve LP and MILP problems. All experimental

<sup>3</sup>These 14 circuits are a mix of real industry designs, IP cores, and generated benchmarks from smaller designs. These designs are used to benchmark Xilinx ISE [Xilinx 2006].

data is collected on a Linux workstation with a 1.9GHz Xeon CPU and 2GB memory.

In all the experiments, we first use VPR [Betz 1999] for single-Vdd placement and routing. Before applying retiming and budgeting algorithms to the Vdd programmable interconnects, a sensitivity-based assignment [Li et al. 2004a] is first performed to assign Vdd levels for Vdd programmable logic blocks without performance loss.<sup>4</sup> And a min-clock retiming [Lin and Zhou 2004] is then performed for further performance improvement without changing placement and routing. After this, the Vdd levels for interconnect switches are assigned by either a timing slack budgeting algorithm (see Section 3) or a simultaneous retiming and timing slack budgeting algorithm (see Section 4). The cycle-accurate FPGA power simulator fpgaEva-LP2 [Lin et al. 2005a] is then used to calculate, interconnect power of the final designs. It has been shown that fpgaEva-LP2 achieves high fidelity as well as high accuracy as compared to SPICE simulation, with an average of absolute error 8.26% [Lin et al. 2005a]. Because the power computation in fpgaEva-LP2 considers short circuit power and uses input vectors, it is more accurate than the power model in our problem formulations. Using fpgaEva-LP2 verifies both our modeling and problem formulations.

### 3. DUAL-VDD TIMING SLACK BUDGETING

In this section, we present an LP-based timing slack budgeting algorithm for mixed interconnect wire-lengths and then reformulate it into an min-cost network flow problem to obtain more efficiency.

The overall flow of our Vdd-level assignment algorithm is as follows. The timing slack is first allocated to each source-to-sink path in every routing tree by solving an LP (or a network flow)-based formulation of the timing slack budgeting problem, which considers the load capacitance of each switch explicitly. We then perform a bottom-up assignment algorithm to achieve the optimal solution within each routing tree for the allocated timing slack,<sup>5</sup> as in Lin and He [2006].

#### 3.1 Estimation of Interconnect Power Reduction

Estimating power reduction for the allocated slack is the key to enable the LP-based algorithm. The slack  $S_{ij}$  of a connection between the source and the  $j$ th sink in  $\mathcal{R}_i$  is defined as the amount of delay which could be added to this connection without increasing the cycle time  $T_{spec}$ . There is an upper bound for useful slack, which is the delay increase when VddL is assigned to all the switches in a tree. Clearly, slack greater than the upper bound cannot lead to more VddL switches. We define the *useful slack* of each routing tree sink as the slack less than this upper bound. For the rest of the article, we use “slack” to represent the useful slack. The slack upper-bound constraints can be expressed as

$$0 \leq S_{ik} \leq D_{ik} \quad 0 \leq i < N_r \wedge 1 \leq k \leq N_k(i), \quad (2)$$

<sup>4</sup>We do consider the fact that VddL logic blocks consume timing slack.

<sup>5</sup>*Allocated timing slack* denotes the timing slack that is assigned to a routing edge after solving the LP (or network flow) problem.

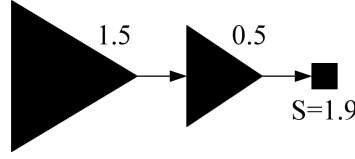


Fig. 4.  $F_n^{low}(i)$  in a mixed-length case.

where  $N_k(i)$  is the number of sinks in  $\mathcal{R}_i$  and  $D_{ik}$  is the delay increase of the path from the source to the  $k$ th sink in  $\mathcal{R}_i$  when VddL is assigned to all the switches in that path.

Given a routing tree with arbitrary topology and allocated slack for each sink, we need to estimate the power reduction that can be achieved. We use  $l_{ik}$  to represent the number of switches in the path from the source to the  $k$ th sink in  $\mathcal{R}_i$ . We first transform slack  $S_{ik}$  into  $s_{ik}$ , which is expressed in number of switches. Intuitively, the likelihood of powering a source-to-sink path by VddL is proportional to the number of switches and the timing slack assigned on the path. Therefore  $s_{ik}$  is expressed as

$$s_{ik} = \frac{S_{ik}}{D_{ik}} \cdot l_{ik}. \quad (3)$$

We then estimate the number of VddL switches that can be achieved using  $s_{ik}$ . Let  $c_{ij}$  represent the load capacitance of the  $j$ th switch in  $\mathcal{R}_i$ , and  $C_{ik}$  represent the total load capacitance of the switches in the path from the source to the  $k$ th sink in  $\mathcal{R}_i$ . We define *sink list*  $\mathcal{SL}_{ij}$  as the set of sinks in the fanout cone of the  $j$ th switch in  $\mathcal{R}_i$ . Lin and He [2006] present the lower bound of the number of VddL switches for the allocated slack in uniform-length cases. A straightforward extension to handle mixed wire length is

$$F_n^{low}(i) = \sum_{j=0}^{N_s(i)-1} \min \left( \frac{s_{ik}}{C_{ik}} \cdot c_{ij} : \forall k \in \mathcal{SL}_{ij} \right), \quad (4)$$

where the term  $\frac{s_{ik}}{C_{ik}}$  expresses the truth that the relative size of the switch compared to the total sizes of all switch buffers determines the likelihood of the switch being powered by VddL.

We find that  $F_n^{low}(i)$  is not a lower bound in mixed-length cases, where the size of switches along a source-to-sink path may be different. In general, interconnects with mixed wire-lengths allow wire segments with different lengths to connect to each other. Figure 4 shows a source-to-sink path which includes two switches with different sizes. They need 1.5 and 0.5 slacks to be powered by VddL, respectively. If we assign 1.9 slack in the sink,  $F_n^{low}(i) = 1.9$ , based on Eq. (4). However, when we perform bottom-up Vdd assignment, the lower stream switch consumes 0.5 slack and there is 1.4 slack left for the upper stream switch, which is not enough for it to be a VddL. Therefore, we get only one VddL switch with used slack of 0.5 instead of the estimated 1.9. In addition, as shown in Section 3.4, the Vdd-level assignment problem can be formulated as a min-cost network flow problem which can be solved more efficiently if we remove the nonlinear term “min” in the lower-bound formulation (4).

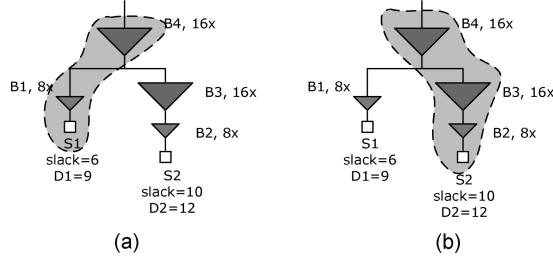


Fig. 5. Illustration of VddL switch assignment estimation.

We propose an upper bound  $F_n^{up}(i)$  of VddL switch number in  $\mathcal{R}_i$  by summing up all  $s_{ik}c_{ij}/C_{ik}$  in its fanout cone as the slack distributed to the switch. It is expressed as

$$F_n^{up}(i) = \sum_{j=0}^{N_s(i)-1} f_n^{up}(i, j) \quad (5)$$

$$f_n^{up}(i, j) = \sum_{\forall k \in SC_{ij}} \frac{s_{ik}}{C_{ik}} \cdot c_{ij}, \quad (6)$$

where  $f_n^{up}(i, j)$  is the upper bound of VddL switch number in the fanout cone of switch  $j$  in routing tree  $\mathcal{R}_i$ .

**THEOREM 1.**  $F_n^{up}(i)$  is the VddL switch number upper bound of routing tree  $\mathcal{R}_i$ .

**PROOF.** It is easy to show that  $\frac{s_{ik}}{C_{ik}} \cdot c_{ij}$  is the upper bound of the VddL switch likelihood in each path from source  $Src_i$  to the  $k$ th sink  $Sink_k$  in routing tree  $\mathcal{R}_i$ , since the given timing slack  $S_k$  will be exceeded if more than  $\frac{s_{ik}}{C_{ik}} \cdot c_{ij}$  switches in this path are powered by VddL. By summing up all VddL switches under a fanout cone of switch  $j$ ,  $f_n^{up}(i, j)$  is always the upper bound of the VddL switch number, since all potential VddL switches are counted at least once.  $\square$

Figure 5 shows an example to calculate  $f_n^{up}(i, B4)$  for switch  $B4$  in routing tree  $\mathcal{R}_i$ . The slacks assigned in sinks are redistributed along each switch-to-sink path, respectively. In this example, the paths from  $B4$  to sink  $S_1$  and  $S_2$  are considered, respectively. For  $B4$  to  $S_2$  path (Figure 5(b)), there are 3 switches lying on it, therefore  $L_{i,2} = 3$ . Suppose the delay increase when setting all three switches to VddL is 12, namely,  $D_{i,2} = 12$ . Given 10-unit slack assigned in sink  $S_2$ , we can represent this timing slack in terms of switch number,  $s_{i,2} = 5/2$ , based on Eq. (3). The VddL possibility for switch  $B4$  with respect to sink  $S_2$  is  $5/2 \cdot 16/(16 + 16 + 8) = 1$ . Similarly, the VddL possibility for switch  $B4$  with respect. to sink  $S_1$  (Figure 5(a)) is  $6/9 \cdot 2 \cdot 16/(16 + 8) = 8/9$ . Therefore  $f_n^{up}(i, B4) = 1 + 8/9 = 17/9$ .

We then estimate the power reduction for  $\mathcal{R}_i$ . The upper bound  $P_{dr}^{up}(i)$  of dynamic power reduction of the tree  $\mathcal{R}_i$  is estimated as the sum of the dynamic

power reduction of each switch in  $\mathcal{R}_i$  and can be expressed as

$$P_{dr}^{up}(i) = 0.5 f_{clk} \cdot (VddH^2 - VddL^2) \sum_{j=0}^{N_s(i)-1} f_n^{up}(i, j) \cdot f_s(i, j) \cdot c_{ij}. \quad (7)$$

Similarly, the upper bound  $P_{lr}^{up}(i)$  of leakage power reduction of  $\mathcal{R}_i$  is the sum of the leakage power reduction of each switch in  $\mathcal{R}_i$  and can be expressed as

$$P_{lr}^{up}(i) = \sum_{j=0}^{N_s(i)-1} f_n^{up}(i, j) \cdot \Delta P_s(i, j), \quad (8)$$

where  $\Delta P_s(i, j)$  is the leakage power difference of the  $j$ th switch in  $\mathcal{R}_i$  between VddH and VddL. Wire segments with different lengths are usually driven by switches with different sizes. The accuracy of our upper-bound-based power reduction estimation will be verified in Section 3.3.

### 3.2 LP-Based Problem Formulation

To formulate budgeting as a mathematical programming problem, we need to explicitly express the constraints and objective function. In this problem, the timing constraints require that the maximal arrival time at  $\mathcal{PO}$  with respect to  $\mathcal{PI}$  is at most  $T_{spec}$ , namely, for all paths from  $\mathcal{PI}$  to  $\mathcal{PO}$ , the sum of edge delays in each path  $p$  must be at most  $T_{spec}$ . As the number of paths from  $\mathcal{PI}$  to  $\mathcal{PO}$  can be exponential, the direct path-based formulation, on timing constraints is impractical for analysis and optimization. Alternatively, we use the net-based formulation which partitions the constraints on path delay into constraints on delay across circuit elements or routing. Let  $a(v)$  be the arrival time for vertex  $v$  in  $\mathcal{G}$  and the timing constraints become

$$a(v) \leq T_{spec} \quad \forall v \in \mathcal{PO} \quad (9)$$

$$a(v) = 0 \quad \forall v \in \mathcal{PI} \quad (10)$$

$$a(u) + d(u, v) \leq a(v) \quad \forall u \in \mathcal{V} \wedge v \in \mathcal{FO}_u, \quad (11)$$

where  $\mathcal{V}$  is the set of vertices in  $\mathcal{G}$ ,  $d(u, v)$  is the delay from vertex  $u$  to  $v$ , and  $\mathcal{FO}_u$  is the set of fanout vertices of  $u$ .

The objective function is to maximize the estimation of interconnect power reduction given by Eqs. (7) and (8). It can be expressed as

$$\begin{aligned} \text{Maximize} \quad & \sum_{i=0}^{N_r-1} \sum_{j=0}^{N_s(i)-1} 0.5 f_{clk} (VddH^2 - VddL^2) f_n^{up}(i, j) f_s(i, j) c_{ij} \\ & + \sum_{i=0}^{N_r-1} \sum_{j=0}^{N_s(i)-1} f_n^{up}(i, j) \Delta P_s(i, j). \end{aligned} \quad (12)$$

We then modify the timing constraints (11) as follows. For the edges corresponding to routing in  $\mathcal{G}$ , the constraints considering slack can be expressed as

$$\begin{aligned} S_{ik} &= a(p_{ik}) - a(p_{i0}) - d(p_{i0}, p_{ik}) \\ 0 &\leq i < N_r \wedge \forall p_{ik} \in \mathcal{FO}_{p_{i0}}, \end{aligned} \quad (13)$$

where vertex  $p_{i0}$  is the source of  $\mathcal{R}_i$  in  $\mathcal{G}$ , vertex  $p_{ik}$  is the  $k$ th sink of  $\mathcal{R}_i$  in  $\mathcal{G}$ ,  $S_{ik}$  is the slack allocated to the  $k$ th sink in  $\mathcal{R}_i$ , and  $d(p_{i0}, p_{ik})$  is the delay from  $p_{i0}$  to  $p_{ik}$  in  $\mathcal{R}_i$  using VddH. For the edges other than routing in  $\mathcal{G}$ , the constraints can be expressed as

$$a(u) + d(u, v) \leq a(v) \quad \forall u \in \mathcal{V} \wedge u \notin \mathcal{SRC} \wedge v \in \mathcal{FO}_u, \quad (14)$$

where  $\mathcal{SRC}$  is a subset of  $\mathcal{V}$  and gives the set of vertices corresponding to routing tree sources.

We formulate the *timing slack allocation problem* using objective function (12), slack upper-bound constraints (2), and timing constraints (9), (10), (13), and (14). It is easy to verify that all the constraints are linear, and the objective function (12) is also linear. Hence we have the following theorem.

**THEOREM 2.** *The timing slack allocation problem with objective function (12), and constraints (2), (9), (10), (13), and (14), is a linear programming (LP) problem.*

Time slack is allocated to each routing tree by solving the timing slack allocation problem. Then the net-level bottom-up assignment from Lin and He [2006] is modified to consider mixed-length interconnects and to leverage the allocated slack.

The procedure of the bottom-up assignment is as follows. For each tree  $\mathcal{R}_i$ , VddH is first assigned to all switches in  $\mathcal{R}_i$ . We then iteratively perform the following steps in bottom-up fashion. We assign VddL to a candidate switch and mark the switch as “tried”. After updating the circuit timing, we reject the assignment and restore the Vdd level of the switch to VddH if the delay increase at any sink exceeds the allocated slack. The iteration terminates when there is no candidate switch in  $\mathcal{R}_i$ .

Note that our upper-bound-based power estimation may give an overestimation of power reduction and the number of VddL switches, and the net-level bottom-up Vdd assignment guarantees the legalization of final solutions.

### 3.3 Experimental Results for LP-Based Dual-Vdd Budgeting

We first verify the effectiveness of the upper-bound-based power reduction estimation. Tested on 14 industrial designs, two cases are compared in Figure 6 as follows: (i) estimated power reduction calculated by the upper-bound-based estimation (12) and (ii) the simulated power reduction collected by fpgaEva-LP2 [Lin et al. 2005a] for the final designs with Vdd-level assignment for all interconnect switches. As shown in the figure, the estimation values provide a consistent upper bound of power reduction and are well correlated to the simulated power reduction values, which means that the upper-bound-based power reduction estimation model provides good guidance for power-aware timing slack budgeting in practice.

Table II shows the interconnect power reduction achieved by LP-based dual-Vdd timing slack budgeting formulation for MCNC benchmarks and industrial designs, respectively. Compared to single-Vdd FPGA designs, the dual-Vdd-level assignment algorithm reduces interconnect power about 50% and assigns

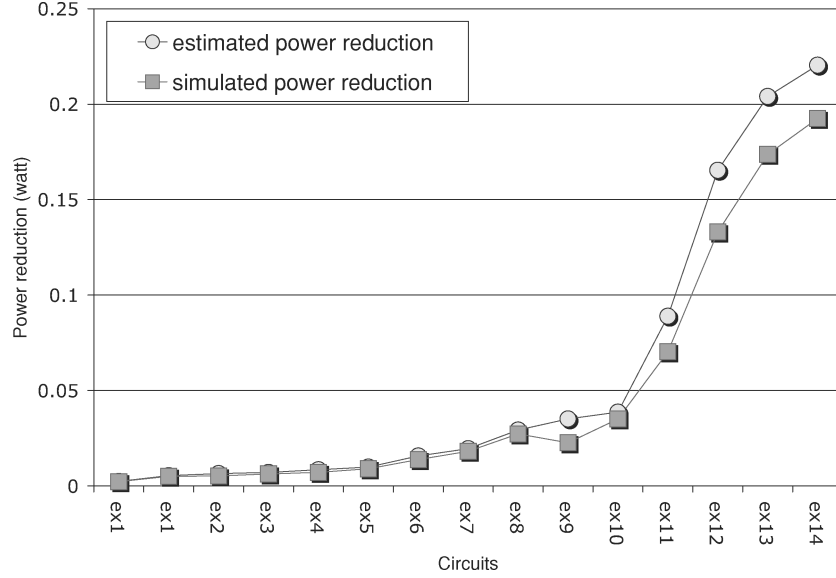


Fig. 6. Comparison between the estimated and real values for power reduction.

85% VddL interconnect switches on average for both groups of test circuits. Figure 7 breaks down the total interconnect power into dynamic and leakage contributors and shows the effectiveness of the algorithm for interconnect dynamic- and leakage power reduction, respectively.

### 3.4 Network-Flow-Based Problem Formulation

The runtime of timing slack budgeting in the LP-based algorithm can be very long for large circuits, mainly due to the expensive computational time of linear programming (as will be shown in Section 3.5). Fortunately, the proposed upper-bound-based power reduction estimation (see Section 3.1) removes the nonlinear term “min” in the lower bound formulation in Lin and He [2006], enabling us to reformulate this problem as a min-cost network flow problem and to solve it with significant speedup. Similar network flow formulation has been used for timing budgeting in high-level synthesis [Ghiasi et al. 2004].

The objective function (12) can be rewritten by merging the coefficient of the slack  $S_{ik}$  of  $k$ th sink in  $\mathcal{R}_i$ ,

$$\text{Maximize} \quad \sum_{i=0}^{N_r-1} \sum_{k=0}^{N_k(i)-1} W_{ik} \cdot S_{ik} = \sum_{\forall \text{Sink}} W_{ik} \cdot S_{ik} \quad (15)$$

$$W_{ik} = \sum_{\forall j \in \mathcal{UBC}_{ik}} [0.5 f_{clk} (VddH^2 - VddL^2) c_{ij} f_s(i, j) + \Delta P_s(i, j)] \cdot \frac{c_{ij} \cdot D_{ik}}{(C_{ik} \cdot l_{ik})}, \quad (16)$$

where set  $\mathcal{UBC}_{ik}$  includes all switches whose fanout cones contain sink  $k$  in routing tree  $\mathcal{R}_i$ . Note that  $W_{ik}$  are constant coefficients in the objective functions.

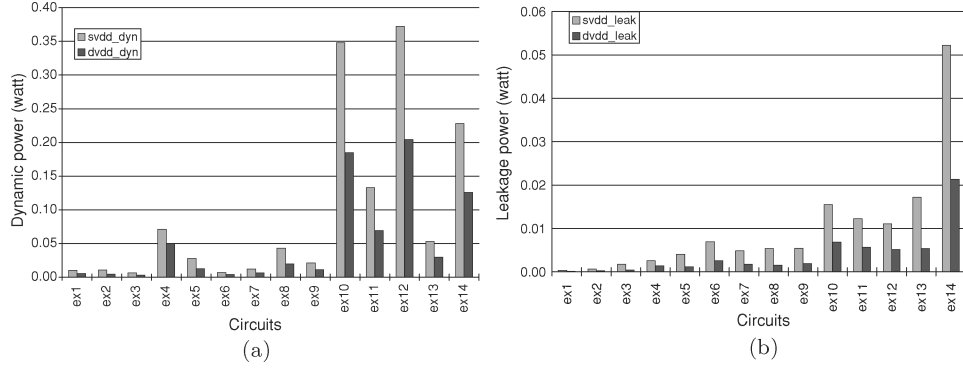


Fig. 7. Comparison of single-Vdd and dual-Vdd FPGAs for (a) interconnect dynamic power (b) interconnect leakage power.

Since  $W_{ik} > 0$  for all sinks, we can restrict timing constraint (13) as the following equation to maximize the objective function.

$$S_{ik} = a(p_{ik}) - a(p_{i0}) - d(p_{i0}, p_{ik}), \quad 0 \leq i < N_r \wedge \forall p_{ik} \in \mathcal{FO}_{p_{i0}} \quad (17)$$

After substituting  $S_{ik}$  using (17) and rearrangement, objective function (15) can be expressed as

$$\text{Maximize} \quad \sum_{i=0}^{N_r-1} \sum_{k=0}^{N_k(i)-1} W_{ik} \cdot [a(p_{ik}) - a(p_{i0}) - d(p_{i0}, p_{ik})]. \quad (18)$$

Similarly, slack bound constraint (2) can be rewritten as

$$a(p_{i0}) - a(p_{ik}) \leq -d(p_{i0}, p_{ik}) \quad (19)$$

$$a(p_{ik}) - a(p_{i0}) \leq d(p_{i0}, p_{ik}) + D_{ik}. \quad (20)$$

We then merge the timing constraints (19) and (14) into the general expression (11).

A virtual input node ( $SI$ ) and a virtual output node ( $SO$ ) are added into  $\mathcal{G}$  to connect all nodes in  $\mathcal{PI}$  and  $\mathcal{PO}$ , respectively. All edges connected to  $SI$  and  $SO$  have zero delay. We add a backward edge  $e(p_{ik}, p_{i0})$  for each source-sink pair in  $\mathcal{R}_i$ . A delay of  $-(d(p_{i0}, p_{ik}) + D_{ik})$  is associated with  $e(p_{ik}, p_{i0})$  to represent the slack upper bound. A virtual edge  $e(SO, SI)$  with delay  $-T_{spec}$  is then added. All constraints can now be represented by edges in  $\mathcal{G}$ . For example, edge  $e(u, v)$  with delay  $d(u, v)$  represents constraint  $a(u) - a(v) < -d(u, v)$ .

To represent the objective function (18) in  $\mathcal{G}$ , we associate a weight  $w_{uv}$  in each edge  $e(u, v)$ . For those edges  $e(p_{i0}, p_{ik})$  corresponding to routing, let  $w_{p_{i0}p_{ik}} = W_{ik}$ . For other edges, let  $w_{uv} = 0$ . The objective function (18) can then be rewritten as

$$\text{Maximize} \quad \sum_{v \in V} a(v) \left( \sum_{u \in \mathcal{FI}_v} w_{uv} - \sum_{u \in \mathcal{FO}_v} w_{vu} \right) - \sum_{i=0}^{N_r-1} \sum_{k=0}^{N_k(i)-1} d(p_{i0}, p_{ik}), \quad (21)$$

where  $\sum_{i=0}^{N_r-1} \sum_{k=0}^{N_k(i)-1} d(p_{i0}, p_{ik})$  is a constant and can be removed from the objective function (21), and  $\mathcal{FI}_v/\mathcal{FO}_v$  is the fanin/fanout set of vertex  $v$ .

Table III. Runtime(s) Comparison for LP-Based and Network-Flow-Based Timing Slack Budgeting

MCNC			Industrial		
circuit	LP-based	Netflow-based	circuit	LP-based	Netflow-based
clma	1499.2 (172 x)	8.7	ex5	0.7 (3x)	0.2
tseng	0.9 (9 x)	0.1	ex6	2.9 (4x)	0.7
dsip	2.2 (7 x)	0.3	ex7	2.3 (3x)	0.7
diffeq	3.2 (11 x)	0.3	ex8	1.4 (4x)	0.3
s298	42.9 (215 x)	0.2	ex9	3.6 (4x)	1.0
bigkey	5.8 (15 x)	0.4	ex10	4.4 (5x)	0.9
elliptic	11.2 (14 x)	0.8	ex11	8.2 (5x)	1.6
frisc	23.6 (16 x)	1.5	ex12	2.4 (4x)	0.6
s38584	13.0 (5 x)	2.7	ex13	3.1 (4x)	0.7
s38417	22.6 (7 x)	3.3	ex14	16.9 (4x)	4.5
harmean	4.3 (11 x)	0.4	harmean	2.3 (4x)	0.6

For the optimization problem with constraints (11) and (20) and objective function (21), its dual problem is

$$\begin{aligned}
& \text{Minimize} && \sum_{e(i,j) \in \mathcal{E}} (d(i,j) + D_{ij}) \cdot z_{ij} - d(i,j) \cdot y_{ij} \\
& \text{such that} && \sum_{e(k,i) \in \mathcal{E}} (y_{ki} - z_{ki}) - \sum_{e(i,j) \in \mathcal{E}} (y_{ij} - z_{ij}) = \rho_i \\
& && \rho_i = \sum_{j \in \mathcal{FI}_i} w_{ji} - \sum_{k \in \mathcal{FO}_k} w_{ik} \\
& && y_{ij}, z_{ij} \in \mathbf{R}_+.
\end{aligned} \tag{22}$$

**THEOREM 3.** *The dual problem of the timing slack allocation problem (22) is a min-cost network flow problem.*

**PROOF.** To verify that the aforesaid dual problem is a min-cost network flow problem on  $\mathcal{G}$ ,  $y_{ij}$  is the flow along  $e(i,j)$  with cost  $-d(i,j)$ ,  $z_{ij}$  is the flow along  $e(j,i)$ , which corresponds to routing and is associated with cost  $d(i,j) + D_{ij}$ . Obviously, no negative cycle is introduced by the backward edges.  $\rho_i$  is the demand in each vertex. Note that  $\sum_{i \in \mathcal{V}} \rho_i = 0$  is satisfied, as required in the min-cost network flow problem.  $\square$

After solving the min-cost network flow problem, we can get the solutions for variables  $y_{ij}$  and  $z_{ij}$ . Similar to Ghiasi et al. [2004], we can calculate the solution of the primal problem. We first construct the residual graph  $\mathcal{G}'(\mathcal{V}, \mathcal{E}')$  from the original  $\mathcal{G}$ . For any edge  $e(i,j)$  in  $\mathcal{G}'$  with nonzero flow, there are two edges  $e(j,i)$  and  $e(i,j)$  in  $\mathcal{G}'$ . The cost of each backward edge  $e(j,i)$  is  $d(i,j)$ , and is equal to the complement of the forward-edge cost. Let  $\delta_i$  be the shortest distance from  $SI$  to vertex  $i$  in  $\mathcal{G}'$ . It has been proved in Ghiasi et al. [2004] that  $a(i) = -\delta_i$  is an optimal solution to the primal problem.

### 3.5 Runtime Comparison for LP- and Network-Flow-Based Algorithms

We use the push-relabel algorithm [Goldberg 1997] for the min-cost flow problem and the Bellman-Ford algorithm [Rivest et al. 1990] for the shortest path problem in our implementation. Table III compares the runtime for timing slack budgeting by the LP-based algorithm and network-flow-based algorithm, and the harmonic mean is used to avoid the case that a few very large values

dominate the rest and skew the results. The network-flow-based algorithm achieves over 11x speedup on MCNC benchmarks compared to the LP-based one and more speedup can be expected when the circuit scale becomes larger. Note that interior point method is used in mosek [Mosek 2006] for solving the LP problem and that its runtime is highly related to the structure of the problem. For certain medium circuits, such as “s298” in the MCNC benchmark, the runtime for solving the LP problem is 2x longer than “s38417”, which is 4x bigger than “s298”. Contrary to the instability of computational time for solving an LP problem, the network flow formulation always has a consistently short runtime.

#### 4. SIMULTANEOUS RETIMING AND DUAL VDD BUDGETING

The dual-Vdd timing slack budgeting algorithm proposed in Section 3 has achieved significant interconnect power reduction by exploring the combinational components of the circuits, whereas the search space is restricted by the given circuit topology (e.g., the placement of flip-flops (FF)). As illustrated in Figure 2, more power reduction can be achieved by simultaneously performing retiming and dual-Vdd timing slack budgeting.

##### 4.1 MILP-Based Problem Formulation for Retiming

A directed retiming graph is constructed to model the circuit for sequential timing analysis. Vertices represent the inputs/outputs of basic circuit elements such as LUTs. Edges are added between the inputs of combinational logic elements (e.g., LUTs) and their outputs, and between the connected pins specified by the circuit netlist. Each edge is annotated with the delay  $d(e)$  required to pass through the circuit element or routing and the weight  $w(e)$ , which is the number of FFs inserted in it. We use  $\mathcal{PI}$  and  $\mathcal{PO}$  to represent the set of primary inputs and outputs, respectively.

The MILP formulation for retiming synchronous circuits is originally presented Leiserson and Saxe [1991] to minimize clock period. We first extend the MILP formulation to consider interconnect delay and get the following theorem.

**THEOREM 4.** *Let  $G = (V, E, d, w)$  be a synchronous circuit, and let target clock period  $\Phi$  be a positive real number. Then there exists a retiming  $r$  of  $G$  such that  $\Phi(G_r) \leq \Phi$  if and only if there exists an assignment of real values  $a(v)$  and an integer value  $r(v)$  to each vertex  $v \in V$  such that the following conditions are satisfied.*

$$-a(v) \leq -\max_{u \in \text{Fanin}(v)} d(u, v), \quad \forall v \in V \quad (23)$$

$$a(v) \leq \Phi, \quad \forall v \in V \quad (24)$$

$$r(u) - r(v) \leq w(e), \quad \forall e(u, v) \in E \quad (25)$$

$$a(u) - a(v) \leq -d(u, v), \quad \forall e(u, v) \text{ s.t. } r(u) - r(v) = w(e), \quad (26)$$

where  $G_r$  is the retimed circuit and  $\Phi(G_r)$  is the clock period of  $G_r$ .

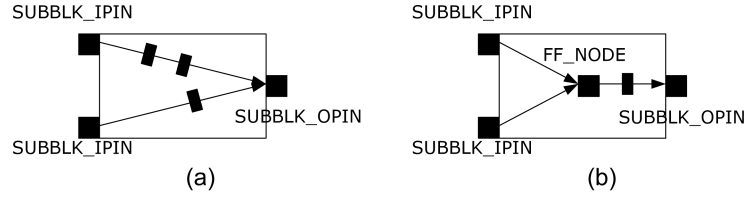


Fig. 8. A dummy node to consider FF sharing. In (b), SUBBLK\_IPIN to FF\_NODE edges are used to annotate the intrinsic delay of a LUT, and node FF\_NODE denotes the output of a LUT when FFs are inserted in FF\_NODE to SUBBLK\_OPIN edge.

To linearize all constraints, a new variable  $R(v) = r(v) + a(v)/\Phi$  is introduced, and the previous retiming constraints can be rewritten as

$$r(v) - R(v) \leq - \max_{u \in Fanin(v)} d(u, v)/\Phi, \quad \forall v \in V \quad (27)$$

$$R(v) - r(v) \leq 1, \quad \forall v \in V \quad (28)$$

$$r(u) - r(v) \leq w(e), \quad \forall e(u, v) \in E \quad (29)$$

$$R(u) - R(v) \leq w(e) - d(u, v)/\Phi, \quad \forall e(u, v) \in E. \quad (30)$$

In the next two subsections, we propose additional constraints to consider placement and FF binding constraints.

#### 4.2 Flip-Flop Constraints in Retiming

As described in Section 2.4, delay values are associated with timing edges. An LC is then represented by several nodes and edges in the retiming graph (see Figure 8). Figure 8(a) connects the inputs of the LUT to the output of the LC directly in the retiming graph. The FF number on each such edges may be different after retiming, which leads to an illegal FF assignment in the FPGA, as the FF number in each input-output pair of an LC should be the same with a legal retiming. To tackle this problem, we add a dummy FF node to each LC (including the LC originally in combinational mode), and constrain that FFs can only be inserted at edges from FF node to LC output node, as shown in Figure 8(b).<sup>6</sup> To ensure that a LUT drives only FFs within the same cluster after retiming, the total FFs used within one cluster cannot exceed the number of available FFs in the cluster. Therefore, we have the constraints

$$\sum_{e(u,v) \in E_{C_i}} w(e) + r(v) - r(u) \leq S_{C_i}, \quad \forall C_i \quad (31)$$

$$r(v) - r(u) = 0, \quad \forall e(u, v) \in E/E_{ff}, \quad (32)$$

<sup>6</sup>The proposed retiming graph only supports a subset of the operating modes of the LC in Figure 3, since it restricts the placement of FFs by excluding the case of FF insertion in non-FF\_NODE-to-SUBBLK\_OPIN edges. This restriction tries to minimize the disturbance of the interconnects within a CLB after retiming. But our formulation can be extended to handle FF insertion in arbitrary timing edges by updating the final FF placement phase (Section 4.4). Note that the two outputs in Figure 3 enable us to use available FFs for the case that there is more than one FF inserted in the FF\_NODE-to-SUBBLK\_OPIN edge. Without two outputs, the FF cascade must be implemented through WIRE-LUT, which increases the logic area and delay.

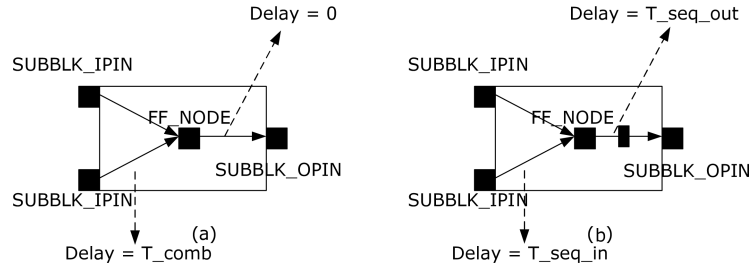


Fig. 9. (a) Combinational mode of LC; (b) sequential mode of LC.

where set  $E_{ff}$  comprises all edges from FF nodes to LC output nodes, set  $E_{Ci}$  comprises all  $E_{ff}$  edges in cluster  $i$ , and  $S_{Ci}$  denotes the FF number in cluster  $i$ .

There are two runtime modes of an LC, namely combinational and sequential modes. Figure 9 shows the LC input delay (from LC input to FF node) and LC output delay (from FF node to LC output). When the LUT in an LC drives at least one FF, it runs in a sequential mode. The LC input delay is  $T_{seq\_in}$  and its output delay is  $T_{seq\_out}$ . Otherwise, it works in a combinational mode, and the LC input and output delays are  $T_{comb}$  and 0, respectively.

Formally, suppose the LC input and output delay are  $D_{in}$  and  $D_{out}$ , respectively. Then we have

$$D_{in} = \begin{cases} T_{seq\_in}, & \text{if } w'(e) > 0 \\ T_{comb}, & \text{if } w'(e) = 0 \end{cases}$$

$$D_{out} = \begin{cases} T_{seq\_out}, & \text{if } w'(e) > 0 \\ 0, & \text{if } w'(e) = 0, \end{cases} \quad (33)$$

where  $w'(e) = w(e) + r(v) - r(u)$  is the FF number in edge  $e(u, v)$  after retiming [Leiserson and Saxe 1991].

To consider FF delay constraints (33) explicitly, we can add the following inequalities into the MILP formulation.

$$\begin{aligned} T_{comb} &\leq D_{in} \leq T_{seq\_in} \\ D_{in} &\geq T_{seq\_in} \cdot w_{bin}(e) \\ 0 &\leq D_{out} \leq T_{seq\_out} \\ D_{out} &\geq T_{seq\_out} \cdot w_{bin}(e) \\ w_{bin} &\leq 1, \\ w_{bin} &\leq w'(e), \end{aligned} \quad (34)$$

where the first inequality guarantees that no timing violation occurs, since the effective LC input delay  $D_{in}$  is always larger than  $T_{comb}$ , which is a lower bound, and the second inequality ensures that  $D_{in}$  is set as  $T_{seq\_in}$  when the LC is working in sequential mode, namely,  $w'(e) = 0$ . Furthermore,  $w_{bin}(e)$  is an intermediate binary variable which is equal to  $\min(1, w'(e))$ .

We incorporate these constraints into our MILP formulation, and rewrite (27) and (30) for timing edges within an LC as

$$\begin{aligned} r(v) - R(v) &\leq -D_{in,out}/c, \quad \forall e(u, v) \in E_{in,out} \\ R(u) - R(v) &\leq w(e) - D_{in,out}/c, \quad \forall e(u, v) \in E_{in,out}, \end{aligned} \quad (35)$$

where  $E_{in}$  and  $E_{out}$  are the LC input and LC output edge sets, respectively.

#### 4.3 MILP-Based Problem Formulation for Simultaneous Retiming and Budgeting

The following important observation enables us to consider timing slack explicitly in our formulation.

**OBSERVATION 1.** *The real value  $a(v)$  assigned in node  $v$  in Theorem 2 is its arrival time after retiming.*

Based on this observation, the timing slack in edge  $e(u, v)$  can be expressed as

$$\begin{aligned} S(u, v) &= a(v) - a(u) - d(u, v) \\ &= [R(v) - R(u) + r(u) - r(v)] \cdot c - d(u, v). \end{aligned} \quad (36)$$

Based on (36), for the path from the source node  $Src_i$  to the  $k$ th sink  $Sink_k$  in routing tree  $\mathcal{R}$ , the slack constraints (13) can be rewritten as follows by setting  $u = Src_i$  and  $v = Sink_k$ .

$$\begin{aligned} S_{ik} &= [R(p_{ik}) - R(p_{i0}) + r(p_{i0}) - r(p_{ik})] \cdot c - d(p_{i0}, p_{ik}) \\ 0 &\leq i < N_r \wedge \forall p_{ik} \in \mathcal{FO}_{p_{i0}} \end{aligned} \quad (37)$$

If we substitute  $S_{ik}$  in (3) with (37), use the power estimation (12) in Section 3 as the objective function, employ retiming and delay constraints (27), (28), (29), (30), (35), (36), and FF number constraints (32), (31) plus the slack bound constraints (2), we can consider timing and slack budgeting simultaneously. It is easy to verify that all the constraints and objective functions are linear. Hence we have the following theorem.

**THEOREM 5.** *The poststage simultaneous retiming and slack budgeting problem with objective function (12) and constraints (27), (28), (29), (30), (35), (36), (32), (31), and (2), is a mixed integer linear programming (MILP) problem.*

#### 4.4 Detailed FF Placement and Vdd Assignment for Interconnect Switches

After solving the MILP problem, we perform detailed FF placement, the purpose of which is to embed the retiming solution described in the retiming graph in the real FPGA circuit. We first assign FFs in the retiming solution to feasible locations. Since our MILP formulation guarantees feasibility of the retiming solution, we can always find a valid embedding for the retimed circuit. When more than one FF is inserted in an FF\_NODE to SUBBLK\_OPIN edge, the detailed FF placer will seek an available FF slot within the same CLB and update the local interconnects accordingly, as shown in Figure 10.

Following the FF placement phase, we perform the Vdd-level assignment for interconnect switches as described in Section 3.2. The local routing delay

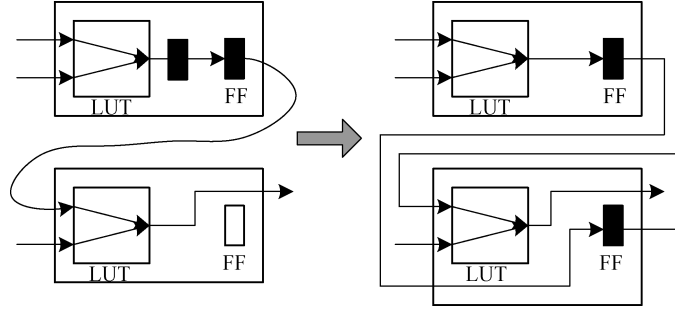


Fig. 10. Detailed FF placement based on retiming solution.

within a CLB is considered explicitly during the static timing analysis in the bottom-up Vdd-level assignment phase to reflect the change of local netlist after retiming and detailed FF placement. However, in our MILP formulation we do not consider the local routing delay within CLBs due to the difficulty to model it in the MILP formulation. Therefore, all possible timing slack estimation errors due to the simplification in MILP formulation are counted in the final design performance evaluation.

#### 4.5 Speedup MILP Formulation by Relaxation and Legalization

Solving the large-scale MILP problem is intractable because of its NP-hardness. Instead of solving the the MILP problem directly, we can consider its LP relaxation by allowing fractional values for  $r(v)$ ,  $\forall v \in V$ . The fractional retiming allows us to adopt the concept of *sequential arrival time* or *l-value* proposed in Pan et al. [1998]. The l-value of node  $v$  is defined recursively as

$$l(v) = \max\{l(u) - \Phi \cdot w(e) + d(u, v), \forall \text{ edge } e(u, v) \in E\}, \quad (38)$$

where  $\Phi$  is the target clock period, and  $w(e)$  and  $d(u, v)$  are the FF number and delay in edge  $e(u, v)$ , respectively.

The fractional retiming solution  $s(v)$  for node  $v$  under clock period  $c$  can be represented as  $s(v) = \frac{l(v)}{c}$ . By introducing sequential arrival time and the fractional retiming, the MILP formulation in Theorem 1 can be relaxed, as

$$l(v) \leq \Phi, \quad \forall v \in PO \quad (39)$$

$$l(u) - l(v) \leq \Phi \cdot w(e) - d(u, v) \quad (40)$$

$$s(v) = \frac{l(v)}{c}, \quad (41)$$

where we use sequential arrival time  $l(v)$  and fractional retiming label  $s(v)$  to replace the combinational arrival time  $a(v)$  and integer retiming label  $r(v)$  in Theorem 1, respectively. Consequently, the corresponding variables in the postlayout FF constraints and objective function presented in Sections 4.2 and 4.3 are substituted in the same manner. Note that the constraints in the MILP formulation (see Section 4.3) are simplified under the fractional retiming and the auxiliary variables  $R(v)$  are no longer needed.

The fractional retiming solution  $s(v)$  can be rounded to the integer retiming label  $r(v)$  in the following way.

$$r(v) = \begin{cases} 0, & v \text{ is a PI or a PO} \\ \lceil s(v) \rceil - 1, & \text{otherwise} \end{cases} \quad (42)$$

Speedup can be achieved via LP relaxation of the MILP formulation. On the other hand, side-effects, such as retiming constraint violations and performance degradation, brought by the relaxation and rounding procedure need to be addressed and justified.

Pan et al. [1998] has proved that Eq. (42) guarantees a legal retiming under basic retiming constraints (25). However, the rounding procedure for the fraction solution may cause constraint violation for FF number constraints (31) and (32). To solve FF number constraint violations, we adopt the local adjustment algorithm (see Algorithm 1) presented in Lin and Zhou [2004] to legalize the rounded retiming results by pushing forward or backward FFs to their feasible locations. The input of the algorithm is the retiming graph  $G$  and node  $v$  of timing edge  $e(u, v)$  that violate the FF number constraint.

---

**Algorithm 1.** Legalization ( $G, v$ ) [Lin and Zhou 2004]

---

```

 $Q \leftarrow \{v\};$ 
While ( $Q \neq \emptyset$ ) do
   $u \leftarrow \text{dequeue}(Q);$ 
  For each  $e(= (x, u) \text{ or } (u, x)) \in E$  do
    If ( $((e \in E/E_{ff}) \wedge (r(x) \neq r(u))) \vee (w_r(e) < 0)$ ) then
       $r(x) \leftarrow r(x) + 1;$ 
   $Q \leftarrow Q \cup \{x\};$ 

```

---

The legalization procedure will increase the clock period if the FF number violation occurs in the critical path, as this procedure may violate the FF delay constraints (34). In fact, we find that the performance degradation is limited for most test circuits, which will be justified by both Theorem 6 and the experimental results. Practically, the FF delay is small compared to the interconnect delay. Theorem 6 shows that the fractional retiming by LP relaxation will not increase the clock period if we do not consider the FF delay (setup and hold time).

**THEOREM 6.** *Without considering FF delay constraints (34), the LP relaxation and rounding procedure based on (42) for the MILP problem preserves the clock period.*

**PROOF.** Consider path  $p(u \rightarrow v)$  with delay  $d(p) \geq \Phi$ , where  $\Phi$  is the target clock period. Based on (40), we have

$$\sum_{\forall e(x,y) \in p(u \rightarrow v)} l(x) - l(y) \leq \sum_{\forall e(x,y) \in p(u \rightarrow v)} \Phi \cdot w(e) - d(x, y). \quad (43)$$

Arranging (43), we have

$$l(v) \geq l(u) - \Phi \cdot w(p) + d(p). \quad (44)$$

Table IV. Power and Delay Comparison for Sequential Approach and Simultaneous Approaches (MILP-based and LP-based)

circuit	Sequential	Simultaneous		
		MILP	LP	best
bigkey	0.1038	0.0739 (−28.8%)	—	0.0739 (−28.8%)
clma	0.0837	0.0765 (−8.6%)	—	0.0765 (−8.6%)
diffeq	0.0061	0.0058 (−5.6%)	—	0.0058 (−5.6%)
dsip	0.12	0.1202 (+0.1%)	0.1202 (+0.1%)	0.1202 (+0.1%)
elliptic	0.0221	0.0208 (−5.8%)	0.0208 (−5.8%)	0.0208 (−5.8%)
frisc	0.0143	0.0128 (−10.1%)	0.012 (−16.0%)	0.012 (−16.0%)
s298	0.0338	0.0307 (−9.1%)	0.0307 (−9.1%)	0.0307 (−9.1%)
s38417	0.096	0.0922 (−3.9%)	—	0.0922 (−4.0%)
s38584.1	0.0645	0.0615 (−4.6%)	0.0605 (−6.2%)	0.0605 (−6.2%)
tseng	0.0065	0.0062 (−4.4%)	0.0063 (−4.0%)	0.0062 (−4.4%)
ave	0.0551	0.0501 (−8.1%)	0.04175 (−6.9%)	0.0500 (−8.2%)
ex1	0.0054	0.0053 (−0.9%)	0.0053 (−0.9%)	0.0053 (−0.9%)
ex2	0.0048	0.0047 (−3.0%)	0.0047 (−3.0%)	0.0047 (−3.0%)
ex3	0.0033	0.0032 (−4.7%)	—	0.0032 (−4.7%)
ex4	0.0509	0.0502 (−1.3%)	—	0.0502 (−1.3%)
ex5	0.0138	0.0131 (−4.6%)	0.0131 (−4.6%)	0.0131 (−4.6%)
ex6	0.0066	0.0061 (−7.0%)	0.0061 (−7.1%)	0.0061 (−7.1%)
ex7	0.008	0.0076 (−5.9%)	0.0076 (−5.9%)	0.0076 (−5.9%)
ex8	0.021	0.0197 (−6.3%)	0.0199 (−5.6%)	0.0197 (−6.3%)
ex9	0.0129	0.0125 (−3.6%)	0.0125 (−3.6%)	0.0125 (−3.6%)
ex10	0.1917	0.1855 (−3.2%)	0.186 (−3.0%)	0.1855 (−3.2%)
ex11	0.0748	0.0687 (−8.2%)	0.0716 (−4.2%)	0.0687 (−8.2%)
ex12	0.2096	0.2079 (−0.8%)	0.2079 (−0.8%)	0.2079 (−0.8%)
ex13	0.0352	0.0342 (−2.9%)	0.0342 (−2.9%)	0.0342 (−2.9%)
ex14	0.1472	0.1429 (−2.9%)	—	0.1429 (−2.92%)
ave	0.0639	0.0545 (−3.8%)	0.0517 (−3.8%)	0.0514 (−3.8%)

Dividing both sides by  $\Phi$  and taking the ceiling, we have

$$r(v) + 1 \geq r(u) + 1 - w(p) + \left\lceil \frac{d(p)}{\Phi} \right\rceil \geq r(u) + 1 - w(p) + 1. \quad (45)$$

Therefore, the number of FFs in path  $p$  is  $w(p) + r(v) - r(u) \geq 1$ , which means that for any path  $p$  such that  $d(p) \geq \Phi$ , there is at least one FF on it. Therefore, the target clock period  $\Phi$  is preserved.  $\square$

#### 4.6 Experimental Results for Simultaneous Retiming and Budgeting

Two CAD flows, namely the sequential approach (min-clock retiming followed by dual-Vdd budgeting) and simultaneous approach (min-clock retiming followed by simultaneous retiming and budgeting) are compared. For both flows, the target clock period is the same and produced by min-clock retiming. For the simultaneous approach, the MILP- and LP-based implementations are compared to each other, as well as to the sequential approach.

Table IV shows the interconnect power reduction achieved by the sequential approach (column Sequential), MILP-based- (column Simultaneous-MILP), and LP-relaxation-based (column Simultaneous-LP) simultaneous approaches, respectively, and the better solutions between them are shown in Column Simultaneous-best. Compared to the sequential approach, the MILP-based

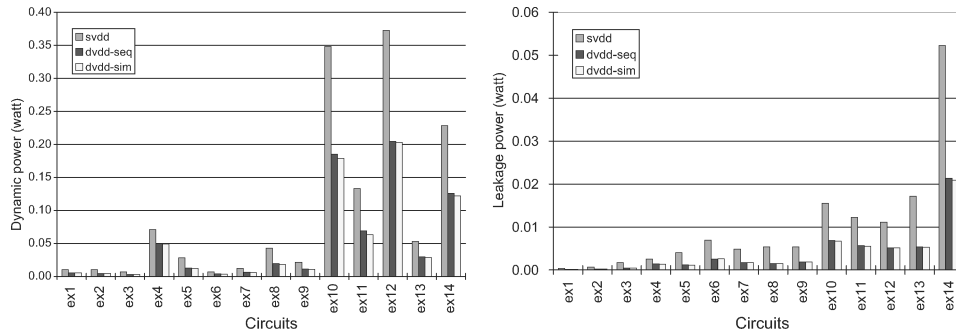


Fig. 11. Comparison of single-Vdd (svdd), sequential dual-Vdd budgeting (dvdd-seq), and simultaneous dual-Vdd budgeting and retiming (dvdd-sim) for: (a) interconnect dynamic power; (b) interconnect leakage power.

approach achieves 8.1% (up to 28.8%) and 3.8% (up to 8.2%) interconnect power reduction on average for MCNC and industrial circuits, respectively, and the clock period is preserved.<sup>7</sup> Compared to the MILP-based simultaneous approach, the LP-based approach relaxes the timing target and generally reduces more<sup>8</sup> interconnect power with acceptable performance degradation. In our implementation, we discard these LP-relaxed solutions (“-” in the table) if they increase clock period by more than 1%. Note that the delay overhead due to the LP-based approach is less than 0.1% for 75% (18 out of 24) circuits. Figure 11 breaks down the total interconnect power into dynamic and leakage contributors and shows the effectiveness of the algorithm for interconnect dynamic- and leakage power reduction, respectively. The logic (including LUTs and FFs) power is not considered here, since the focus of this work is interconnect power reduction and the interconnect power is the dominant component of overall power dissipation [Lin et al. 2005a]. In the future, we will consider the extra FF power dissipation (due to retiming) as a constraint in our formulation.

Table V compares the runtime of MILP- and LP-based simultaneous approaches. The LP-based approach is more than 7x faster than the MILP-based one on average. Note that the LP relaxation and local legalization overcome the essential intractability of the MILP-based approach. More speedup and memory efficiency are expected for larger circuits. As shown in Table V, our MILP-based algorithm finishes the biggest design in MCNC benchmark, clma, within 1.5 hours. Note that the runtime for placing and routing clma (by VPR) is around 1 hour in our test machine, and is comparable to the runtime consumed by our physical synthesis algorithm. Despite the NP-hardness of the generic MILP problem, our MILP formulation contains a big portion of unimodule constraints, such as (27) and (28), which accelerate the solution time of the MILP

<sup>7</sup>For circuit “dsip”, the simultaneous approach increases interconnect power due to the overestimation of the upper bound estimation of the power reduction. The superfluous timing slack will not lead to more VddL switches.

<sup>8</sup>For certain circuits such as “tseng” and “ex11”, the MILP-based approach reduces more power than LP-based approach, as the legalization step (Algorithm 1) after the timing budgeting will degrade the optimal solution.

Table V. Runtime(s) Comparison for MILP-Based and LP-Based Simultaneous Retiming and Budgeting

MCNC			Industrial		
circuit	MILP-based	LP-based	circuit	MILP-based	LP-based
clma	4491	1024	ex5	9	5
tseng	7	2	ex6	136	4
dsip	2	2	ex7	18	4
diffeq	38	3	ex8	7	6
s298	55	8	ex9	59	6
bigkey	30	20	ex10	28	6
elliptic	51	7	ex11	97	23
frisc	1163	33	ex12	205	8
s38584.1	325	18	ex13	8	8
s38417	221	25	ex14	57	50
ave	(10x)		ave	(7x)	

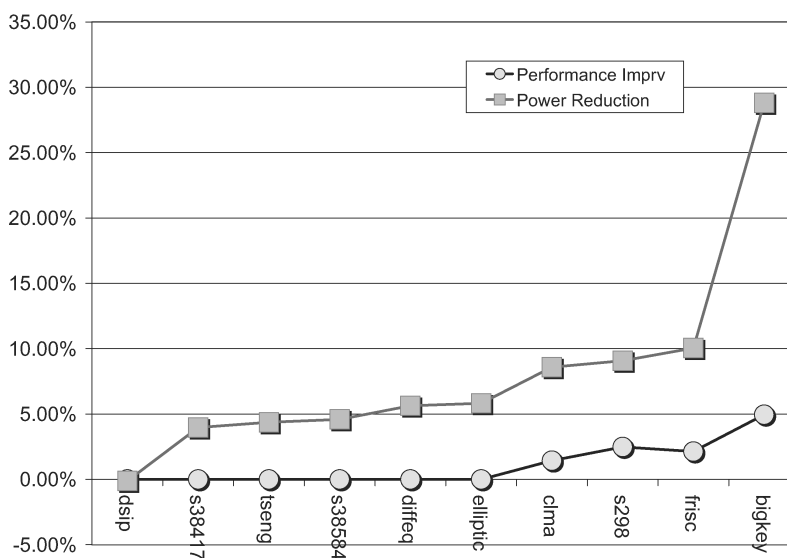


Fig. 12. Indicator for retiming gain.

solver in practice. Nevertheless, we believe that the physical synthesis phase for large designs (e.g., with 330,000 logic cells in the largest available FPGAs) has to be applied to the partial circuit instead of the full chip. A hierarchical or multilevel optimization framework can be adopted for the seek of the efficiency. Alternatively, we can select a critical subcircuit and perform our physical synthesis algorithms.

#### 4.7 Indicator of Power Reduction and Its Application

For large circuits such as clma and ex14, the computational cost is expensive even after LP relaxation. Fortunately, we find an effective indicator of the gain (further power reduction) using the simultaneous approach. Figure 12 compares the power reduction (percentage) obtained by the simultaneous approach and the performance improvement (percentage) by postlayout min-clock

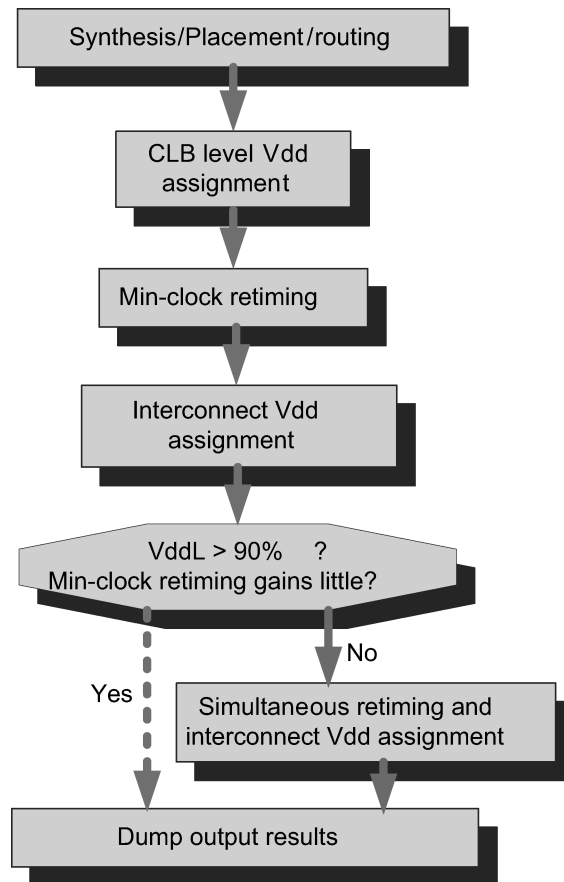


Fig. 13. Runtime-efficient postlayout resynthesis CAD flow.

retiming. An interesting observation is that they are closely correlated to each other; in other words, the larger performance improvement obtained by postlayout min-clock retiming indicates that greater power reduction can be achieved by simultaneous retiming and budgeting. In general, the topology (logic structure and packing result) of a circuit impacts the effectiveness of retiming significantly, and is an efficient way to implicitly measure the suitability of a topology for retiming by performing the min-clock retiming.<sup>9</sup> Figure 13 shows the overall CAD flow for power-aware postlayout re-synthesis processes. The simultaneous approach is performed only when necessary.

## 5. CONCLUSION AND FUTURE WORK

To reduce interconnect power in dual-Vdd FPGAs with mixed-length wire segments, we have presented a new method to give a tight upper bound estimation for power reduction using dual-Vdd, given the budgeted timing slack, and

<sup>9</sup>The min-clock retiming [Lin and Zhou 2004] adopts the combinational optimization algorithm in the graph and is extremely fast (0.1 second for ex14).

formulated the dual-Vdd timing slack allocation problem with linear programming (LP) and min-cost network flow, respectively. Tested on both MCNC and industrial circuits, the network-flow-based approach is more efficient and both approaches (LP- and network-flow-based) reduce interconnect power by 50% on average compared to single-Vdd designs. To explore a larger solution space for more power reduction, an MILP-based simultaneous retiming and timing slack budgeting (SRSB) formulation has been proposed and sped-up by LP relaxation and FF local legalization. Compared to the sequential approach, the MILP-based SRSB algorithm reduces 8.1% (up to 28.8%) and 3.8% (up to 7.0%) interconnect power, on average, for MCNC and industrial circuits, respectively. The LP-relaxation-based speedup technologies achieve 7x (up to 30x) average speedup with similar power-delay product. To further reduce runtime, a reliable indicator for the potential power reduction by SRSB is proposed and an efficient postlayout power-aware resynthesis CAD flow is presented to apply SRSB only when necessary. In the future, we will consider the extra FF power dissipation (due to retiming) in our problem formulation.

#### REFERENCES

- ALTERA. 2006. The Stratix III devices. <http://www.altera.com/literature/lit-stx3.jsp>.
- ANAN, U. N., EICHEN PAN, P., AND LIU, C. L. 1998. Low power logic synthesis under a general delay model. In *Proceedings of the International Symposium on Low-Power Electronics and Design (ISLPED)*.
- ANDERSON, J. H. AND NAJM, F. N. 2004. Low-Power programmable routing circuitry for FPGAs. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.
- BETZ, V., ROSE, J., AND MARQUARDT, A. 1999. *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic.
- BPTM. 2002. Berkeley predictive technology model. <http://www.device.eecs.berkeley.edu/ptm/mosfet.html>.
- CHABINI, N. AND CHABINI, I. 2003. Unification of basic retiming and supply voltage scaling to minimize dynamic power consumption for synchronous digital designs. In *Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI)*.
- CHEN, D. AND CONG, J. 2004. Daomap: A depth-optimal area optimization mapping algorithm for FPGA designs. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.
- CONG, J., LI, H., AND WU, C. 1999. Simultaneous circuit partitioning/clustering with retiming for performance optimization. In *Proceedings of the IEEE ACM Design Automation Conference (DAC)*.
- CONG, J. AND LIM, S. K. 2000. Physical planning with retiming. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.
- CONG, J. AND WU, C. 1996. An improve algorithm for performance optimal technology mapping with retiming in LUT-based FPGA design. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.
- CONG, J. AND WU, C. 1997. Fpga synthesis with retiming and pipelining for clock period minimization of sequential circuits. In *Proceedings of the IEEE ACM Design Automation Conference (DAC)*.
- CONG, J. AND WU, C. 1998a. An efficient algorithm for performance-optimal FPGA technology mapping with retiming. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* 17, 9, 489–498.
- CONG, J. AND WU, C. 1998b. Optimal FPGA mapping and retiming with effiecient initial state computation. In *Proceedings of the IEEE ACM Design Automation Conference (DAC)*.
- CONG, J. AND YUAN, X. 2003. Multilevel global placement with retiming. In *Proceedings of the IEEE ACM Design Automation Conference (DAC)*.

- FISCHER, R., BUCHENRIEDER, K., AND NAGELDINGER, U. 2005. Reducing the power consumption of fpgas through retiming. In *Proceedings of the 12th IEEE Conference and Workshops on the Engineering of Computer-Based Systems (ECBS)*.
- GAYASEN, A., LEE, K., VIJAYKRISHNAN, N., KANDEMIR, M., IRWIN, M. J., AND TUAN, T. 2004. A dual-VDD low power FPGA architecture. In *Proceedings of the International Conference on Field Programmable Logic and Applications (FPL)*.
- GHIASI, S., BOZORGZADEH, E., CHOUDHURI, S., AND SARRAFZADEH, M. 2004. A unified theory of timing budget management. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.
- GOLDBERG, A. V. 1997. An efficient implementation of a scaling minimum-cost flow algorithm. *J. Algor.* 22.
- HAMADA, M. 1998. A top-down low power design technique using clustered voltage scaling with variable supply-voltage scheme. In *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*.
- HSU, Y. L. AND WANG, S. J. 2002. Retiming-Based logic synthesis for low-power. In *Proceedings of the International Symposium on Low-Power Electronics and Design (ISLPED)*.
- LEISERSON, C. L. AND SAXE, J. B. 1991. Retiming synchronous circuitry. *Algorithmica*, 5–35.
- LEWIS, D., AHMED, E., BAECKLER, G., BETZ, V., BOURGEOULT, M., CASHMAN, D., GALLOWAY, D., HUTTON, M., LANE, C., LEE, A., LEVENTIS, P., MARQUARDT, S., MCCLINTOCK, C., PADALIA, K., PEDERSEN, B., POWELL, G., RATCHEV, B., REDDY, S., SCHLEICHER, J., STEVENS, K., YUAN, R., CLIFF, R., AND ROSE, J. 2005. The Stratix II logic and routing architecture. In *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (ISFPGA)*.
- LEWIS, D., BETZ, V., JEFFERSON, D., LEE, A., LANE, C., LEVENTIS, P., MARQUARDT, S., MCCLINTOCK, C., PEDERSEN, B., POWELL, G., REDDY, S., WYSOCKI, C., CLIFF, R., AND ROSE, J. 2003. The stratix routing and logic architecture. In *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (ISFPGA)*.
- LI, F., LIN, Y., AND HE, L. 2004a. FPGA power reduction using configurable dual-VDD. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC)*.
- LI, F., LIN, Y., AND HE, L. 2004b. VDD programmability to reduce FPGA interconnect power. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.
- LI, F., LIN, Y., HE, L., AND CONG, J. 2004c. Low-power FPGA using pre-defined dual-VDD/dual-VT fabrics. In *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*.
- LIN, C. AND ZHOU, H. 2004. Optimal wire retiming without binary search. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.
- LIN, Y. AND HE, L. 2006. Dual-VDD interconnect with chip-level time slack allocation for FPGA power reduction. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.*
- LIN, Y., LI, F., AND HE, L. 2005a. Circuits and architectures for field programmable gate array with configurable supply voltage. *IEEE Trans. VLSI* 13, 9, 1035–1047.
- LIN, Y., LI, F., AND HE, L. 2005b. Power modeling and architecture evaluation for FPGA with novel circuits for vdd programmability. In *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*.
- MONTERIRO, J., DEVADAS, S., AND GHOSH, A. 1993. Retiming sequential circuits for low power. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.
- MOSEK. 2006. *MOSEK Optimization Toolbox*. <http://www.mosek.com>.
- PAN, P., KARANDIKAR, A. K., AND LIU, C. L. 1998. Optimal clock period clustering for sequential circuits with retiming. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.*
- RIVEST, R., CORMEN, T., AND LEISERSON, C. 1990. *An Introduction to Algorithms*. MIT Press.
- TIEN, T. C., SU, H. P., AND TSAY, Y. W. 1998. Integrating logic retiming and regist placement. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.
- WIRTH, G. I., VIEIRA, M. G., HENES NETO, E., AND KASTENSMIDT, F. G. L. 2005. Single event transients in combinational circuits. In *Proceedings of the 18th Annual Symposium on Integrated Circuits and System Design (SBCCI)*.
- XILINX. 2005. Xilinx product datasheets. <http://www.xilinx.com/literature>.
- XILINX. 2006. Xilinx ISE software manuals and help. [http://www.xilinx.com/support/sw\\_manuals/xilinx9/index.htm](http://www.xilinx.com/support/sw_manuals/xilinx9/index.htm).

- YANG, S. 1991. *Logic Synthesis and Optimization Benchmarks, Version 3.0*. Microelectronics Center of North Carolina (MCNC).
- YEH, C.-Y. AND MAREK-SADOWSKA, M. 2003a. Delay budgeting in sequential circuit with application on FPGA placement. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC)*.
- YEH, C.-Y. AND MAREK-SADOWSKA, M. 2003b. Minimum-area sequential budgeting for FPGA. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*.

Received October 2006; revised October 2007; accepted October 2007