

# Unification of Basic Retiming and Supply Voltage Scaling to Minimize Dynamic Power Consumption for Synchronous Digital Designs

Noureddine Chabini

Computer Engineering, Dep. of Electrical Engineering,  
Princeton University, Princeton, NJ, USA, 08544.  
nchabini@princeton.edu

El Mostapha Aboulhamid

DIRO, Université de Montréal, C.P. 6128, Succursale  
Centre-ville, Montréal, Qc, Canada, H3C 3J7.  
aboulham@iro.umontreal.ca

Ismaïl Chabini

Massachusetts Institute of Technology, Room 1-263,  
Cambridge, MA, USA, 02139-4307.  
chabini@mit.edu

Yvon Savaria

Dep. of Elec. Eng., École Polytechnique de Montréal, C.P.  
6079, Suc. Centre-ville, Montréal, Qc, Canada, H3C 3A7.  
savaria@vlsi.polymtl.ca

## ABSTRACT

We address the problem of minimizing dynamic power consumption for single-phase synchronous digital designs, under timing constraints, using an unification of basic retiming and supply voltage scaling. We assume that the number of supply voltages and their values are known for each computation element. Our main objective is then to change the location of registers using basic retiming while maximizing the number of computation elements off critical paths that can operate under a low available supply voltage, and can lead to a maximum dynamic power saving. We address the problem at the system-level. We formulate the problem as a Mixed Integer Linear Program (*MILP*). The exact optimal solution for the problem is then guaranteed. We also devise an algorithm to compute bounds on the values assigned by basic retiming to each computational element. Besides helping to find the optimal solution to the problem, these bounds also allow to reduce the run-time for finding this solution. The proposed approach can produce converter-free designs and can also minimize short-circuit power consumption. Experimental results have shown that dynamic power consumption can be reduced by factors that range from 2.78% to 37.24% for single-phase designs with minimal clock period. For these experimental results, the run-time for solving the *MILP* is under 2min.

## Categories and Subjects Descriptors

B. Hardware, B.0 General.

## General Terms

Design, Performance.

**Keywords:** CMOS, Power Consumption, Performance, Retiming, Supply Voltage Scaling, Digital Design.

## 1. INTRODUCTION

The average power consumption in CMOS circuits is constituted by dynamic power, short-circuit power, leakage power,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GLSVLSI'03, April 28-29, 2003, Washington, DC, USA.

Copyright 2003 ACM 1-58113-677-3/03/0004...\$5.00.

and static power [8]. Dynamic power, denoted  $P_{dyn}$ , is the dominant component of power dissipation in CMOS circuits. It is a quadratic function of the supply voltage, denoted  $V_{dd}$  [8]:

$$P_{dyn} = KCfV_{dd}^2, \quad (1)$$

where  $K$  is the switching activity factor,  $C$  is the loading capacitance, and  $f$  is the clock frequency.

The short-circuit power is approximately proportional to  $(V_{dd} - 2V_{th})^3$  [8], where  $V_{th}$  is the threshold voltage.

Due to the quadratic term in Equation (1), the dynamic power may be significantly reduced by scaling down the supply voltage. When  $V_{th}$  is assumed to be fixed, short-circuit power is reduced ever faster, but it is usually small compared to the dynamic power. In the rest of this paper, we focus on dynamic power only.

Minimizing dynamic power consumption by scaling down the supply voltage of computation elements off critical paths has been addressed in the case of combinational designs. For this kind of designs, interested reader is referred to [1][3][8][9][10][13][14] for a literature review. Designs with multiple supply voltages are reported in [11][12].

In this paper, we address the problem of minimizing dynamic power consumption, under timing constraints, for synchronous sequential designs, by scaling down the supply voltage of computation elements off critical paths. Since critical paths are related to the position of registers in the designs, our aim is not just to scale down the supply voltage of computation elements off critical paths, but also to simultaneously move registers from their positions in order to maximize the number of computation elements off critical paths that can lead to a minimum dynamic power consumption. The process of moving registers from their positions in a sequential design is called basic retiming [4]. In the sequel, we refer to that problem as the *MDP* problem.

The *MDP* problem is NP-hard in general, since it is already NP-hard in the case of combinational designs [3].

Minimizing dynamic power for synchronous sequential designs is addressed. In [7], heuristics to minimize the switching activity are presented. The approach in [7] is based on the fact that registers have to be positioned on the output of computational elements of high switching activity, since the output of a register switches only at the arrival of the clock signal compared to a computational element that may switch many times during the clock period.

In [6], fixed-phase retiming is proposed. The edge-triggered circuit is first transformed to a two-phase level-clocked circuit, by replacing each edge-triggered flip-flop by two latches. Next, the latches of one phase are kept fixed, while the latches belonging to the other phase are moved onto wires with high switching activity and loading capacitance.

To the best of our knowledge, no approach is reported yet for addressing the *MDP* problem. In this paper, we focus on solving the problem at the system level. The computational elements in the design can be, for instance, adders and multipliers. We assume that the number of supply voltages and their values are known for each computation element. We formulate the problem as a Mixed Integer Linear Programming (*MILP*) problem. This formulation is flexible. For instance, we can incorporate constraints into the set of constraints of this *MILP* in order to produce converter-free designs. We show that there is an infinite number of basic retimings that lead to the same minimum value of the objective function in the *MILP*. To find one of these basic retimings, one needs to find bounds on the values assigned by basic retiming to each computational element, but without affecting the minimal value of the objective function of the *MILP*. We devise an algorithm to find this kind of bounds, which allow to also reduce the run-time to solve this *MILP*. Experimental results have shown a significant reduction of dynamic power consumption using this proposed approach. Also, they have shown that this *MILP* can be solved with a short run-time.

## 2. PRELIMINARIES

### 2.1- Cyclic Graph Model

A synchronous sequential design is modeled (as in [4]) as a directed cyclic graph  $G = (V, E, d, w)$ , where  $V$  is the set of computational elements in the design, and  $E$  is the set of edges that represent interconnections between vertices. Each vertex  $v$  in  $V$  has a non-negative integer execution delay  $d(v) \in N$ . Each edge  $e_{u,v}$ , from node  $u$  to node  $v$ , in  $E$  is weighted with a register count  $w(e_{u,v}) \in N$ , representing the number of registers on the wire between  $u$  and  $v$ .

### 2.2- Basic Retiming

Let  $G = (V, E, d, w)$  be a synchronous sequential design. Basic retiming (or retiming for short in the sequel)  $r$  [4] is defined as a function  $r : V \rightarrow Z$ , which transforms  $G$  to a functionally equivalent synchronous sequential design  $G_r = (V, E, d, w_r)$ . The weight of each edge  $e_{u,v}$  in  $G_r$  is defined as follows:

$$w_r(e_{u,v}) = w(e_{u,v}) + r(v) - r(u), \forall e_{u,v} \in E. \quad (2)$$

Since the weight of each edge in  $G_r$  represents the number of registers on that edge, then we must have:

$$w_r(e_{u,v}) \geq 0, \forall e_{u,v} \in E. \quad (3)$$

Any retiming  $r$  that satisfies Equation (3) is called a valid retiming.

Equation (2) implies that for every two nodes  $u$  and  $v$  in  $V$ , the change in the register count along any path  $P(u, v)$  from node  $u$  to node  $v$  depends only on its two endpoints:

$$w_r(P(u, v)) = w(P(u, v)) + r(v) - r(u), \forall u, v \in V, \quad (4)$$

where:

$$w(P(u, v)) = \sum_{e_{x,y} \in P(u, v)} w(e_{x,y}). \quad (5)$$

Let  $\zeta$  be the set of cycles in  $G$ . Since a cycle is a path that start from and ends to the same node in the graph, then from Equation (4), we have that:

$$w_r(C) = w(C), \forall C \in \zeta. \quad (6)$$

**Property 1 :** Any valid retiming  $r$  does not change the total number of registers in any cycle.  $\square$

One application of retiming is to minimize the clock period of synchronous sequential designs.

A mathematical programming framework for retiming has been proposed in [4]. For the purpose of this paper, we extract from that framework the following three theorems (Theorems 1, 2, and 3), which are also proved in [4].

**Theorem 1 :** Let  $G = (V, E, d, w)$  be a synchronous circuit, and let  $P$  be a positive real number. Then the clock period of  $G$  is less than or equal to  $P$  if and only if there exists a function  $s : V \rightarrow [0, P]$  such that  $-s(v) \leq -d(v), \forall v \in V$  and such that  $s(u) - s(v) \leq -d(v), \forall e_{u,v} \in E$  and  $w(e_{u,v}) = 0$ .  $\square$

**Theorem 2 :** Let  $G = (V, E, d, w)$  be a synchronous circuit, and let  $P$  be a positive real number. Then there is a retiming  $r$  of  $G$  such that the clock period of the resulting retimed graph is less than or equal to  $P$  if and only if there exists an assignment of real value  $s(v)$  and integer value  $r(v)$  to each node  $v$  in  $V$  such that the following conditions are satisfied: (1)  $-s(v) \leq -d(v), \forall v \in V$ , (2)  $s(v) \leq P, \forall v \in V$ , (3)  $r(u) - r(v) \leq w(e_{u,v}), \forall e_{u,v} \in E$ , and (4)  $s(u) - s(v) \leq -d(v), \forall e_{u,v} \in E$ , and  $r(u) - r(v) = w(e_{u,v})$ .  $\square$

**Theorem 3 :** Let  $G = (V, E, d, w)$  be a synchronous circuit, and let  $P$  be a positive real number. Then there is a retiming  $r$  of  $G$  such that the clock period of the resulting retimed graph is less than or equal to  $P$  if and only if there exists an assignment of real value  $R(v)$  and integer value  $r(v)$  to each node  $v$  in  $V$  such that the following conditions are satisfied: (1)  $r(v) - R(v) \leq -d(v)/P, \forall v \in V$ , (2)  $R(v) - r(v) \leq 1, \forall v \in V$ , (3)  $r(u) - r(v) \leq w(e_{u,v}), \forall e_{u,v} \in E$ , and (4)  $R(u) - R(v) \leq w(e_{u,v}) - d(v)/P, \forall e_{u,v} \in E$ .  $\square$

In the following, we give other results related to retiming, which are not developed elsewhere. They will be used in Section 3.

**Lemma 1 :** Any retiming  $r : V \rightarrow Z$  can be transformed to a non-negative retiming  $r^+ : V \rightarrow N$ .  $\square$

**Proof:** Let  $m = \text{Min}\{r(v), \forall v \in V\}$ . The function  $r^+$  defined as  $r^+(v) = r(v) - m, \forall v \in V$ , is a non-negative retiming.  $\square$

**Theorem 4 :** Let  $\zeta_v$  be the set of the cycles in  $G$  that pass through the node  $v \in V$ . For any valid retiming  $r$ ,  $U_v = \text{Min}\{w(C), \forall C \in \zeta_v\}$  is a bound on  $r(v), \forall v \in V$ .  $\square$

Proof of Theorem 4 is omitted due to space limit. It is in [2].

**Remark:** If  $G$  is not a strongly connected directed graph, then Theorem 4 can be applied by adding a dummy node as done in [4].

## 3. PROBLEM FORMULATION, AND OPTIMAL SOLUTION

We are given a synchronous sequential design  $G = (V, E, d, w)$  that operates with a given supply voltage, which is called here highest supply voltage. The design is assumed to operate at a target clock period,  $P$ , that is given by the designer or determined by applying a retiming for clock period minimization on  $G$ . We are also given multiple supply voltages. The number of supply voltages and their values are assumed to be known and can be not the same for all the computational elements. Our objective is then to assign new supply voltages, from the set of the given supply voltages, to the computational elements of  $G$  in order to minimize the total dynamic power consumption. Each computational element will have one and only one supply voltage. Since we want that  $G$  keeps operating at the target clock period  $P$ , the computational element on critical paths will be kept operating at their original supply voltages, while the supply voltages of those off critical paths are replaced by low supply voltages.

Minimizing dynamic power consumption in synchronous sequential designs by only changing the supply voltages of computational element off critical paths is not enough. Retiming is required to shift computational elements from the critical paths.

As a summary, our main problem in this paper is then to provide a manner to simultaneously apply retiming and voltage scaling, to have a functionally equivalent design that operates at the same given clock period and consumes the minimum dynamic power. We refer to this problem as the *MDP* problem.

We formulate the *MDP* problem as an *MILP*. The optimal solution to the problem is then obtained by solving this *MILP*.

Before presenting this *MILP*, let us first provide additional notation and definitions. Based on the supply voltages used, we assume that we have  $n_v$  different implementations for each computational element  $v$ . If supply voltage  $V_{dd}^k$ , where  $1 \leq k \leq n_v$ , is used, then the computational element  $v$  has an execution delay  $d(v) = d_{v,k}$  and consumes the dynamic power  $p_{v,k}$ . Assume that supply voltages are sorted from the highest to the smallest. For each  $v \in V$ , and for each  $k$  such that  $1 \leq k \leq n_v$ , let us denote by  $x_{v,k}$  a binary variable, which is equal to 1 if supply voltage  $k$  is used, and to 0 otherwise.

The objective function to minimize in the *MILP* is the total dynamic power consumption, which is equivalent to (7):

$$\text{Minimize} \left( \sum_{v \in V} \sum_{k=1}^{n_v} p_{v,k} \cdot x_{v,k} \right). \quad (7)$$

By definition of the inputs of the problem, if one uses the highest supply voltages only, then by Theorem 3, the following system of inequalities,  $S_1$ :

$$r(v) - R(v) \leq -d(v)/P, \forall v \in V \quad (8)$$

$$R(v) - r(v) \leq 1, \forall v \in V \quad (9)$$

$$r(u) - r(v) \leq w(e_{u,v}), \forall e_{u,v} \in E \quad (10)$$

$$R(u) - R(v) \leq w(e_{u,v}) - d(v)/P, \forall e_{u,v} \in E \quad (11)$$

has a solution.

As proved in [4], any solution to the system of inequalities in Theorem 2 can be transformed to a solution to the system of inequalities in Theorem 3. This is possible by making the following transformation:  $P \cdot R(v) = P \cdot r(v) + s(v), \forall v \in V$ . (12)

By making the following transformation:

$$g(v) = s(v) - d(v), \forall v \in V, \quad (13)$$

$S_1$  can be transformed to the equivalent system of inequalities,  $S_2$ :

$$r(v) - R(v) \leq 0, \forall v \in V \quad (14)$$

$$R(v) - r(v) \leq 1 - d(v)/P, \forall v \in V \quad (15)$$

$$r(u) - r(v) \leq w(e_{u,v}), \forall e_{u,v} \in E \quad (16)$$

$$R(u) - R(v) \leq w(e_{u,v}) - d(u)/P, \forall e_{u,v} \in E \quad (17)$$

The system of inequalities  $S_2$  expresses a valid retiming of the design  $G$  operating at the highest supply voltages and the target clock period  $P$ . If instead of keeping the execution delays (i.e.,  $d(v)$ 's) constant one makes them variables, it is then possible to have a system,  $S_3$ , of equalities and inequalities that expresses the combination of retiming and supply voltage scaling to minimize dynamic power consumption. Indeed, one can build that system as follows:

$$\sum_{k=1}^{n_i} x_{v,k} = 1, \forall v \in V \quad (18)$$

$$r(v) - R(v) \leq 0, \forall v \in V \quad (19)$$

$$R(v) - r(v) \leq 1 - \left( \sum_{k=1}^{n_v} d_{v,k} \cdot x_{v,k} \right) / P, \forall v \in V \quad (20)$$

$$r(u) - r(v) \leq w(e_{u,v}), \forall e_{u,v} \in E \quad (21)$$

$$R(u) - R(v) \leq w(e_{u,v}) - \left( \sum_{k=1}^{n_u} d_{u,k} \cdot x_{u,k} \right) / P, \forall e_{u,v} \in E \quad (22)$$

$$x_{v,k} \in \{0, 1\} \text{ and } r(v) \in \mathbb{Z}, \forall v \in V \text{ and } k = 1, \dots, n_v \quad (23)$$

The system  $S_3$  can then be used as set of constraints of the *MILP*.

**Theorem 5 :** a) The *MILP* formed by (7) and  $S_3$  has always a solution. b) The design  $G$  obtained from the optimal solution to this *MILP* has the clock period  $P$ . c) This *MILP* has an infinite number of optimal solutions (because there is an infinite number of retiming).  $\square$

Proof of Theorem 5 is omitted due to space limit. It is in [2].

To find one of the optimal solutions of the *MILP* constituted by (7) and  $S_3$ , we need to determine a lower and an upper bound on each  $r(v)$ . Tight bounds on  $r(v)$  may also help to prune the solution space, and hence to speed up the process of finding this optimal solution.

From Theorem 4, we have that:

$$|r(v)| \leq U_v, \forall v \in V, \quad (24)$$

which is equivalent to:

$$r(v) \leq U_v, \forall v \in V, \quad (25)$$

and

$$-r(v) \leq U_v, \forall v \in V. \quad (26)$$

$U_v$ 's values can be determined by the algorithm in Figure 1.

From Lemma 1,  $r(v)$  can be made non-negative. Let

$m = \text{Min}\{-U_v, \forall v \in V\}$ , and  $r^+(v) = r(v) - m, \forall v \in V$ .

Equations (25) and (26) can then be transformed to:

$$r^+(v) \leq U_v - m, \forall v \in V, \quad (27)$$

and

$$-r^+(v) \leq U_v + m, \forall v \in V. \quad (28)$$

From Equations (21), (27) and (28), one can deduce that:

$$r^+(u) - r^+(v) \leq \text{Min}(w(e_{u,v}), (U_u + U_v)), \forall e_{u,v} \in E. \quad (29)$$

**Input:**  $G = (V, E, d, w)$ . **Output:** bounds on  $r(v)$ 's,  $\forall v \in V$ .

**Begin**

- 1- Weight each edge  $e_{u,v}$  in  $G$  by  $w(e_{u,v})$ ;
- 2- Split each  $v \in V$  to two nodes:  $\tilde{v}$  and  $\hat{v}$ .
- 3- Connect all the input edges of  $v$  to  $\tilde{v}$ .
- 4- Connect all the output edges of  $v$  to  $\hat{v}$ .
- 5- Add an edge of weight 0 between  $\tilde{v}$  and  $\hat{v}$ .
- 6- Compute the length of the shortest path from  $\tilde{v}$  to  $\hat{v}$ .
- 7-  $U_v$  is the length of the shortest path from  $\tilde{v}$  to  $\hat{v}$ .
- 8- Return  $U_v$ 's.

**End**

**Figure 1 : Determining bounds on  $r(v)$ 's.**

Let  $r^+(v) = R(v) - m, \forall v \in V$ . Using the developments above, we can then obtain the final form of the *MILP* to optimally solve the *MDP* problem as presented in Figure 2.

$$\text{Minimize} \left( \sum_{v \in V} \sum_{k=1}^{n_v} p_{v,k} \cdot x_{v,k} \right) \quad (30)$$

Subject to:

$$\sum_{k=1}^{n_i} x_{v,k} = 1, \forall v \in V \quad (31)$$

$$r^+(v) - R^+(v) \leq 0, \forall v \in V \quad (32)$$

$$R^+(v) - r^+(v) \leq 1 - \left( \sum_{k=1}^{n_v} d_{v,k} \cdot x_{v,k} \right) / P, \forall v \in V \quad (33)$$

$$r^+(u) - r^+(v) \leq \text{Min}(w(e_{u,v}), (U_u + U_v)), \forall e_{u,v} \in E \quad (34)$$

$$R^+(u) - R^+(v) \leq w(e_{u,v}) - \left( \sum_{k=1}^{n_u} d_{u,k} \cdot x_{u,k} \right) / P, \forall e_{u,v} \in E \quad (35)$$

$$r^+(v) \leq U_v - m, \forall v \in V \quad (36)$$

$$-r^+(v) \leq U_v + m, \forall v \in V \quad (37)$$

$$x_{v,k} \in \{0, 1\} \text{ and } r^+(v) \in \mathbb{N}, \forall v \in V \text{ and } k = 1, \dots, n_v \quad (38)$$

**Figure 2 : Final form of the proposed *MILP*.**

## 4. EXPERIMENTAL RESULTS

We assess the effectiveness and efficiency, in terms of reducing dynamic power consumption and the run-time, of Figure 2. We use a set of benchmarks proposed in the literature. As supply voltages, we use the first  $x$ -supply voltages from the set  $\{5V, 4.5V, 4V, 3.5V, 3V, 2.5V, 2V, 1.5V\}$ , where  $x = 1, 2, \dots, 8$ . The difference between two successive supply voltages is fixed to 0.5V, since we also want to test the effectiveness of the proposed approach in producing converter-free designs based on [10]. The approach in [10] assumes that a level-converter between computational element  $u$  and  $v$  can be omitted if (39) is satisfied:

$$\forall e_{u,v} \in E, \sum_{k=1}^{n_v} x_{v,k} V_{dd}^k - \sum_{k=1}^{n_u} x_{u,k} V_{dd}^k \leq V_{st}, \quad (39)$$

where  $V_{st}$  is a given value, which is fixed here to 0.5V. We assume that supply voltages are greater than  $2 \cdot V_{th}$ , where  $V_{th}$  is the threshold voltage. We use  $V_{th} = 0.7V$ , which is a typical value used in the literature. To determine the delay  $d_{i,k}$  and the power consumed  $p_{i,k}$ , for a given computational element  $v$  and supply voltage  $V_{dd}^k$ , we proceed as follows. First, we use the expression

$$d_{v,k} = (V_{dd}^k / (V_{dd}^k - V_{th})^2) \cdot ((V_{dd}^1 - V_{th})^2 / V_{dd}^1) \cdot d_{v,1}$$

described in [9], where  $V_{dd}^1$  is assumed to be known. Second,  $p_{v,k}$ 's are determined assuming that the fanout of  $v$  has the same loading capacitance. From Equation (1), we then have:

$$p_{v,k} = KC_{0f} \cdot (Fanout_v \cdot (V_{dd}^k)^2) \cdot$$

For each circuit, the clock period  $P$  is fixed to the minimal value determined by applying a retiming on the circuit operating at the highest supply voltages (i.e., 5V).

All experiments were done using an UltraSparc 10 with 1GB RAM. We use [5] to solve the MILPs. For each circuit, the MILP is automatically generated by a module we coded in C++.

Table 1 summarizes numerical results. The first column of the table gives the name of the circuits used. The second column presents the dynamic power consumed, denoted  $P_h$ , divided by  $(KC_{0f})$ , if one uses only the highest supply voltages. Results in the  $x$ th column, where  $x = 3, 4, \dots, 9$  correspond to the use of  $(x-1)$ -supply voltages if one uses Figure 2 (results in bold), or Figure 2 with (39) incorporated in its set of constraints (results between  $[$  and  $]$ ). The  $x$ th column gives the Relative Saving ratio (RS), and the run-time in seconds to solve the MILP. We have that:

$$RS = ((P_h - P_m) / P_h) \times 100,$$

where  $P_m$  denotes the dynamic power consumed if one used the first  $(x-1)$ -supply voltages as defined above.

As Table 1 exhibits, significant reductions of dynamic power consumption are obtained using from 2 to 6 supply voltages. The RS values range from 2.78% to 37.24%, and from 2.78% to 21.94% when Inequality (39) is incorporated in the MILP. As Table 1 reports, the use of more than 6 supply voltages does not lead to any dynamic power reduction. The run-time to solve the MILP is very short, and is less than 2min when 2 to 6 supply voltages are used. This short run-time is due in part to the bounds found by Figure 1.

## 5. CONCLUSIONS

We addressed the problem of minimizing the dynamic power consumption for synchronous sequential designs operating with a given clock period. The value of the clock period is preserved while the supply voltage of some computational elements off critical paths are re-assigned to low supply voltages in order to minimize the dynamic power consumption. Maximizing the number of computational elements that can re-operate under a low supply voltage (to have the minimal dynamic power consumption) is done by combining retiming and supply voltage scaling.

To the best of our knowledge, this is the first time that combining retiming and supply voltage scaling to minimize dynamic power consumption is proposed as a problem to solve. We formulated the problem as an MILP, which allows to determine the exact optimal solution to the problem. We also devised an algorithm to find bounds on the lags determined by retiming; this allows to find an optimal solution to the MILP and reduce run-times to find it.

The proposed approach can significantly reduce the dynamic power consumption. The run-times to solve the proposed MILP are very short; this is due in part to the bounds found by Figure 1.

The proposed MILP can be used at the system level, and could be used heuristically at lower levels of abstraction when computational elements are for instance gates, and the design is of a very large size. It is also flexible. For instance, the cost of level-converters as well as the dynamic power due to registers can both be added to the formulation to control them.

## ACKNOWLEDGMENTS

This research benefited from financial support from Le Fonds Nature et Technologies (Quebec, Canada), NSF (USA), NSERC (Canada), and Canada Research Chair to Savaria.

## REFERENCES

- [1] N.Chabini, I.Chabini, E.-M.Aboulhamid, and Y.Savaria, "Methods for Minimizing Dynamic Power Consumption in Synchronous Designs with Multiple Supply Voltages," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, V.22, N.3, 2003, pp. 346-351.
- [2] —, "An Approach for Integrating Retiming and Supply Voltage Scaling to Minimize Power Consumption for Synchronous Digital Designs," Tech. Rep., 2002, DIRO, Univ. of Montreal.
- [3] J.-M.Chang and M.Pedram, "Energy Minimization Using Multiple Supply Voltages," *IEEE Trans. on VLSI*, Vo.5, No.4, 1997, pp.1-8.
- [4] C.E. Leiserson and J.B. Saxe, "Retiming Synchronous Circuitry," *Algorithmica*, Jan., 1991.
- [5] The LP\_Solve Tool: [ftp://ftp.ics.ele.tue.nl/pub/lp\\_solve/](ftp://ftp.ics.ele.tue.nl/pub/lp_solve/)
- [6] K.N. Lalgudi and M. Papaefthymiou, "Fixed-phase retiming for low power," *Proc. of the Int. Symp. on Low Power Electronics and Design*, 1996, pp. 259-264.
- [7] J. Monteiro, S. Devadas, and A. Ghosh, "Retiming sequential circuits for low power," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, 1993, pp. 398-402.
- [8] A. Raghunathan, N. K. Jha, and S. Dey, *High-Level Power Analysis and Optimization*, Kluwer Academ. Pub., MA, 1997.
- [9] K.Usami and M.Horowitz, "Clustered Voltage Scaling Technique for Low-Power Design," *Proc. Int. Workshop on Low Power Design*, 1995, pp. 3-8.
- [10] Y.-J.Yeh, S.-Y.Kuo, and J.-Y.Jou, "Converter-Free Multiple-Voltage Scaling Techniques for Low-Power CMOS Digital Design," *IEEE Trans. on CAD*, V.20, N.1, 2001, pp.172-6.
- [11] I.Mutsaers, and al., "A low-power design method using multiple supply voltages," *Proc. of the Int. Symp. on Low Power Electronics and Design*, August 1997, pp. 36-41.
- [12] T. Pering, T.D. Burd, and R.W. Brodersen, "Voltage Scheduling in the IpARM Microprocessor System," *Int. Symp. on Low Power Electronics and Design*, 2000, pp. 96-101.
- [13] S.Raje and M.Sarrafzadeh, "Scheduling with multiple voltages," *Integration, the VLSI Journal*, Vol. 23, 1997, pp. 37-59.
- [14] K. Roy, Wei Liqiong, Chen Zhanping, "Multiple-V<sub>dd</sub> multiple-V<sub>th</sub> CMOS (MVCMS) for low power applications," *Proceedings of IEEE International Symposium on Circuits and Systems*, Vol.1, 1999, pp. 366-370.

Table 1 : Assessment of the proposed MILP.

Circuit Name	1-V <sub>dd</sub> $P_h^{dyn}$	2-V <sub>dd</sub> RS(%), T(s) [RS,T]	3-V <sub>dd</sub> RS(%), T(s) [RS,T]	4-V <sub>dd</sub> RS(%), T(s) [RS,T]	5-V <sub>dd</sub> RS(%), T(s) [RS,T]	6-V <sub>dd</sub> RS(%), T(s) [RS,T]	7-V <sub>dd</sub> RS(%), T(s) [RS,T]	8-V <sub>dd</sub> RS(%), T(s) [RS,T]
AllPoleLatticeFilter	414	<b>4.83, 1</b> [4.83, 1]	<b>6.03, 1</b> [6.03, 1]	<b>6.03, 1</b> [6.03, 1]	<b>6.03, 1</b> [6.03, 1]	<b>6.03, 2</b> [6.03, 1]	<b>6.03, 3</b> [6.03, 1]	<b>6.03, 3</b> [6.03, 1]
BiquadraticFilter	343	<b>7.48, 1</b> [7.48, 1]	<b>14.97, 1</b> [11.97, 1]	<b>17.66, 1</b> [13.77, 1]	<b>18.86, 1</b> [13.77, 1]	<b>18.86, 1</b> [13.77, 1]	<b>18.86, 2</b> [13.77, 1]	<b>18.86, 2</b> [13.77, 1]
SecondAvenhausFilter	412	<b>7.28, 1</b> [7.28, 1]	<b>13.34, 1</b> [12.13, 1]	<b>16.26, 1</b> [12.86, 1]	<b>17.23, 2</b> [12.86, 1]	<b>17.23, 2</b> [12.86, 1]	<b>17.23, 7</b> [12.86, 1]	<b>17.23, 7</b> [12.86, 1]
FifthOrderWaveDigitalFilter	1281	<b>7.49, 1</b> [7.49, 1]	<b>11.55, 1</b> [10.30, 3]	<b>12.95, 1</b> [10.53, 1]	<b>14.44, 4</b> [10.53, 2]	<b>15.06, 18</b> [10.53, 3]	<b>15.53, 7</b> [10.53, 34]	<b>15.53, 41</b> [10.53, 38]
ThirdAvenhausFilter	566	<b>8.83, 1</b> [8.83, 1]	<b>16.78, 1</b> [15.90, 1]	<b>21.02, 1</b> [18.02, 1]	<b>23.14, 10</b> [18.02, 1]	<b>23.85, 16</b> [18.02, 1]	<b>23.85, 115</b> [18.02, 1]	<b>23.85, 118</b> [18.02, 1]
Differential Equation Solver	588	<b>10.71, 1</b> [10.71, 1]	<b>22.61, 1</b> [12.41, 1]	<b>27.72, 2</b> [12.41, 1]	<b>34.52, 3</b> [12.41, 1]	<b>37.24, 3</b> [12.41, 1]	<b>37.24, 9</b> [12.41, 1]	<b>37.24, 11</b> [12.41, 1]
FourAvenhausFilter	720	<b>9.72, 1</b> [9.72, 1]	<b>18.75, 1</b> [18.05, 1]	<b>23.75, 1</b> [21.38, 1]	<b>26.52, 34</b> [21.94, 2]	<b>27.63, 63</b> [21.94, 2]	<b>27.63, 1037</b> [21.94, 2]	<b>27.63, 1417</b> [21.94, 2]
PolynomialDivider	309	<b>9.7, 1</b> [9.7, 1]	<b>19.41, 1</b> [17.79, 1]	<b>25.24, 1</b> [20.71, 1]	<b>27.83, 3</b> [20.71, 1]	<b>29.12, 2</b> [20.71, 1]	<b>29.12, 10</b> [20.71, 1]	<b>29.12, 11</b> [20.71, 1]
SecondOrderIIR	359	<b>2.78, 1</b> [2.78, 1]	<b>5.57, 1</b> [2.78, 1]	<b>5.57, 1</b> [2.78, 1]	<b>5.57, 1</b> [2.78, 1]	<b>5.57, 1</b> [2.78, 1]	<b>5.57, 1</b> [2.78, 1]	<b>5.57, 1</b> [2.78, 1]
Correlator	283	<b>8.48, 1</b> [8.48, 1]	<b>15.54, 1</b> [13.78, 1]	<b>18.02, 15</b> [13.78, 3]	<b>19.43, 50</b> [13.78, 6]	<b>19.78, 118</b> [13.78, 8]	<b>19.78, 306</b> [13.78, 8]	<b>19.78, 401</b> [13.78, 10]