

Simultaneous Peak and Average Power Optimization in Synchronous Sequential Designs Using Retiming and Multiple Supply Voltages

A. K. Allam and J. Ramanujam

Electrical and Computer Engineering Department, Louisiana State University, USA
(atef, jxr)@ece.lsu.edu

Abstract

In this paper, we present a combination of basic retiming and multiple voltage scheduling (MVS) techniques in order to optimize dynamic peak power as well as average power consumption in synchronous sequential circuits under timing constraints. First, we devise a mixed-integer linear programming (MILP) formulation for the problem of scheduling for optimal peak and/or average power consumption through a unification of retiming and MVS techniques. Then, to alleviate the problem of variable explosion in MILP, we present a two-stage algorithm for peak and average power optimization. First, power-oriented retiming is proposed to restructure the input SDFG in order to achieve parallelization to favor nodes with high power consumption followed by an MILP formulation for peak and/or average power optimization using MVS technique.

I. Introduction

High-level synthesis (HLS) is the process of mapping the behavioral specification of the system into register transfer description. The outcome of high-level synthesis is a structural description of the data path and a logical view of the control unit. High-level synthesis involves three main tasks, scheduling, allocation, and binding. The central task is the scheduling, which is the process of determining at which control step(s) each operation in the DFG executes. Although conventional design metrics such as performance, size and testability are important, the most crucial design metric nowadays is power. The demand for long-life batteries within tolerable size and weight and the reliability of integrated circuits are the main factors that dictate the low power design of embedded systems. Dynamic power due to charging and discharging in the CMOS gates is a dominant part in the total power consumption in CMOS circuits and is given by the following equation

$$P_{dynamic} = \frac{1}{2} \alpha C_L V_{dd}^2 f_{clock} \quad (1)$$

where C_L is the load capacitance at the gate output, f_{clock} is the circuit clock frequency, V_{dd} is the supply voltage, and α is the average number of transitions per clock cycle at the gate output, referred to as the switching activity.

Because of the quadratic relationship of the supply voltage to the dynamic power consumption, lowering

the supply voltage is considered as the most effective approach for minimizing power/energy consumption in circuits. However, lowering the supply voltage increases the circuit delay. Parallelization and pipelining can compensate for the introduced delay but at the cost of area overhead. Using multiple voltages scheduling (MVS) technique solves the problem. The idea is to assign the highest voltage level to the operations in the critical path to meet the time constraint and to use a lower voltage level for the operations in the non-critical paths to achieve the power/energy minimization.

Retiming was introduced by Leiserson and Saxe [1] as an optimization technique for the synchronous sequential digital circuits. Since retiming can be used to shorten the critical path delay and to increase parallelism among the computation nodes, it can be used to increase the number of computational elements that are candidates for voltage scaling.

In combinational circuits, many of the research have tackled the dynamic power minimization problem such as [5, 6, 7, 8, 9]. For synchronous sequential circuits, many research work have addressed the problem of minimizing the dynamic power (average power) consumption such as [2, 4, 3].

In this paper, we address the problem of minimizing dynamic peak and average power consumption in synchronous sequential circuits by first presenting a mathematical MILP formulation for the exact solution by simultaneously combining retiming and MVS for power (peak and/ or average) consumption optimization. This formulation is found to be expensive with a large number of variables resulting in long solution time; therefore we present an efficient two-step approach. We have derived a polynomial-time power-oriented retiming that is used first, followed by an MILP formulation for the resultant retimed circuit with the objective of optimizing power consumption. Experimental results demonstrate that our power-oriented retiming approach generates better graph structure candidates for reducing peak and average power consumption, compared with [2, 3] which only exploit maximizing the parallelism among graph nodes.

A. Synchronous Circuit Representation

A synchronous graph model, $G = (V, E, d, w)$, is used as in [1] to model the synchronous sequential

digital circuit; this is also referred to as a synchronous data flow graph (SDFG). Here V is the set of computational nodes, E is the set of interconnection edges between these nodes, d is a nonnegative number attached to each node representing its execution delay, and w is the set of weights in which each edge (u, v) has a weight $w(u, v)$ that is the number of registers associated with that edge. Figure 1 is an example of SDFG.

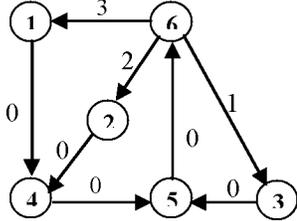


Figure 1: An example SDFG, nodes 1, 2, 3 are multiplication elements while nodes 4, 5, 6 are addition elements.

B. Notation

Here is some notation that is used in this paper:

$p(i, v)$: power consumed by operation i using voltage level v .

$d(i, v)$: delay (in # of control steps) of operation i using voltage level v .

P_j : power consumed by all functional units at step j .

P_{peak} : maximum power consumed over all steps.

The rest of this paper is organized as follows. Section II introduces the problem of unifying retiming and power optimization. In section III, we develop our power-oriented retiming algorithm followed by the second step as an MILP formulation for the resultant retimed circuit with the objective of optimizing power consumption. Section IV shows the results of some benchmarks to illustrate our proposed solutions. Section V concludes with a summary.

II. Unifying Retiming and Power Optimization

Combining retiming with voltage scaling is a useful technique for power optimization since it can restructure the critical path in such a way as to move power-expensive nodes out of the critical path and increase the number of nodes that are potential candidates for scheduling at lower supply-voltages.

While combining retiming and average power minimization can be done as in [2, 3], optimizing for peak power is harder to achieve by a simple formulation. This is because expressing the peak power consumption requires accounting for the contribution of the power consumption of a computation node towards power at every control step (or cstep) in which that computation node is active. In our MILP formulation, we introduce 0-1 variables to capture this activity information. Our MILP formulation is flexible

in that peak power can be treated as an objective to be optimized for or can be used as a constraint for those applications with hard limits on peak power. The mathematical foundation for the MILP formulation is based on the following lemma and theorem.

Lemma1:

Let $G = (V, E, d, w)$ be a synchronous sequential digital circuit and let λ be a positive real number. There exists a legal retiming r of G such that the clock period of the retimed graph G_r is less than or equal to λ , if and only if there is an assignment of a real value $h(v)$ and an integer value $r(v)$ to each node $v \in V$ that meets the following conditions:

1. $h(v) \geq 1 \quad \forall v \in V$
2. $h(v) + d(v) \leq \lambda + 1 \quad \forall v \in V$
3. $r(u) - r(v) \leq w(u, v) \quad \forall (u, v) \in E$
4. $h(u) - h(v) \leq -d(u) \quad \forall (u, v) \in E$ such that $r(u) - r(v) = w(u, v)$

Proof: It follows Lemma 9 in [1] by using $h(v) = s(v) - d(v) + 1$.

Define x_{ijv} to be a 0-1 unknown variable that takes value 1 if node i starts execution at cstep j with voltage level v and 0 otherwise. Then we can derive the following theorem.

Theorem 1:

Let $G = (V, E, d, w)$ be a synchronous sequential digital circuit, and let λ be a positive real number. Then, there exists a legal retiming r of G such that the clock period of the resultant retimed graph G_r is less than or equal to λ , if and only if there exists an assignment of a 0/1 value for each x_{ijv} , real values $h(i)$, and integer values $r(i)$ to each node $i \in V$ such that the following conditions are satisfied:

- T1. $\sum_v \sum_j x_{ijv} = 1 \quad \forall i \in V$
- T2. $\lambda R(i) - \lambda r(i) - \sum_v \sum_j j x_{ijv} = 0 \quad \forall i \in V$
- T3. $-\lambda R(i) + \lambda r(i) \leq -1 \quad \forall i \in V$
- T4. $\lambda R(i) - \lambda r(i) + \sum_v \sum_j x_{ijv} d(i, v) \leq \lambda + 1 \quad \forall i \in V$
- T5. $r(i) - r(l) \leq w(i, l) \quad \forall (i, l) \in E$
- T6. $\lambda R(i) - \lambda R(l) + \sum_v \sum_j x_{ijv} d(i, v) \leq \lambda w(i, l) \quad \forall (i, l) \in E$

Proof: By using a transformation similar to the one from [1] in which, for each node i , we substitute $h(i) = \lambda R(i) - \lambda r(i)$ in Lemma 1 to mathematically formulate condition (4) in Lemma 1. Since the quantity $h(i)$ as well as the quantity $\sum_v \sum_j j x_{ijv}$ represent the start

scheduling step, we can relate the two quantities by using condition T2 above. Thus, the proof follows the

proof of Lemma 1 noting that the delay of node i , $d(i) = \sum_v \sum_j x_{ijv} d(i,v)$ \square .

Using the introduced 0-1 variables, the power consumption at any cstep j can be written as $P_j = \sum_v \sum_i x_{ijv} p(i,v)$. The peak power consumption therefore is $P_{peak} = \max_j(P_j)$. Hence, the peak power constraint can be written as:

$$\sum_v \sum_i x_{ijv} * p(i,v) \leq P_{peak} \quad \forall cstep j \quad (2)$$

We can use a weighted sum of the average and peak power consumption as the objective function:

$$\text{Minimize : } \alpha P_{peak} + \beta (\gamma/\lambda) \sum_i \sum_j \sum_v x_{ijv} * d(i,v) * p(i,v) \quad (3)$$

Equations (2) and (3) and the set of constraints in Theorem 1 form the MILP formulation for simultaneous retiming and (peak and/or average) power optimization. This MILP formulation is suitable only for small circuits because of the increased number of 0-1 variables. Thus, we propose an efficient two-step heuristic based on the idea of power-oriented retiming, as it will be clear in the next section.

III. Two-Stage Power Optimization Solution

A. Motivating Example

Chabini and Wolf [3] have proposed a way of retiming with the objective of maximizing the total number of non-zero-delay (NZD) edges in order to maximize the parallelization among the graph nodes. Their approach results in many nodes being scheduled with lower supply-voltages and power-saving. But just maximizing the total number of NZD edges without taking the graph structure into consideration to control the choice of the edges does not exploit the potential rooms of the graph nodes to be scheduled with lower supply-voltages. As an Example, consider the SDFG in Figure 1, assuming that the circuit can be operated at two supply-voltages, v_H and v_L using the modules library in Table 1. Retiming SDFG in Figure 1 with the objective of maximizing the total number of non-zero-delay (NZD) edges under the time constraint of 4 results in the retimed graph G_{ru} shown in Figure 2(a). The maximum power saving can be obtained by scheduling nodes $\{1, 4\}$ at v_L and $\{2, 3, 5, 6\}$ at v_H and that results in 91 μ watt average-power saving. Optimizing G_{ru} for average and peak power results in the schedule shown in Figure 2(b) with 111.5 μ watt and 181 μ watt average and peak power respectively.

The SDFG in Figure 1 can be retimed differently with another objective, which we call *power-oriented* retiming, under the same time constraint of 4 to get the power-retimed graph, G_{rp} , shown in Figure 2(c). The

maximum power saving for Figure 2(c) can be obtained by scheduling nodes $\{1, 2, 4\}$ at v_L and $\{3, 5, 6\}$ at v_H and that results in 162 μ watt average-power saving. Optimizing the power-oriented retimed graph for average and peak power will result in the optimal schedule shown in Figure 2(d) with 82.5 μ watt average power and 116 μ watt peak power consumption, which is significantly lower than that for Figure 2(b). This example shows how power-oriented retiming can direct the choice of zero-delay edges for the sake of power saving (average and peak) which is the goal in this work.

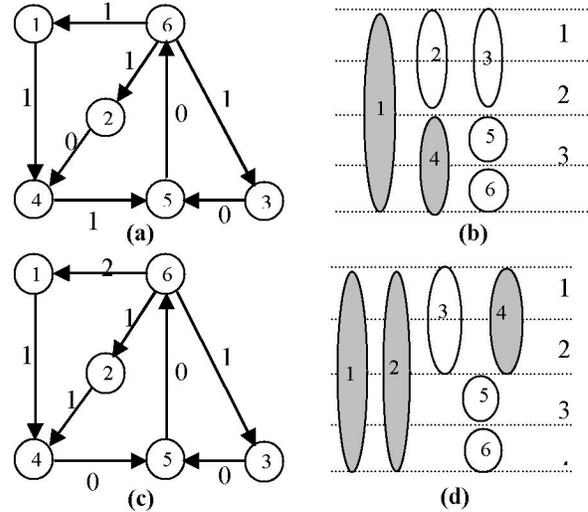


Figure 2: Results of retiming Figure 1 with two different retiming vectors for clock period = 4. (a) $r = (0 1 2 1 2 2)$; (b) power-optimized schedule of (a); (c) $r = (0 0 1 1 1 1)$; (d) power-optimized schedule of (c)

B. Power-Oriented Retiming

The main idea of power-oriented retiming is exploiting the circuit structure to impose control over edge selection in the retiming process in order to achieve power saving. This is achieved through associating with each edge (u,v) a cost function that captures the relative importance of edges according to the power consumption of the two end-nodes of this edge and achieve parallelism to favor the nodes with higher power consumption. The basic intuition here is that an edge with power-hungry nodes as end-points should not be on the critical path if at all possible. This allows power-expensive nodes to be scheduled at lower supply-voltages (which is reflected in average and peak power saving) and allows one to distribute the power-expensive nodes to different control steps obtain more peak power saving.

Other heuristics are possible here; we define the cost function attached with each edge as the sum of the power consumptions of the two end-nodes of that edge. Since the power-expensive nodes have larger execution delay relative to the other nodes, the simplest and yet

the most efficient cost function is to sum the delays of the two end nodes of that edge, i.e:

$$\text{cost}(u,v) = d(u) + d(v) \quad \forall \text{edge}(u,v) \quad (4)$$

In Figure 1, $\text{cost}(2,4) = 2+1 = 3$, while $\text{cost}(4,5) = 1+1 = 2$. Thus, the power-oriented retiming chose edge (4,5) instead of edge (2,4) as shown in Figure 2-(c) and Figure 2-(a) respectively. That left a room for node 2 to be scheduled in a lower supply-voltage as shown in Figure 2-(d).

Power-oriented retiming is based on the mathematical framework introduced by Leiserson and Saxe [1]. It is cast as an ILP formulation and because that the constraint matrix is *totally unimodular* (as it will be shown later), it turns out to be an LP formulation which can be solved in polynomial time. Power-oriented retiming is intended to solve the following problem:

Problem 1: Given a synchronous digital circuit represented as an SDFG, $G = (V, E, d, w)$, find a retiming $r: V \rightarrow Z$ that transforms G to a retimed graph $G_r = (V, E, d, w_r)$, which achieves power minimization while preserving the same circuit functionality as the original circuit.

Define $f(u,v)$ to be a 0-1 variable that takes on a value 1 if $w_r(u,v) = 0$, and 0 otherwise. For each edge (u,v) , this gives $f(u,v) + w_r(u,v) \geq 1 \quad \forall (u,v) \in E$; using (1), it can be written as:

$$-f(u,v) + r(u) - r(v) \leq w(u,v) - 1 \quad \forall (u,v) \in E. \quad (5)$$

The following theorem is the mathematical basis for power-oriented retiming and is based on Theorem 7 in [1].

Theorem 1:

Let $G = (V, E, d, w)$ be a synchronous digital circuit, and let λ be a positive real number. Then, there exists a legal retiming r of G such that the clock period of the retimed graph G_r is less than or equal to λ , if and only if there is a 0/1 value for each $f(u,v)$ variable and integer values for r , $r: V \rightarrow Z$ such that:

- $0 \leq f(u,v) \leq 1 \quad \forall (u,v) \in E$
- $-f(u,v) + r(u) - r(v) \leq w(u,v) - 1 \quad \forall (u,v) \in E$.
- $r(u) - r(v) \leq W(u,v) - 1 \quad \forall u, v \in V$
such that $D(u,v) > \lambda$.

Proof: Omitted; similar to the proof of Theorem 7 in [1].

The power-oriented cost function developed in Equation (4) when used in an objective function as in Equation (6) together with the set of inequalities in Theorem 1 forms the ILP formulation for the power-oriented retiming as follows:

$$\text{Minimize: } \sum_{(u,v) \in E} f(u,v) * \text{cost}(u,v) \quad (6)$$

$$-f(u,v) \leq 0 \quad \forall (u,v) \in E \quad (7)$$

$$f(u,v) \leq 1 \quad \forall (u,v) \in E \quad (8)$$

$$-f(u,v) + r(u) - r(v) \leq w(u,v) - 1 \quad \forall (u,v) \in E \quad (9)$$

$$r(u) - r(v) \leq W(u,v) - 1$$

$$\forall u, v \in V \text{ such that } D(u,v) > \lambda \quad (10)$$

It can be proved that the constraint matrix of the ILP formulation (6)-(10) for the power-oriented retiming is *totally unimodular* [10]. For lack of space, we do not discuss *total unimodularity* here; the reader is referred to [10]. The practical impact of *total unimodularity* is that linear programming (LP) formulation without the integrality constraints produces the optimal solution, which can be obtained in a polynomial run time.

As mentioned before, the second step in our solution is an MILP formulation for the DAG resulting from power-oriented retiming, which is discussed next.

C. Peak and Average Power Optimization for DAGs

The retimed graph is preprocessed to take off all the non-zero-delay edges. So, the resultant DFG is a graph representation of a combinational circuit in which each vertex represents a computation node, and the edges represent the precedence relation among the vertices in the same iteration.

Problem 2: Given a DFG representation of the design problem, the latency (greater than or equal to the minimum clock period of the retimed graph), a set of voltage levels for the operating resources, and a power/delay table that contains the average power consumption and the delay time needed for each resource operating on each voltage level. Find a schedule that minimizes the energy/power (average/peak) of the given DFG.

Our solution for the multiple supply voltages scheduling (MVS) problem is a mixed integer linear programming (MILP) formulation targeting peak power consumption as well as average power and energy consumption for a given latency constraint. Moreover, we are trying to integrate a combination of these factors simultaneously. The solution to problem 2 is much less expensive than the solution of the *simultaneous retiming and (peak and/or average) power optimization* problem because in the resultant DFG, the structure of the graph is fixed and so a time-frame for each node can be obtained using the as-soon-as-possible (ASAP) and as-late-as-possible (ALAP) schedule. Define x_{ijv} to be a 0-1 unknown variable that takes value 1 if node i starts execution at cstep j with

voltage level v and 0 otherwise. Then, the MILP formulation is as follows:

$$\text{Minimize} \quad \alpha P_{peak} + \beta (\forall \lambda) \sum_i \sum_j \sum_v x_{ijv} d(i, v) \cdot p(i, v) \quad (11)$$

$$\sum_v \sum_j x_{ijv} = 1 \quad \forall i \in V \quad (12)$$

$$\sum_v \sum_j (j + d(i, v) - 1) x_{ijv} + \sum_v \sum_j j x_{jiv} \leq -1 \quad \forall (i, l) \in E \quad (13)$$

$$\sum_i \sum_v \sum_{j_1=j-d(i,v)+1}^j x_{ijv} p(i, v) \leq P_{peak} \quad \forall j \in [1, \lambda] \quad (14)$$

$$\sum_v \sum_j (j + d(i, v) + 1) x_{ijv} \leq \lambda \quad \forall \text{node } i \text{ without successors} \quad (15)$$

Equation (11) is a flexible weighted objective function, where the weight factors can be set according to the design requirements. Equation (12) forces each node to start at only one cstep and scheduled using one and only one voltage level. The precedence relations are satisfied by Equation (13). Peak power can be set as a constraint or can be used as a variable to be minimized in the objective function as shown in Equation (14). Finally, each node without successors is forced to meet the latency requirement by Equation (15).

Power-oriented retiming strategy sometimes over-exploits circuit structure during retiming process, which may result in a retimed circuit that is not power-efficient candidate. To achieve the best power minimal schedule, the circuit is optimized through two passes of the two-stage algorithm using two different cost functions in the objective of the mathematical formulation in each. In one pass, the circuit is retimed using power-oriented retiming presented above; then the output retimed circuit is optimized for peak and/or average power consumption. In the other pass, the circuit is retimed using unit-weight cost in the objective function as presented in [3]; then the retimed circuit is also passed on to the same peak and/or average power optimization algorithm. The best power minimal solution is considered as the final schedule.

IV. Experimental Results

Our MILP formulation for the exact solution as well as the two-stage heuristic (power-oriented retiming followed by MVS) for peak and average power minimization are tested on standard benchmarks such as the differential equation solver (HAL), and several DSP benchmarks such as the fifth-order elliptic wave filter (EWF), third-order direct-form filter (DFF) and lattice filter (LF). Although the solution of the combined retiming and peak power MILP formulation finds the optimal solution, it suffers from inordinately long solution time because of the large number of binary variables needed to model peak power. In order to overcome this problem, the optimality can be

sacrificed for the sake of short solution run-time. Thus, we set a time limit for the solver (CPLEX in our case) to terminate with a feasible integer solution if the optimal solution is not found or verified yet.

The capability of our power-oriented retiming technique to restructure the sequential circuit to the favor of power minimization is compared to the unit-weight cost retiming (where the objective of the retiming process is just minimizing the non-zero-delay edges) by feeding both of the retimed DFGs to the same power optimization technique. The power optimization results for different benchmarks are tabulated in Table 2, using the module library in Table 1, for input timing constraints ranging from the critical path length to about double the critical path length. As shown in Table 2, our power-oriented technique is more efficient in minimizing both peak and average power than the unit-weight retiming for most benchmarks under different time constraints. In some benchmarks such as LF in Table 2-(d), the resultant retimed DFGs from both retiming techniques are the same, while under few time constraints, the unit-weight retiming gives better results as in Table 2-(a) under time constraint 8 csteps. This is because power-oriented retiming strategy sometimes over-exploits circuit structure during retiming process. The solution of the MILP formulation for the combined retiming and peak and average power consumption (for the restricted solution run-time) is inserted in Table 2 under the title “**Exact Sol**” to show the quality of the two-phase heuristic solution for peak and average power using less run time. That is in most benchmarks the two-phase heuristic matches well the solution obtained from the combined MILP formulation.

Table 1: Modules library

Module	5.0 V		3.3 V	
	del	power	del	power
MULT16	2	84	4	13
ADD16	1	26	2	6
SUB16	1	26	2	6

V. Conclusion

We have presented two methods for dynamic average power as well as peak power consumption in sequential synchronous circuits under time constraints using a combination of basic retiming and multiple voltage scheduling (MVS) techniques. Retiming technique is used to restructure the SDFG representation of the sequential circuit in order to increase the parallelism between operations and thus to increase the number of operations off the critical path to be candidates for scheduling at lower supply voltages. First, we devised an MILP formulation for optimal peak and/or average power consumption scheduling problem through a unification of retiming

and MVS techniques. Mathematical formulation for peak power dictates piece of information for each operation in order to capture its activeness in a certain time step. Thus, we used binary variables for that purpose, which turns to be very large because tight time-frame for an operation cannot be obtained apriori. Then, to alleviate the problem of variable explosion, we presented a two-stage algorithm for peak and/or average power optimization. First, power-oriented retiming is proposed to restructure the input SDFG in order to achieve parallelization to the favor of nodes

with high power consumption. Second, an MILP formulation is presented that takes the retimed DFG as an input and produces an optimal peak and/or average power schedule using MVS technique. Our proposed power-oriented retiming generates a graph structure candidate for better peak and average power saving than similar works that depends on only maximizing the parallelism among graph nodes as shown by the experimental results for many benchmarks.

Table 2: Power resultant of both power- oriented and unit-weight retiming

(a): HAL							(b): EWF						
L	Exact Sol		Unit-weight		Power-oriented		L	Exact Sol		Unit-weight		Power-oriented	
	Avg	Peak	Avg	Peak	Avg	Peak		Avg	Peak	Avg	Peak	Avg	Peak
6	143	181	162.3	252	144.8	181	16	111.375	204	111.4	204	111.4	204
7	106	123	124.3	194	124.2	181	17	111.647	168	96.7	168	96.7	168
8	66.5	97	78.3	110	92.75	168	20	65.9	107	83.85	181	77.5	116
12	31	39	42.5	84	32.8	39	22	54.136	91	80.5	168	55.63	97

(c): DFF							(d): LF						
L	Exact Sol		Unit-weight		Power-oriented		L	Exact Sol		Unit-weight		Power-oriented	
	Avg	Peak	Avg	Peak	Avg	Peak		Avg	Peak	Avg	Peak	Avg	Peak
4	130.5	155	159.5	220	130.5	155	8	79	110	79	110	79	110
5	104.4	149	104.4	149	106.6	149	10	51.6	84	51.6	84	51.6	84
6	67.67	78	67.67	78	67.67	78	12	33.33	36	33.3	36	33.3	36
10	38.4	39	38.4	39	38.4	45	16	22.25	26	22.3	36	22.3	36

References

- [1] C. E. Leiserson and J. B. Saxe, "Retiming synchronous circuitry," Tech Report: SRC-RR-13, Digital Systems Research Center, August 20, 1986.
- [2] N. Chabini, I. Chabini, E.-M. Aboulhamid, and Y. Savaria, "Unification of basic retiming and supply voltage scaling to minimize dynamic power consumption for synchronous digital designs," in *Proc. Great Lakes Symp. VLSI*, Washington, D. C., pp. 221-224, April 2003.
- [3] N. Chabini and W. Wolf, "Reducing Dynamic Power Consumption in Synchronous Sequential Digital Designs Using Retiming and Supply Voltage Scaling," *IEEE Trans. on VLSI Systems*, Vol. 12, No. 6, pp. 573-589, June 2004.
- [4] J. Monteiro, S. Devadas, A. Ghosh, "Retiming sequential circuits for low power," in *Proc. 1993 IEEE/ACM International Conference on Computer-Aided Design*, pp.398-402, November 07-11, 1993.
- [5] A. Raghunathan and N.K. Jha, "An ILP formulation for low power based on minimizing switched capacitance during data path allocation", in *Proc. IEEE Symposium on Circuits and Systems*, 1995.
- [6] Y.-R. Lin, C.-T. Hwang, and A. C.-H. Wu, "Scheduling techniques for variable voltage lowpower designs," *ACM Trans. Design Automat. Electron. Syst.*, Vol. 2, No. 2, pp. 81-97, April 1997
- [7] M.C. Johnson and K. Roy, "Datapath scheduling with multiple supply voltages and level converters," *ACM Trans. Design Automation Electronic Syst.*, Vol. 2, No. 3, pp. 227-248, July 1997.
- [8] J.-M. Chang and M. Pedram, "Energy minimization using multiple supply voltages," *IEEE Trans. on VLSI Systems*, Vol. 5, No. 4, pp. 436-443, Dec. 1997.
- [9] A. Manzak and C. Chakrabarti, "A low power scheduling scheme with resources operating at multiple voltages," *IEEE Trans. on VLSI Systems*, Vol. 10, no 1, pp. 6-14, Feb 2002.
- [10] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*, John Wiley, New York, 1988.