Hindawi Publishing Corporation
VLSI Design
Volume 2009, Article ID 156751, 10 pages
doi:10.1155/2009/156751

Research Article

Floorplan-Driven Multivoltage High-Level Synthesis

Xianwu Xing and Ching Chuen Jong

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798

Correspondence should be addressed to Ching Chuen Jong, eccjong@ntu.edu.sg

Received 13 November 2008; Revised 23 March 2009; Accepted 20 June 2009

Recommended by Sheldon Tan

As the semiconductor technology advances, interconnect plays a more and more important role in power consumption in VLSI systems. This also imposes a challenge in high-level synthesis, in which physical information is limited and conventionally considered after high-level synthesis. To close the gap between high-level synthesis and physical implementation, integration of physical synthesis and high-level synthesis is essential. In this paper, a technique named FloM is proposed for integrating floorplanning into high-level synthesis of VLSI system with multivoltage datapath. Experimental results obtained show that the proposed technique is effective and the energy consumed by both the datapath and the wires can be reduced by more than 40%.

1. Introduction

The ever advance of semiconductor technology has inspired many new research topics in high-level synthesis. In VLSI systems, interconnects, specifically connecting wires, have high impact on many performance parameters such as area, delay, and power/energy consumption. It was reported that power consumed by interconnects (wires, multiplexers MUX, buffers, etc.) could contribute at least 20% of the total power dissipation [1] in a VLSI system and could be up to 60% in some cases [2]. The high-level synthesis of VLSI system is a top-down process, with the high-level synthesis preceding the physical synthesis much earlier in the process. Therefore, there is little information of physical level available for the high-level synthesis tasks such as operation scheduling and module binding. As a result, the optimal results in the high level are most likely not the optimal ones in the physical level.

To reduce the gap between high-level synthesis and physical level design, predicting and extracting physical level information during high-level synthesis has been attracting much research attention [3–7]. As the floorplanning process is usually the first stage of physical synthesis, it is relatively closer to as well as easier and less computationally intensive to be incorporated into high-level synthesis. The information of wire connection obtained during floorplanning is vital to high-level synthesis. Although the exact positions (placement) of hardware blocks and their connections (routing) are not available in the floorplan, wire length can be much more accurately estimated at the floorplanning stage than at the behavioral level. Several high-level synthesis techniques that incorporate the floorplan information for optimizing the power consumed by connecting wires were found in the literatures. They are summarized hereinafter.

In [8], a technique is proposed for simultaneous scheduling, binding, and floorplanning based on the simulated annealing algorithm. The combined scheduling and binding problem is formulated as a placement problem in a two-dimensional table. When a local move is performed, fast evaluation is performed by using a constructive power driven floorplanning algorithm that considers both the switching activity on the wires and the wire length. The technique is effective in reducing interconnect power. However, as power consumed by datapath is not accounted, the tradeoff between power consumed by interconnects (interconnect power or wire power) and the power consumed by datapath (datapath power) is not explored.

In [6], the floorplan-driven low-power high-level synthesis is performed by an algorithm that consists of two levels of simulated annealing algorithm. The inner loop uses the floorplanning algorithm proposed in [9] with the wire power consumption as its cost function to find a power-optimized floorplan. The binding is refined in an outer loop to find a more power efficient floorplan such that the wire power consumption is optimized.

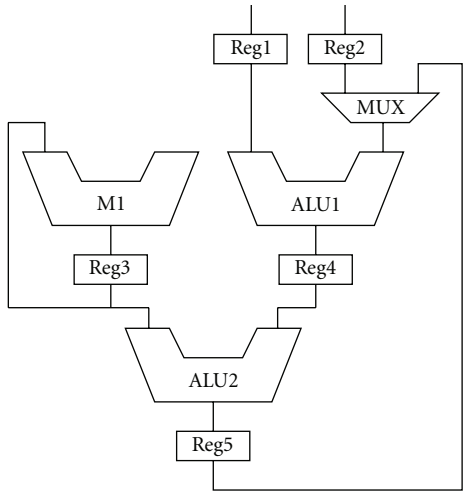

Figure 1: A datapath example.

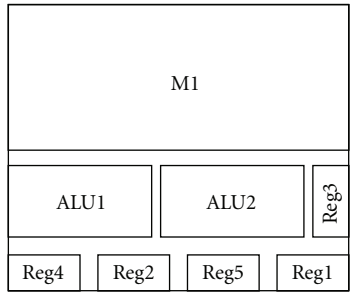

Figure 2: A floorplan for Figure 1.

An innovative technique presented in [7] performs simultaneous power-driven binding and floorplanning based on a probabilistic approach [10] to improve the estimation accuracy of the net parameters, such as delay and power. In the probabilistic approach, the distribution of net length rather than the fixed estimation is modeled in such a way that a range of values for the net length together with their associated probabilities is considered. Using the probabilistic model, the technique aims to maximize the likelihood of meeting the design constraints while minimizing the overall power. The probabilistic model also achieves more accurate estimation of power during optimization. One of the drawbacks is that the influence of scheduling, which impacts the binding, is not considered.

Loosely coupled high-level synthesis and physical synthesis usually suffers from largely increased CPU time. In [11], an effective algorithm is proposed to achieve design closure for integrated high-level synthesis and physical synthesis. Dramatically reduced CPU time has been reported. [11] demonstrates that integrated synthesis is able to meet algorithmic complexity requirements in practical applications.

In this paper, a technique named FloM (Floorplan-driven Multivoltage Synthesis) is proposed. Because voltage has quadratic impact on power consumption, we attempt to optimize power by applying multivoltage datapath synthesis in FloM. In literatures, some multivoltage synthesis techniques [12–16] are reported. These techniques tackle multivoltage synthesis and achieve good power reduction, but none of them considers the physical level information during the synthesis. Here, we proposed a floorplan-driven multivoltage high-level synthesis technique. The technique, which is based on the simulated annealing approach, not only considers different supply voltages for the functional units (FUs) but it also generates and evaluates different scheduling and bindings as well as the corresponding floorplans. As a result, the energy consumed by both datapath and interconnect is effectively optimized.

The rest of the paper is organized as follow. Section 2 discusses the influence of floorplanning on power optimization of multivoltage datapath. The proposed floorplan-driven multivoltage high-level synthesis technique is described in Section 3 in details. The experimental results obtained for testing FloM with some benchmarks and the comparisons with some existing techniques are presented in Section 4. Section 5 concludes the paper.

2. Interconnect Power Optimization and Multivoltage Design

As interconnects consume a considerable amount power, optimization of interconnect power should be considered at all the design levels. In high-level synthesis, interconnect information should be included if accurate power estimation, and thus better optimization result, is to be achieved. This section discusses the impact of floorplanning on power and the floorplanning for multivoltage design.

2.1. Impact of Floorplanning on Power. Floorplanning is an essential step in physical synthesis. It determines the relative positions of all the blocks, including FUs, registers, and memories, in a design. Although a floorplan does not provide detailed physical information, it guides the lower-level physical synthesis steps, such as placement and routing. Thus floorplanning has a major impact on the final physical implementation. Consider the datapath example shown in Figure 1, which consists of 1 multiplier (M1), 2 ALUs and 5 registers. A feasible floorplan is shown in Figure 2.

The floorplan provides a good estimation of the circuit area. It determines the relative positions of the hardware resources, thus influencing the interconnects among the resources. Consider another floorplan shown in Figure 3. Obviously this floorplan occupies a much larger area than that of Figure 2 does. Some spaces are wasted due to bad positioning. As floorplanning is crucial in physical synthesis and is critical for placement and routing, many algorithms have been proposed for floorplan optimization in literature. These algorithms mainly target at area optimization, trying to group resources together as compact as possible [9, 17]. There also exist some algorithms which consider other parameters, such as the total length of connecting wires [18, 19]. In recent years, the impact of interconnects on power and energy consumptions has attracted unprecedented attention [6–8].

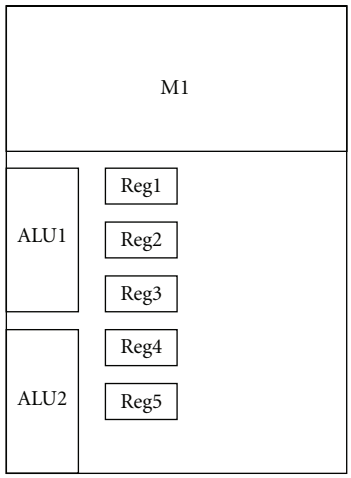

Figure 3: A larger floorplan for Figure 1.

In general, the power of wire can be modeled as (1)

$$P_{\text{wire}} = \alpha \cdot c \cdot f \cdot V^2 \cdot \sum L_i, \tag{1}$$

where α is the average switching activity (SA) occurred on the wire, V is the voltage of the signal that flows through the wire, f is the operating frequency, c is the wire capacitance per unit area, and $\sum L_i$ is the total wire length. In the floorplanning stage, detailed routing and layout information are not available. Therefore the total wire length can be modeled as(2)

$$\sum L_i = \sum_{i=1}^{n} \sum_{j=1}^{n} N_{i,j} \cdot M_{i,j} \cdot \text{distance}\left(p_i, p_j\right), \tag{2}$$

where $N_{i,j}$ is the number of connections from the outputs of resource i to the inputs of resource j, $M_{i,j}$ is the bit width, and distance(p_i, p_j) is the distance between the center points of the two resources, that is, the average length of the connecting wires between two resources is estimated by the distance between their center points.

2.2. Multivoltage Design. In a multivoltage datapath, different FUs operate with different power supplies. Multivoltage technique utilizes the available timing slacks to reduce power by lowering the voltage supply of FUs that are not in the critical path, as lowering the voltage of an FU reduces the power consumed by the FU but increases its delay. Recently multivoltage technique has gained more attention in both low-power design and low-power synthesis because power has a quadratic dependence on voltage supply, which makes lowering voltage very efficient for power optimization. An example of multivoltage datapath is shown in Figure 4.

In Figure 4, there are two FUs (FU1 and FU2). FU1 is supplied with 5.0 V and FU2 with 3.3 V. The register Reg1 (Reg2) that stores the output of FU1 (FU2) is supplied with the same voltage as the FU so that they can be placed close to each other. An FU can take input signals with voltage equal to or higher than its supply voltage. For example, FU2

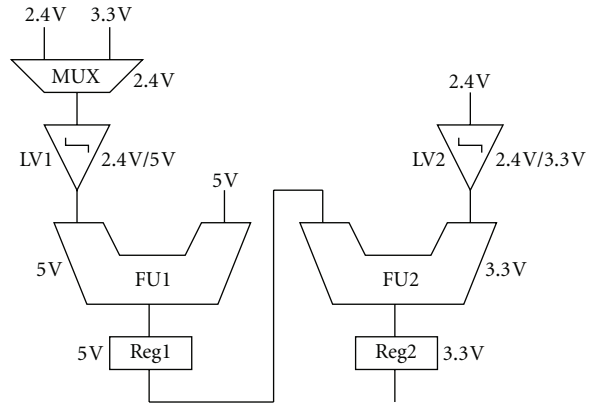

Figure 4: A multivoltage datapath.

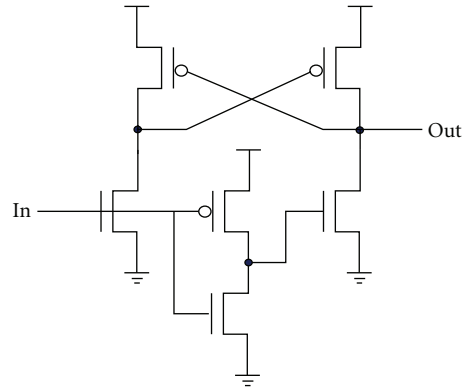

Figure 5: DCVS level converter.

(3.3 V) takes a 5.0 V signal (from Reg1) as one input and a 3.3 V signal (from the level converter LV2) as the other. Similarly, the multiplexer (MUX) takes a 2.4 V signal as one input and a 3.3 V signal as the other. If a lower-voltage signal drives a higher-voltage signal, a level converter must be used. Therefore the output of MUX (2.4 V) cannot be connected directly to the input of FU1 (5.0 V). Instead, a 2.4 V to 5.0 V level converter (LV1) is placed between them to convert the 2.4 V signal to a 5.0 V signal. LV2 is a converter for 2.4 V to 3.3 V. Generally level converters convert lower-level signals into higher-level signals so that the signals can act as drivers to modules. There are two parameters to be considered in a level converter: the input signal level and the output signal level. Note that the power consumption of a level converter can be considerably large and may vary according to the two parameters. Figure 5 illustrates a Differential Cascade Voltage Switch (DCVS).

In high-level synthesis, power optimization is usually performed under a resource constraint in which every resource has a preassigned supply voltage. On the contrary, the technique we propose here does not predefine the voltage of each resource but intelligently assigns the optimal supply voltage to each resource on the fly. A thorough power estimator based on (3), which considers the power consumed

by registers, MUXes, level converters, wires as well as FUs, is used to evaluate the power cost during the synthesis

$$P_{\text{total}} = P_{\text{FU}} + P_{\text{reg}} + P_{\text{wire}} + P_{\text{level\_conv}} + P_{\text{MUX}}, \quad (3)$$

where P_{FU}, P_{reg}, P_{wire}, $P_{\text{level\_conv}}$, and P_{MUX} are the power consumed by FUs, registers, wires, level converters, and multiplexers, respectively. Note that under the assumption that the capacitance of a wire does not change with its voltage level, P_{wire} is formulated as (4)

$$P_{\text{wire}} = COEFF_{\text{wire}} \cdot \sum L_i \cdot V_i^2 , \quad (4)$$

where $COEFF_{\text{wire}}$ is the product of SA (α), capacitance (c) and frequency (f). It can be seen that the optimization of power on interconnects in a multivoltage design is no longer the same as the optimization of the total wire length.

2.3. Floorplan Optimization for Multivoltage Design. In a multivoltage design, the optimization of wire power is more difficult than the optimization of just the total wire length. The challenge is that the supply voltage of a resource is not fixed but is determined by the behavioral synthesis algorithm. The scheme of on the fly determination of voltage supply imposes a dilemma for both the behavioral synthesis and the floorplanning. On one hand, the detailed floorplan information is not available during the behavioral synthesis; hence the power consumed by wire cannot be estimated. On the other hand, both the schedule and resource binding and the actual supply voltage of a resource are not determined yet at the floorplanning stage; hence the wire power cannot be calculated with (4) because V_i is not known. Therefore a single sequential execution of behavioral synthesis and floorplanning is unlikely to achieve good result even if the behavioral synthesis algorithm and the floorplan algorithm both render the optimal results. Based on this observation, an integrated algorithm is proposed in the next section to address this dilemma.

In addition to the difficulty of integrating the behavioral synthesis process and the floorplanning process, the multivoltage model further complicates the floorplanning process with another issue. In [13], it was pointed out that multivoltage design imposes a challenge on the circuit layout and it is desirable to partition the circuit into several regions, each operating at its own voltage. Some circuit isolation is needed to mitigate the crosstalk of those regions, probably by increasing substrate contacts as well as spacing between wells. This makes existing design tools hard to handle, thus considerably increases the work of layout design. As such, multivoltage design is not an easy task in the real world. A real-world multivoltage media processor is presented in [20], in which two supply voltages are used. It shows that considerable amount of power can be saved in a multivoltage design.

3. Algorithm Formulation of FloM

As discussed above, it is difficult to achieve an optimal result in terms of both power consumption and floorplan, due to the unknown lower-level floorplan in the high-level synthesis stage. In addition, as the supply voltage of a resource is determined on the fly during optimization, the floorplanning process must consider the supply voltages, because the resources with the same voltage level need to be grouped together in the floorplan (Section 2.3). An optimal behavioral synthesis result may not lead to an overall optimal result when the power consumed by wire is also a factor to be optimized. Therefore, these two processes must be integrated in one way or another, so that the overall cost is estimated relatively accurate during the synthesis process.

3.1. Algorithm Flow. The proposed algorithm FloM incorporates a behavioral synthesis and a floorplan synthesis. It iteratively refines the behavioral synthesis result and the floorplan result in order to optimize the overall power consumption. The algorithm flow is shown in Figure 6 and described below.

FloM takes as its input a Data Flow Graph (DFG), a library of components, and various constraints, such as resource constraint, delay constraint (in real time unit), clock cycle time, and the voltage domain (available supply voltages). After an initial scheduling and binding is performed, FloM starts an outer-level simulated annealing process. In each iteration of the simulated annealing process, a floorplanner (another level of simulated annealing process described below), a power/energy estimator, and a local transformation are performed in sequence, until an acceptable synthesis result is obtained. The outer-level process produces an optimized result for scheduling, binding as well as floorplanning. The cost function used is the total power/energy consumption by modules, registers, MUXes, level converters, and wires. The local moves are described below.

3.2. Local Moves in the Outer-Level Algorithm. A local move is a move that changes the voltage assignment of a resource, or the resource binding, or the operation schedule. Whenever a local move is performed, an optimized floorplan corresponding to the scheduled and bound DFG is generated by an inner-loop simulated annealing process. Therefore, a local move changes also the floorplanning, and hence the value of the cost function as defined in (3). In FloM, we propose five types of move, namely, voltage scaling, FU rebinding, FU swap, register rebinding, and register swap. These moves change only a small portion of the datapath, therefore only local resynthesis is needed, and the other parts of datapath are untouched, leading to an incremental datapath synthesis.

3.2.1. Voltage Scaling. Voltage scaling changes the supply voltage of an FU from $v0$ to $v1$. If $v0 < v1$, the timing will certainly still be met, because the latency of the resource is decreased with the new higher voltage supply. Therefore rescheduling is not necessary. If $v0 > v1$, rescheduling is performed on all the operations executed on the FU and all their successors. The rescheduling will usually cause some succeeding operations being rescheduled into later clock

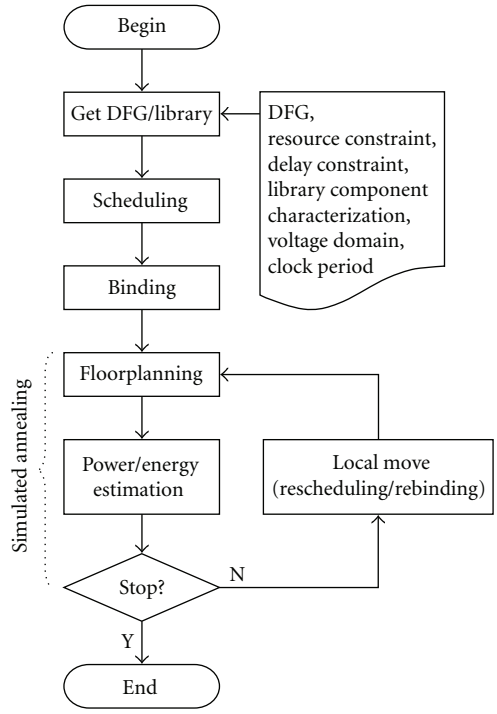

Figure 6: Overall algorithm flow of FloM.

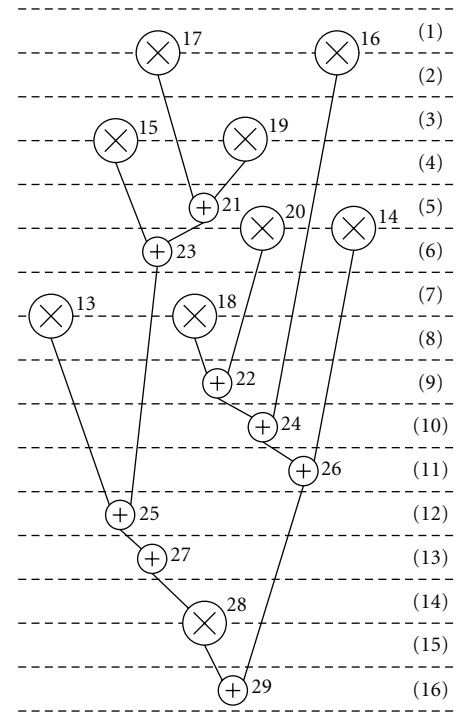

Figure 7: DFG of an IIR filter.

cycles than their original ones. A valid voltage scaling move must not incur violation on either the latency constraint or the resource constraint. We illustrate the voltage scaling move with the DFG of an IIR filter shown in Figure 7.

Suppose that Operations 20 and 28 are executed on a resource R1. If the voltage of R1 is scaled up from 2.4 V to 3.3 V, the latency of R1 is reduced, the time taken to complete Operations 20 and 28 is reduced, and the timing constraint will not be violated. Note that this move requires reallocation and rebinding of registers, because the voltage of all the registers that store the input and output data of R1 must change accordingly, resulting in possible register sharing conflict. Suppose that a register REG1 (2.4 V) stores both the data of R1 (2.4 V) and R2 (2.4 V). If the voltage of R1 is now scaled to 3.3 V, sharing REG1 by R1 and R2 will no longer be possible. The number of registers needed may also change, thus a register reallocation process must be performed. Scaling up the voltage of a resource will generally increase the power, but as it will change other power factors, such as registers, MUXes, level converters, and floorplan, the overall power consumption may decrease. Even if the overall power consumption increases, unfavorable moves are allowed in the simulated annealing algorithm in order to introduce new favorable moves.

On the other hand, if the voltage of a resource is scaled down, timing violation may occur. For example, if the voltage of R1 is scaled from 3.3 V to 2.4 V, the latency of all the operations (say again, Operations 20 and 28) is increased. Suppose that the latency now spans three clock cycles, there would be a timing violation on Operation 28, because its operating time overlaps with its successor, Operation 29. Note that in this case a resource violation also takes place in Operation 20, because it now operates as late as in Control-step (C-step) 7, making three multiplications (20, 13, 18) running in the same C-step. This would require three multipliers (possibly with different voltages), breaking the resource constraint, as originally only two multipliers are needed. Therefore, both timing violation and resource violation are possible when the voltage of a resource is scaled down. When a violation does occur, resource-constrained rescheduling must be performed to balance the operation distribution. If it fails, the move is considered as an invalid move and it is rejected. The rescheduling is performed in two steps.

(i) Update the delay of all the operations, N, on which the voltage scaling move is performed, and sort the operations in ascending order of their start time.

(ii) Sequentially, move each operation $n_i \in N$ upwards as much as possible, so that operation dependency is maintained, and the start time of n_i is larger (later) than the end time of n_{i-1}. If the new end time of n_i is larger than one of its successors, all the operations whose start time is larger than the end time of n_i are pushed downwards to satisfy the dependency constraint.

3.2.2. FU Rebinding. This move rebinds an operation n from Resource R1 to another compatible and vacant resource R2 with a different supply voltage. Similar to the voltage scaling, if the voltage of R1 is lower, no timing violation will occur, and only reallocation and rebinding of registers are needed. Otherwise, local rescheduling of some succeeding

operations is performed to resolve any timing violation. Note that R2 must be vacant during the lifetime of n; otherwise resource conflict occurs. If this condition is satisfied, resource violation will not occur. This move also requires reallocation of the registers that are associated with the affected FUs.

3.2.3. FU Swap. This move swaps the bindings of two operations (OP1 and OP2) to FUs (FU1 and FU2). The two FUs must have the same supply voltage, and there is no operation lifetime conflict after the swap, which means that the move does not require rescheduling and will only affect the two swapped FUs. The operation lifetime conflict occurs in a situation like this: suppose that another operation OP3, which is bound to FU2, operates in the same control steps as OP1. After the swap, OP1 would also be executed on FU2; so now OP1 and OP3 are bound to the same resource (FU2) and operate in the same control steps, resulting in a resource violation.

3.2.4. Register Rebinding. This move rebinds an edge E1 (a value) of the DFG from a register REG1 to another register REG2 with the same supply voltage. This move does not require rescheduling and will only affect the two swapped registers. A valid register rebinding move must incur no violation on the lifetime of the variables. The variable lifetime violation occurs in a situation like this: after reassigning E1 to REG2, there is another edge E2 which is also bound to REG2, and the lifetimes of E1 and E2 overlap.

3.2.5. Register Swap. This move swaps the bindings of two variables (E1 and E2) to registers (REG1 and REG2). The two registers must have the same supply voltage, and no variable lifetime conflict must occur after the swap. This move does not require rescheduling and will only affect the two swapped registers. The variable lifetime conflict happens if, after the swap, the lifetime of E1 overlaps with the lifetime of any edge that is assigned to REG2, or the lifetime of E2 overlaps with the lifetime of any edge that is assigned to REG1.

In each iteration of the simulated annealing process, one of the five local moves is randomly selected to generate a new solution. A set is used to record the FUs/registers that are selected to perform a move in an iteration. In the next iteration, the FUs/registers in the set are given lower priority to be selected for a move, and the set is updated with the newly selected FUs/registers. The local moves work together to explore synthesis alternatives to relieve local effects. Some moves, such as the change of resource voltages, FU binding, and register binding, also result in changes in interconnects, including MUXes and wires so that better solutions can be found for power/energy optimization. The experimental results (see Section 4) show that they are adequate for achieving good results.

3.3. Floorplanning Algorithm for Multivoltage Design. As can be seen from the algorithm flow of Figure 6, whenever a synthesis solution is found, a floorplanner is executed to search for an optimized floorplan corresponding to the synthesized multivoltage datapath. The floorplanner partitions the datapath into different regions, each of which is occupied by floorplan blocks with same supply voltage. Floorplan blocks include FUs, registers, and MUXes. The only objective of the floorplanner is to optimize the power consumed by the wires. The power consumed by FUs, registers, MUXes and level converters is not considered here, because it is not determined by the floorplan, but by the synthesis process.

In general, floorplan can be categorized as slicing floorplan or nonslicing floorplan. As indicated in [21], slicing structures have several advantages and have been widely explored. One of the most popular representations of slicing floorplan is the normalized Polish expression, first proposed in [9] and popularly adopted afterward [6, 18]. The floorplanning algorithm in FloM is based on a modified simulated annealing algorithm proposed in [9]. FloM makes two modifications in order to handle multivoltage design with power optimization, which was not considered in [9]. (i) FloM aims to optimize the total power consumed by the wires, while the original algorithm optimizes area and total wire length. These are different optimization targets according to (4), because, in multivoltage design, different wires could carry signals of different voltages. (ii) FloM partitions the datapath into multiple regions; each consists of resources with the same supply voltage, while the original algorithm handles floorplan with a single voltage supply.

In FloM, a floorplan consists of several super blocks, each of which consists of blocks with the same voltage. Therefore there are two levels of Polish expressions. The first level has one normalized Polish expression, representing the floorplan of the super blocks. The second level has several normalized Polish expressions, each representing the floorplan of one super block. An optimal floorplan solution is the one yielding the minimal cost for the overall flattened floorplan. As there are two levels of Polish expressions, a local move can perform on either the upper level Polish expression or one of the lower level Polish expressions. The details on Polish expressions and the local moves of the simulated annealing algorithm can be found in [9].

One of the major advantages of iterative algorithms is that the optimization target can easily be adjusted by simply changing the cost function, which is conventionally the area and the total wire length. FloM uses (3) as the cost function, which shows that in a multivoltage design the floorplan with the shortest wire length does not necessarily means a solution with the minimum power/energy consumption in wire. FloM can also be modified easily to take a composite cost function consisting of the power/energy consumption, the total wire length, and the area, each with a different weight.

4. Experimental Results

To evaluate the efficiency of FloM, the experiments on the benchmarks [12] listed in Table 1 were carried out, where the latency is in terms of the number of clock cycles. In the experiments, the clock period was set at 30 nanoseconds. The voltages allowed were 5.0 V, 3.3 V, and 2.4 V. The delay (D in nanosecond) and energy (E in pJ) characterizations

Table 1: Benchmarks and constraints for testing FloM.

Benchmark	Number of MULT	Number of ALU	Latency
EWF	1	3	25
IIR	2	1	16
DCT	2	3	25
DiffEq	2	1	10
FDCT	2	3	25
FIR	2	2	20
TFIR	2	2	15
Lattice	2	1	20

Table 2: Delay and energy characterization of FUs.

FU	5.0 V		3.3 V		2.4 V	
	D	E	D	E	D	E
ALU	25.7	57	45.5	25	76.8	13
Multiplier	54.0	2202	96.6	960	163.7	507
Mux	—	9	—	4	—	2

Table 3: Energy characterization of level converter.

V1 \ V2	5.0 V	3.3 V	2.4 V
5.0 V	—	—	—
3.3 V	178.1	—	—
2.4 V	139.4	53.04	—

of the FUs were adopted from [22] and reported in Table 2. Note that the delay of an FU is the worst case delay, which is computed by (5)

$$d = d_{\mathrm{FU}} + d_{\mathrm{MUX}} + d_{\mathrm{Reg}} + d_{\mathrm{Conv}}, \tag{5}$$

where d_{FU}, d_{MUX}, d_{Reg}, and d_{Conv} are the delay of FU, multiplexer, register, and level converter, respectively.

For registers, the same ALU to register power consumption ratio as reported by the Synopsys Design Compiler on a TSMC 0.6 μm CMOS cell library is used, and thus the energy consumption of a register with 5.0 V, 3.3 V, and 2.4 V is 57/3, 25/3, and 13/3, respectively. The energy characterization of level converters is reported in Table 3. Note that level converter is not needed when a high-voltage cell drives a low-voltage cell and is marked as "—" in Table 3. The following sections report the various experimental results on datapath and floorplan obtained by FloM. The energy consumption on datapath is estimated assuming that average switching activity is 0.5. The energy consumption on floorplan is estimated assuming that average wire length between two blocks is the distance of their central points.

4.1. Experimental Results on Synthesis. In this section, the results obtained from the experiments carried out to evaluate the multivoltage low power synthesis algorithm, specifically, the five local moves described in Section 3.3, are presented. As the floorplanning process does not affect the moves, it is not invoked in this experiment, and the cost function of the outer level simulated annealing algorithm is defined as the total energy consumed by the datapath, that is, (3) excludes P_{wire}. The results obtained are shown in Table 4, where E1 is the energy consumption when list scheduling and single voltage (5.0 V) are used. E2 is the energy consumption when list scheduling and multivoltage are used (the process first fixes the scheduling and then assigns voltages as low as possible to every resource under timing constraints). E3 is the energy consumption when the proposed outer level simulated annealing algorithm of FloM is used. It is observed that by applying multivoltage assignment, an average improvement of 17.8% (over E1) is achieved. This reduction is purely caused by the reduction of voltage supply of some resources (scheduling is not changed). The proposed algorithm refines the solution (including rescheduling) iteratively and achieves an average energy reduction of 42.4% over E1. This further reduction is because the scheduling is refined by the five local moves proposed. As different scheduling provides different space for multivoltage assignment, larger design space is explored by FloM.

Table 4: Experimental results on datapath energy consumption (pJ).

Benchmark	List Sch + Single V	List Sch + Multi V		FloM	
	E1	E2	−Δ%	E3	−Δ%
EWF	20525	18723	8.8	13634	33.6
IIR	21077	17656	16.2	13285	37.0
DCT	42834	29123	32.0	23663	44.8
DiffEq	13964	10545	24.5	7852	43.8
FDCT	38169	32821	14.0	21773	43.0
FIR	27045	23774	12.1	9191	66.0
TFIR	25834	21161	18.1	16442	36.4
Lattice	30129	26719	11.3	20637	31.5
Average	—	—	17.8	—	42.4

Table 5: Dimensions of resources (Unit).

Resource	Multiplier	ALU	Register
Dimensions	5×20	3×8	2×4
	10×10	4×6	

Table 6: Experimental results on wire energy consumption (pJ).

Benchmark	Optimize Area	Optimize Wire Length		Optimize Energy	
	E1	E2	−Δ%	E3	−Δ%
EWF	7286	5421	25.6	4138	43.2
IIR	6892	4818	30.1	4287	37.8
DCT	10275	6915	32.7	5620	45.3
DiffEq	4100	2821	31.2	2476	39.6
FDCT	11236	7292	35.1	5315	52.7
FIR	7967	5832	26.8	5035	36.8
TFIR	8360	6713	19.7	5743	31.3
Lattice	8261	5973	27.7	4337	47.5
Average	—	—	28.9	—	42.6

Table 7: Experimental results on total energy consumption (pJ).

Benchmark	Separate Flow				Integrated Flow				Improvement (%)		
	Wire	Datapath	Total	Time (s)	Wire	Datapath	Total	Time (s)	Wire	Datapath	Total
EWF	7286	13634	20920	5.34	4138	14547	18685	16.22	43.2	−6.7	10.7
IIR	6892	13285	20177	1.69	4287	13843	18130	4.77	37.8	−4.2	10.1
DCT	10275	23663	33938	9.01	5620	25509	31129	62.04	45.3	−7.8	8.3
DiffEq	4100	7852	11952	0.21	2476	8152	10628	1.05	39.6	−3.9	11.1
FDCT	11236	21773	33009	9.30	5315	23754	29069	55.64	52.7	−9.1	12.0
FIR	7967	9191	17158	4.69	5035	9384	14419	30.21	36.8	−2.1	16.0
TFIR	8360	16442	24802	1.84	5743	16985	22728	12.37	31.3	−3.3	8.4
Lattice	8261	20637	28898	1.98	4337	23567	27904	13.08	47.5	−14.2	3.4
Average	—	—	—	—	—	—	—	—	42.6	−7.3	9.5

Table 8: Average improvement (%) comparisons on datapath.

Algorithms	[14]	[23]	[15]	FloM
Energy Δ(%)	40.19	43.29	39.00	42.40

Table 9: Benchmark improvement (%) comparison on datapath.

Benchmark	[14]	[23]	[15]	FloM
FIR	—	41.55	33.20	66.00
EWF	44.47	43.01	44.20	33.60
DCT	—	43.58	30.40	44.80
FDCT	43.44	—	—	43.00
DiffEq	40.87	—	53.00	43.80

Table 10: Energy reduction (%) comparisons.

Benchmark	[6]		FloM	
	Wire	Datapath	Wire	Datapath
FDCT	35.63	−0.79	52.70	−9.10
FIR	37.79	−0.07	36.80	−2.10
DiffEq	29.05	−0.02	39.60	−3.90
Average	41.24	−7.70	42.60	−7.30

4.2. Experimental Results on Synthesis and Floorplanning. In the following experiments, the floorplanning process is included. To compute the floorplan area and wire energy consumption, FloM uses a simple model for the area characterization of modules and wire capacitance. The area of multipliers, ALUs, and registers used are listed in Table 5, where "unit" is used. The wire capacitance is modeled as 0.01 pf per unit. In fact, the proposed technique does not depend on area and power models. New and more accurate models can easily be adopted by simply changing the cost function.

As FloM tries to integrate seamlessly high-level synthesis and floorplanning for low power/energy, we investigate the wire energy consumption under different optimization schemes in order to evaluate the efficacy of the proposed low-power floorplanning technique. The results are shown in Table 6, where E1 is the wire energy consumption obtained by the proposed power-conscious iterative synthesis algorithm with the floorplanning targeting at area minimization. E2 is the wire energy consumption using the same process as E1 but with floorplanning that optimizes only the total wire length. E3 is the wire energy consumption when energy is optimized using the proposed integrated flow with multivoltage power supplies. It is observed from the results that, using the integrated flow, the energy consumed by wires is reduced averagely by a factor of 28.9%. This reduction is achieved by integrating the behavioral synthesis and floorplanning in the proposed technique. When the multivoltage factor is also considered (in E3), FloM achieves an average reduction of 42.6%. This reduction is achieved because, in multivoltage design, different wires carry signals with different voltages, and the voltages are optimally selected to optimize the power consumption. Note that the datapath energy is not considered in Table 6.

4.3. Floorplan-Unaware versus Floorplan-Aware Low-Power Synthesis. Experiments were also performed to compare the total energy consumption (by both datapath and wire) of floorplan-unaware low power synthesis and floorplan-aware low power synthesis. The results obtained are shown in Table 7. The column "Separate Flow" reports the wire energy, datapath energy, and total energy using a separate flow. In this flow, power-conscious synthesis is performed using the proposed iterative algorithm without floorplanning at first. Then a power-unconscious floorplan is performed to optimize the total wire length. The column "Integrated Flow" reports the wire energy, datapath energy and total energy using the proposed integrated flow in FloM. From the "Improvement" column, it is observed that an average wire energy reduction of 42.6% is obtained (same as in Table 6). This is at the cost of 7.3% increase in the datapath energy consumption. This increase is because the cost function in the separate flow is the datapath energy consumption (i.e., energy consumption by wire is not counted), while the cost function in the integrated flow is the total energy consumption. Averagely, the integrated flow achieves a reduction of 9.5% in total energy reduction compared to the separate flow. As the wire energy is to be the dominate factor, the energy reduction of the integrated flow over the separate flow would be even higher.

Table 7 also shows that the computational time will be increased dramatically for floorplan-aware low power synthesis. The reason is that although datapath synthesis in FloM is incremental, the floorplan synthesis is not. To reduce the algorithmic complexity, more sophisticated floorplan technique needs to be adopted. The authors in [11] have proposed a technique that can significantly reduce the CPU time for floorplan-aware high-level synthesis.

4.4. Comparison. In this section, FloM is compared with some published results. A fair comparison is extremely difficult to obtain because different techniques have different objectives and constraints, and results are much affected by many parameters. The objective of this comparison is to provide a general idea of the effectiveness of the proposed technique.

In Table 8, the average improvements on the multivoltage datapath energy consumption are compared. FloM can achieve a comparable results compared to those published techniques.

In Table 9, comparisons on the multivoltage datapath energy consumption are performed on some benchmarks. It can be seen that FloM obtains comparable results on all benchmarks except the EWF case. FloM does not perform well on EWF, because the EWF datapath has very limited parallelism, while FloM depends on parallelism to explore alternative solutions.

To our best knowledge, we think that FloM is the first work on the floorplan-driven multivoltage low-power high-level synthesis; a direct comparison with published results on the wire energy consumption and total energy consumption is not possible. Table 10 attempts to compare the reduction on datapath energy and wire energy achieved by [6] and FloM. Note that [6] is power and floorplan aware but does not perform any multivoltage optimization. Also note that the average reduction in Table 10 is over all the experimental benchmarks instead of just the three listed. We compare the wire energy and the datapath energy separately and not the total energy, as total energy reduction has little meaning in the comparison because different wire energy models are used. For the three benchmarks listed, FloM reduces the wire energy better than [6], but at a larger cost in the datapath energy. Averagely, FloM performs slightly better and achieves higher wire energy reduction and suffers less datapath energy increase, although the results are very close.

5. Conclusion

Physical synthesis is usually performed after high-level synthesis. There is no physical layout information available to the high-level synthesis tasks. This problem becomes more critical when interconnects dominate circuits in area, speed as well as power consumption. When no physical information can be used, high-level synthesis could generate results that are optimal at the behavioral level but inferior at the physical level. To close this gap, integration of floorplanning and high-level synthesis is attracting more attention, mainly because the integration of even lower-level physical information (placement, routing, and layout) is too computationally costly. In this paper, floorplan information is integrated in the low-power high-level synthesis targeting multivoltage design using a two-level simulated annealing algorithm. The proposed technique considers the floorplanning of resources with different voltages and the wire energy consumption in addition to the energy consumption of FUs, MUXes, registers as well as level converters. With the five efficient local moves, which lead to different schedulings and bindings and hence different floorplans, FloM achieves 42.4% energy reduction in datapath compared to the results of power-unconscious synthesis. Compared to the traditional power optimization flow, the proposed integration flow achieves 42.6% energy reduction in wires at the cost of 7.3% energy increase in datapath. This energy reduction is a considerable amount in VLSI systems where the energy consumed by wires dominates the total energy consumption. FloM is also technology independent. When the technology advances, the cell library and its corresponding energy model change, FloM can adopt the new model for the cost function. In fact, FloM should perform even better for the newer technologies where the wire power dominates datapath power even more significantly.

References

[1] L. Zhong and N. K. Jha, "Interconnect-aware low-power high-level synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 336–351, 2005.

[2] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 6, pp. 663–670, 1994.

[3] M. McFarland and T. Kowalski, "Incorporating bottom-up design into hardware synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 9, pp. 938–950, 1990.

[4] D. W. Knapp, "Fasolt: a program for feedback-driven datapath optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 6, pp. 677–695, 1992.

[5] M. Xu and F. J. Kurdahi, "Layout-driven RTL binding techniques for high-level synthesis using accurate estimators," *ACM Transactions on Design Automation of Electronic Systems*, vol. 2, no. 4, pp. 312–343, 1997.

[6] A. Stammermann, D. Helms, M. Schulte, A. Schulz, and W. Nebel, "Binding, allocation and floorplanning in low power high-level synthesis," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '03)*, pp. 544–550, November 2003.

[7] A. Davoodi and A. Srivastava, "Power-driven simultaneous resource binding and floorplanning: a probabilistic approach," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 8, pp. 934–942, 2005.

[8] P. Prabhakaran, P. Banerjee, J. Crenshaw, and M. Sarrafzadeh, "Simultaneous scheduling, binding and floorplanning for interconnect power optimization," in *Proceedings of the IEEE International Conference on VLSI Design*, pp. 423–427, 1999.

[9] D. Wong and C. Liu, "A new algorithm for floorplan design," in *Proceedings of the 23rd IEEE/ACM Design Automation Conference*, pp. 101–107, 1986.

[10] A. Davoodi, V. Khandelwal, and A. Srivastava, "Empirical models for net-length probability distribution and applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 10, pp. 1066–1075, 2004.

[11] Z. Gu, J. Wang, R. P. Dick, and H. Zhou, "Unified incremental physical-level and high-level synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 9, pp. 1576–1588, 2007.

[12] X. Xing and C. C. Jong, "A look-ahead synthesis technique with backtracking for switching activity reduction in low power high-level synthesis," *Microelectronics Journal*, vol. 38, no. 4-5, pp. 595–605, 2007.

[13] M. C. Johnson and K. Roy, "Datapath scheduling with multiple supply voltages and level converters," *ACM Transactions on Design Automation of Electronic Systems*, vol. 2, no. 3, pp. 227–248, 1997.

[14] J.-M. Chang and M. Pedram, "Energy minimization using multiple supply voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, no. 4, pp. 436–443, 1997.

[15] A. Manzak and C. Chakrabarti, "A low power scheduling scheme with resources operating at multiple voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 1, pp. 6–14, 2002.

[16] S. P. Mohanty and N. Ranganathan, "Simultaneous peak and average power minimization during datapath scheduling," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 6, pp. 1157–1165, 2005.

[17] S. H. Gerez, *Algorithms for VLSI Design Automation*, John Wiley & Sons, New York, NY, USA, 1999.

[18] P.-N. Guo, T. Takahashi, C.-K. Cheng, and T. Yoshimura, "Floorplanning using a tree representation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 2, pp. 281–289, 2001.

[19] X. Tang, R. Tian, and M. D. F. Wong, "Minimizing wire length in floorplanning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 9, pp. 1744–1753, 2006.

[20] K. Usami, M. Igarashi, F. Minami, et al., "Automated low-power technique exploiting multiple supply voltages applied to a media processor," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 463–471, 1998.

[21] R. Otten, "Layout structures," in *Proceedings of the IEEE International Large Scale Systems Symposium*, Virginia Beach, Va, USA, 1982.

[22] S. Mohanty and N. Ranganathan, "Energy efficient scheduling for datapath synthesis," in *Proceedings of the 16th International Conference on VLSI Design*, p. 446, January 2003.

[23] S. P. Mohanty and N. Ranganathan, "A framework for energy and transient power reduction during behavioral synthesis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 6, pp. 562–572, 2004.