Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2012, Article ID 105250, 14 pages
doi:10.1155/2012/105250

Research Article

Parametric Yield-Driven Resource Binding in High-Level Synthesis with Multi-V_{th}/V_{dd} Library and Device Sizing

Yibo Chen,[1] Yu Wang,[2] Yuan Xie,[1] and Andres Takach[3]

[1] *Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA*
[2] *Department of Electronics Engineering, Tsinghua University, Beijing 100084, China*
[3] *Design Creation and Synthesis, Mentor Graphics Corporation, Wilsonville, OR 97070, USA*

Correspondence should be addressed to Yibo Chen, yxc236@cse.psu.edu

Received 3 August 2011; Revised 4 January 2012; Accepted 15 January 2012

Academic Editor: Zhiru Zhang

The ever-increasing chip power dissipation in SoCs has imposed great challenges on today's circuit design. It has been shown that multiple threshold and supply voltages assignment (multi-V_{th}/V_{dd}) is an effective way to reduce power dissipation. However, most of the prior multi-V_{th}/V_{dd} optimizations are performed under deterministic conditions. With the increasing process variability that has significant impact on both the power dissipation and performance of circuit designs, it is necessary to employ statistical approaches in analysis and optimizations for low power. This paper studies the impact of process variations on the multi-V_{th}/V_{dd} technique at the behavioral synthesis level. A multi-V_{th}/V_{dd} resource library is characterized for delay and power variations at different voltage combinations. Meanwhile, device sizing is performed on the resources in the library to mitigate the impact of variation, and to enlarge the design space for better quality of the design choice. A parametric yield-driven resource binding algorithm is then proposed, which uses the characterized power and delay distributions and efficiently maximizes power yield under a timing yield constraint. During the resource binding process, voltage level converters are inserted between resources when required. Experimental results show that significant power reduction can be achieved with the proposed variation-aware framework, compared with traditional worstcase based deterministic approaches.

1. Introduction

Integrating billions of transistors on a single chip with nanoscale transistors has resulted in great challenges for chip designers. One of these challenges is that the pace of productivity gains has not kept up to address the increases in design complexity. Consequently, we have seen a recent trend of moving design abstraction to a higher level, with an emphasis on *electronic system level (ESL)* design methodologies. A very important component of ESL is raising the level of abstraction of hardware design. High-level synthesis (HLS) provides this component by providing automation to generate optimized hardware from a high-level description of the function or algorithm to be implemented in hardware. HLS generates a cycle-accurate specification at the register-transfer level (RTL) that is then used in existing ASIC or FPGA design methodologies. Commercial high-level synthesis tools [1] have recently gained a lot of attention as evidenced in recent conference HLS workshops (DATE2008, DAC2008, and ASPDAC2009), conference panels, and publications that track the industry. While high-level synthesis is able to quickly generate implementations of circuits, it is not intended to replace the existing low-level synthesis. The major benefit coming from high-level synthesis is the high design efficiency, the ability to perform fast prototyping, functional verification, and early-stage design space exploration, which in turn provide guidance on succeeding low-level design steps and help produce high-quality circuits.

Power consumption and process variability are among other critical design challenges as technology scales. While it is believed that tackling these issues at a higher level of the design hierarchy can lead to better design decisions, a lot of work has been done on low-power-high-level synthesis [2–4] as well as process-variation-aware-high-level synthesis [5–8]. These techniques have been successfully implemented but most of the existing work focuses on one side of the issues

in isolation. Recently, Srivastava et al. [9] explore the multi-$V_{th}/V_{dd}/T_{ox}$ design space with the consideration of process variations at the gate level. Nevertheless, variation-aware-low power exploration for behavioral synthesis is still in its infancy.

Multiple threshold and supply voltages assignment (multi-V_{th}/V_{dd}) has been shown as an effective way to reduce circuit power dissipation [2, 3, 10, 11]. Existing approaches assign circuit components on critical paths to operate at a higher V_{dd} or lower V_{th}, and noncritical portions of the circuit are made to operate at lower V_{dd} or higher V_{th}, respectively. The total power consumption is thus reduced without degrading circuit performance. However, nowadays circuit performance is affected by process variations. If the variations are underestimated, for example, using nominal delays of circuit components to guide the design, non-critical components may turn to critical ones due to the variations, and circuit timing constraints may be violated. On the other hand, in existing corner-based worst-case analysis, variations are overestimated resulting in design specs that are hard to meet, and this consequently increases design effort and degrades circuit performance.

Device sizing is a well-studied technique for performance and power tuning at gate- or circuit-level [12]. To improve performance, upsizing of a high-V_{th} transistor, which increases switching power and die area, can be traded off against using a low-V_{th} transistor, which increases leakage power. Therefore, combining multi-V_{th} assignment and device sizing as integrated problem, can increase the design flexibility and further improve the design quality. Meanwhile, in terms of mitigating process variations, it is possible that increasing the size of transistors can reduce the randomness of the device parameters through averaging.

This paper presents a variation-aware power optimization framework in high-level synthesis using simultaneous multi-V_{th}/V_{dd} assignment and device sizing. Firstly, the impact of parameter variations on the delay and power of circuit components is explored at different operating points of threshold and supply voltages. Device sizing is then performed to mitigate the impact of variations and to enlarge the design space for better quality of the design choice. A variation-characterized resource library containing the parameters of delay and power distributions at different voltage "corners" and different device sizes, is built once for the given technology, so that it is available for high-level synthesis to query the delay/power characteristics of resources. The concept of parametric yield, which is defined as the probability that the design meets specified constraints such as delay or power constraints is then introduced to guide design space exploration. Statistical timing and power analysis on the data flow graph (DFG) is used to populate the delay and power distributions through the DFG and to estimate the overall performance and power yield of the entire design. A variation-aware resource binding algorithm is then proposed to maximize power yield under a timing yield constraint, by iteratively searching for the operations that have the maximum potential of performance/power yield improvement, and replacing them with better candidates in the multi-V_{th}/V_{dd} resource library. During the resource binding process, voltage level converters are inserted for chaining of resource units having different V_{dd} supplies.

The contribution of this paper can be summarized as follow:

(i) first, this is the first work to apply multi-V_{th}/V_{dd} techniques during high-level synthesis under the context of both delay and power variations. A flow for variation-aware power optimization in multi-V_{th}/V_{dd} HLS is proposed. This flow includes library characterization, statistical timing and power analysis methodologies for HLS, and resource binding optimization with variation-characterized multi-V_{th}/V_{dd} library;

(ii) combined multi-V_{th}/V_{dd} assignment and device sizing for high-level synthesis are performed at the granularity of function unit level, to improve the design quality and at the same time to reduce the design complexity;

(iii) voltage level conversion is explored during the resource binding in high-level synthesis, enabling the full utilization of multi-V_{dd} components for parametric yield maximization.

2. Related Work

Prior research work tightly related to this paper mainly falls into two categories: (1) gate level power minimization by simultaneous multi-V_{th} assignment and gate sizing; (2) low-power high-level synthesis using multi-V_{th} or multi-V_{dd}; (3) process variation aware high-level synthesis.

Several techniques were proposed to consider V_{th} allocation and transistor sizing as an integrated problem [13–16]. Wei et al. [14] presented simultaneous dual-V_{th} assignment and gate sizing to minimize the total power dissipation while maintaining high performance, while Karnik et al. [16] improved the simultaneous V_{th} allocation and device sizing using a Lagrangian Relaxation method. However, all of the reported techniques focus on tuning at transistor level or gate-level. While the fine granularity can yield optimal results, it also lead to high design complexity.

Shiue [2] proposed low-power scheduling schemes with multi-V_{dd} resources by maximizing the utilization of resources operating at reduced supply voltages. Khouri and Jha [3] performed high-level synthesis using a dual-V_{th} library for leakage power reduction. Tang et al. [4] formulated the synthesis problem using dual-V_{th} as a maximum weight-independent set (MWIS) problem, within which near-optimal leakage power reduction is achieved with greatly reduced run time. Very recently, Insup et al. explored optimal register allocation for high-level synthesis using dual supply voltages [17]. However, all of these techniques were applied under deterministic conditions without taking process variation into consideration.

Process variation-aware high-level synthesis has recently gained much attention. Jung and Kim [6] proposed a timing yield-aware HLS algorithm to improve resource sharing and reduce overall latency. Lucas et al. [8] integrated timing-driven floorplanning into the variation-aware high-level

design. Mohanty and Kougianos's work [18] took into account the leakage power variations in low-power high-level synthesis; however, the major difference between [18] and our work is that, the delay variation of function units was not considered in [18], so the timing analysis during synthesis was still deterministic. Recently, Wang et al. [19] proposed a joint design-time optimization and postsilicon tuning framework that tackles both timing and power variations. Adaptive body biasing (ABB) was applied to function units to reduce leakage power and improve power yield.

3. Multi-V_{th}/V_{dd} Library Characterization under Process Variations

Scheduling and resource binding are key steps during the high-level synthesis process. The scheduler is in charge of determining the sequencing the operations of a control/data flow graph (CDFG) in control steps and within control steps (operator chaining) while obeying control and data dependencies and cycle constraints while optimizing for area/power/performance. The binding process binds operations to hardware units in the resource library to complete the mapping from abstracted descriptions of circuits into practical designs. This section presents the characterization of the variation-aware multi-V_{th}/V_{dd} resource library, including the delay and power characterization flow and the selection of dual threshold and supply voltages.

3.1. Variation-Aware Library Characterization Flow. In order to facilitate the design space exploration while considering process variations, the resource library of functional units for HLS has to be characterized for delay/power variations. As shown in Figure 1, under the influence of process variations, the delay and power of each component are no longer fixed values, but represented by probability density functions (PDFs). Consequently, the characterization of function units with delay and power variations requires statistical analysis methodologies.

Process variations come from a set of sources, including random doping fluctuation (RDF) [20] and geometric variations of the gate (primarily on channel length) [21]. Since both RDF and channel length variations manifest themselves as fluctuations on the effective threshold voltage of the transistor [22], their effects can be expressed by the variations of V_{th}. Since this work focuses on demonstrating the effectiveness of variation-aware synthesis, rather than a comprehensive modeling of all variation effects, we try to focus on V_{th} variations with a simplified assumption of normal distribution of V_{th} variations, rather than covering all physical-level variation factors with different distributions. The magnitude of V_{th} variations in real circuits can be obtained via on-chip sensing and measurement. In this work, we use NCSU FreePDK 45 nm technology library [23] for all the characterization and experiments. We set the standard deviation σ of V_{th} to be 50 mV, which is projected from the silicon measurement data in [24].

We then use a commercial gate-level statistical timing analysis tool, Synopsys PrimeTime *VX* to perform the

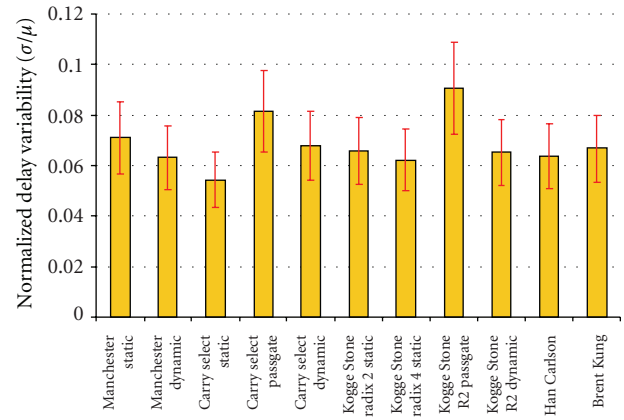


Figure 1: The delay variation for 16-bit adders in IBM Cu-08 technology (courtesy of IBM).

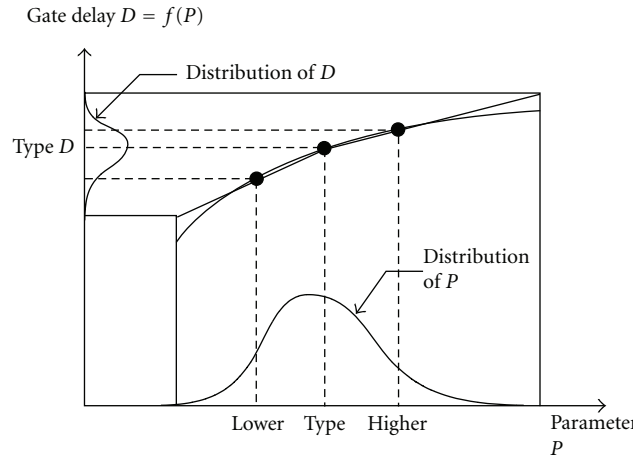


Figure 2: Calculating the delay distribution as a function of parameter P.

characterization. This variation-aware tool increases the accuracy of timing analysis by considering the statistical distribution of device parameters such as threshold voltage and gate oxide thickness. Given the distribution of a device parameter P, PrimeTime *VX* calculates the distribution of gate delay continuously throughout the range of values, using linear interpolation and extrapolation at the library-defined functional operating points, as shown in Figure 2. Validation against SPICE Monte Carlo statistical analysis [25] shows that PrimeTime *VX* analysis holds the similar accuracy but significantly reduces the running time.

The characterization flow takes as input the statistical distributions of process parameters (V_{th} in this work) and generates the statistical distributions of delay and power for each resource in the library. To characterize the delay of function units under the impact of process variations, the following steps are performed:

(1) all the standard cells in a technology library are characterized using variation-aware analysis, and the

results including parameters of cell delay distributions are collected to build a variation-aware technology library;

(2) the function units used in HLS are then synthesized and linked to the variation-aware technology library;

(3) statistical timing analysis for the function units is performed using PrimeTime *VX*, and the parameters of delay distributions are reported.

Statistical power characterization for function units in the resource library can be done using Monte Carlo analysis in SPICE. The power consumption of function units consists of dynamic and leakage components. While dynamic power is relatively immune to process variation, leakage power is greatly affected and becomes dominant as technology continues scaling down [26]. Therefore, in this paper only leakage power is characterized using statistical analysis. However, this do not mean that considerations for dynamic power can be omitted. In fact, dynamic power optimization in high-level synthesis has been a well-explored topic [1]. Our variation-oriented work emphasizing leakage power optimization can be stacked on or integrated into existing power optimization approaches in high-level synthesis, to further reduce the total power consumption of circuits.

According to Berkeley short-channel BSIM4 model [27], higher threshold voltages can lead to exponential reduction in leakage power, which is given by:

$$I_{\text{leakage}} = I_0 \exp\left(\frac{V_{\text{gs}} - V_{\text{th}}}{nV_t}\right)\left(1 - \exp\left(\frac{-V_{\text{ds}}}{V_t}\right)\right), \tag{1}$$

$$I_0 = \mu_0 C_{\text{ox}} \frac{W}{L}(n-1)V_{\text{th}}^2, \tag{2}$$

where I_{leakage} is the gate leakage current, V_{gs} and V_{ds} are the gate voltages, V_t is the thermal voltage, and n is the subthreshold swing factor. Since we assume that the device parameter V_{th} follows normal distribution, I_{leakage} follow log-normal distribution. Therefore, the leakage power of a function units is the sum of a set of log-normal distributions, which describe the leakage power of each library cell. According to [28], the sum of several log-normal random variables can be approximated by another log-normal random variable, as shown in (3):

$$P_{\text{FU}} = P_1 + P_2 + \cdots + P_n = k_1 e^{V_1} + k_2 e^{V_2} + \cdots + k_n e^{V_n}, \tag{3}$$

where P_{FU} describes the leakage power distribution of the function unit; while P_n, k_n, and V_n are the corresponding variables for library cells that build up the function unit. The mean and deviation of P_{FU} can be estimated via iterative moment matching out of the leakage power distributions of library cells [28].

The power characterization flow is stated as follows. Process variations are set in the MOS model files, and 1000 runs of Monte Carlo iterations are performed for each library cell. After the characterization, the parameters of the leakage power distributions of library cells are extracted.

Note that in our work we only characterize subthreshold leakage, since it starts dominant for technology nodes of 45 nm and below. The gate leakage can also be characterized with similar methods.

3.2. Multi-V_{th}/V_{dd} Library Characterization. Previous implementations using multiple threshold and supply voltages in conjunction have shown a very effective reduction in both dynamic and leakage power [11]. Therefore, our approach considers the combination of dual threshold and dual supply voltages, and characterizations are performed at the four "corners" of voltage settings, namely $(V_{\text{th}}^L, V_{\text{dd}}^H)$, $(V_{\text{th}}^H, V_{\text{dd}}^H)$, $(V_{\text{th}}^L, V_{\text{dd}}^L)$, and $(V_{\text{th}}^H, V_{\text{dd}}^L)$, where $(V_{\text{th}}^L, V_{\text{dd}}^H)$ is the nominal case and the other three are low-power settings. Note that although only 4 voltage settings are discussed in this paper, it is natural to extend the approach presented here to deal with more voltage settings. To reduce the process technology cost, in this paper, the multi-V_{th}/V_{dd} techniques are applied at the granularity of function units. That means, all the gates inside a function unit operate at the same threshold and supply voltages. Voltages only differ from function units to function units.

The selection of appropriate values of threshold and supply voltages for power minimization has been discussed under deterministic conditions [11]. Rules of thumb are derived for the second V_{dd} and V_{th} as functions of the original voltages [11]:

$$\begin{aligned} V_{\text{dd}}^L &= 0.43V_{\text{dd}}^H + 0.82V_{\text{th}}^H + \frac{0.72}{K} - \frac{0.55}{K^2} - 0.2, \\ V_{\text{th}}^L &= -0.024V_{\text{dd}}^H + 1.14V_{\text{th}}^H + \frac{0.72}{K} - \frac{0.49}{K^2} - 0.18, \end{aligned} \tag{4}$$

where K stands for the initial ratio between dynamic and static power. While the empirical models in [11] are validated on actual circuit benchmarks [29], they may not be accurate under the impact of process variations. A refined model taking into account the process variations is presented in [9]. As shown in Figure 3, the total power reduction with variation awareness is plotted under different combinations of $V\text{th}_2$ (V_{th}^H) and $V\text{dd}_2$ (V_{dd}^L), and this guides the optimal value selection in this work.

The characterization results (which will be further discussed in Section 6) show that, power reduction is always achieved at the cost of delay penalties. Moreover, larger delay variations are observed for slower units operating at high-V_{th} or low-V_{dd}, which means larger probability of timing violations when they are placed on the near-critical paths. This further demonstrates the necessity of statistical analysis and parametric yield-driven optimization approaches.

3.3. Device Sizing for the Resource Library. Conventionally, device sizing is an effective technique to optimize CMOS circuits for dynamic power dissipation and performance. In this work, we show that device sizing may also be utilized to mitigate the impact of process variations. As previously mentioned, the sources of process variations mainly consists of random doping fluctuation (RDF) [20] and geometric variations (GVs). GV affect the real V_{th} through the drain

induced barrier lowering (DIBL) effect. Originally, both RDF and GV have almost the equal importance in determining the V_{th} variance. As we propose to use low-V_{dd} and high-V_{th} resource units in the design, the difference between supply voltage and threshold voltage diminishes, and this reduces DIBL effect. As a result, the uncertainty in V_{th} arising from GV rapidly falls as V_{dd}. On the other hand, the RDF-induced V_{th} variation is independent of V_{dd} changes and solely a function of channel area [30]. Therefore, V_{th} variation resulting from RDF becomes dominating as V_{dd} approaches V_{th}.

Due to the independent nature of RDF variations, it is possible to reduce their impact on circuit performance through averaging. Therefore, upsizing the device can be an effective way for variability mitigation with enlarged channel area. According to [31], V_{th} variance $\sigma_{V_{th}}$ resulting from RDF is roughly proportional to $(WL)^{-1/2}$, which means we can either increase the transistor width or channel length or both. Conventional sizing approaches focus on tuning the transistor width for performance. In terms of process variability mitigation, the measurement data of V_{th} variation for 4 different device sizes is plotted in Figure 4, which shows that increasing transistor width is a more effective way to reduce the V_{th} variance [24]. Although larger transistor width means larger leakage power, the fluctuations on leakage power are reduced, and the design space for resource binding is significantly enlarged, thus using resources with larger size in the design may still be able to improve the parametric yield.

In this work, we upsize all the function units in the resource library to generate alternatives for power tuning and variability mitigation. The sizing is performed on all the gates with two different settings: the basic size (1W1L) and the double-width size (2W1L). We then perform the variation characterization for the upsized function units under all the four voltage "corners" presented in the previous section. The characterization results will be presented in Section 6.

4. Yield Analysis in Statistical High-Level Synthesis

In this section, a parametric yield analysis framework for statistical HLS is presented. We first show the necessity of statistical analysis by a simple motivational example and then demonstrate the statistical timing and power analysis for HLS as well as the modeling and integration of level converters for multi-V_{dd} HLS.

4.1. Parametric Yield. To bring the process-variation awareness to the high-level synthesis flow, we first introduce a new metric called parametric yield. The parametric yield is defined as the probability of the synthesized hardware meeting a specified constraint Yield $= P(Y \leq Y_{max})$, where Y can be delay or power.

Figure 5 shows a motivational example of yield-aware analysis. Three resource units $R1$, $R2$, and $R3$ have the same circuit implementation but operate at different supply or threshold voltages. Figure 5 shows the delay and power distributions for $R1$, $R2$, and $R3$. In this case the mean power follows up $\mu_P(R3) < \mu_P(R2) < \mu_P(R1)$, and the mean delay follows $\mu_D(R1) < \mu_D(R2) < \mu_D(R3)$, which is as expected since power reduction usually comes at the cost of increased delay. The clock cycle time T_{CLK} and the power consumption constraint P_{LMT} (e.g., the TDP (thermal design power) of most modern microprocessors) are also shown in the figure. If the variation is disregarded and nominal-case analysis is used, any of the resource units can be chosen since they all meet timing. In this case, $R3$ would be chosen as it has the lowest power consumption. However, under a statistical point of view, $R3$ has a low timing yield (approximately 50%) and is very likely to cause timing violations. In contrast, with corner-based worst-case analysis only $R1$ can be chosen under the clock cycle time constraint (the worst-case delay of

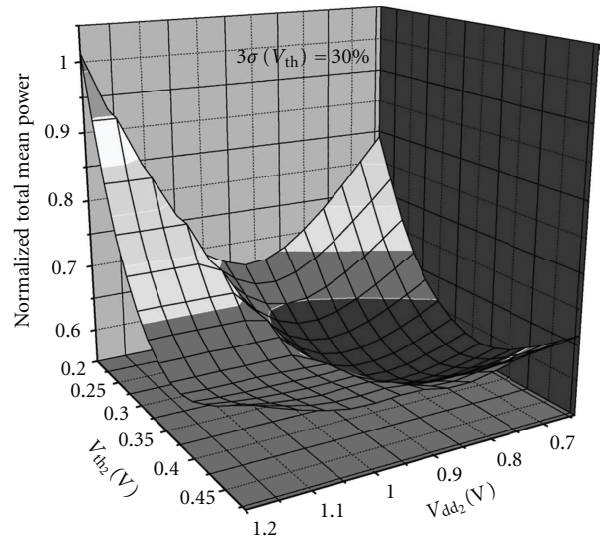


Figure 3: Optimal selection of dual threshold and supply voltages under process variation.

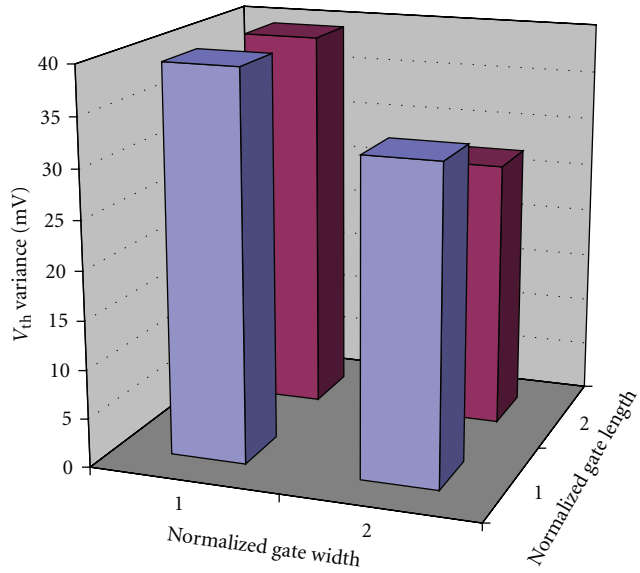


Figure 4: V_{th} variations with difference device sizes (normalized to minimal size).

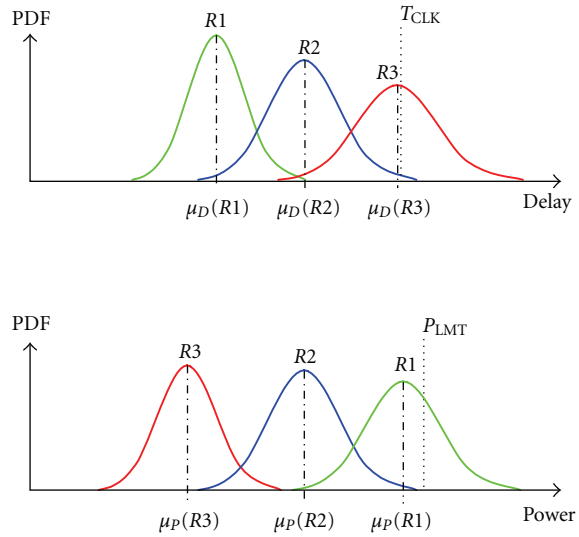


Figure 5: Motivating example of yield-driven synthesis.

$R2$ slightly violates the limit), whereas $R1$ has a poor power yield. In fact, if we set a timing yield constraint instead of enforcing the worst-case delay limitation, $R2$ can be chosen with a slight timing yield loss but a well-balanced delay and power tradeoff. Therefore, a yield-driven statistical approach is needed for exploring the design space to maximize one parametric yield under other parametric yield constraints.

4.2. Statistical Timing and Power Analysis for HLS. High-level synthesis (HLS) is the process of transforming a behavioral description into register level structure description. Operations such as additions and multiplications in the DFG are scheduled into control steps. During the resource allocation and binding stages, operations are bound to corresponding function units in the resource library meeting type and latency requirements.

Given the clock cycle time T_{CLK}, the timing yield of the entire DFG, Yield_T is defined as

$$\text{Yield}_T = P(T_1 \le T_{CLK}, T_2 \le T_{CLK}, \ldots, T_n \le T_{CLK}), \tag{5}$$

where $P()$ is the probability function, $T_1, T_2, \ldots, T_n$ are the arrival time distributions at control step $1, 2, \ldots, n$, respectively.

The arriving time distribution of each clock cycle can be computed from the delay distributions of function units bound at that cycle. Two operations, sum and max, are used to compute the distributions:

(i) sum operation is used when two function units are chained in cascade within a clock cycle, as shown in CC1 and CC2 of Figure 6. The total delay can be computed as the "sum" of their delay distributions (normal distribution assumed);

(ii) max operation is used when the outputs of two or more units are fed to another function unit at the same clock cycle, as shown in CC1 of Figure 6. The "maximum" delay distribution can be computed out of the contributing distributions using tightness probability and moment matching [19].

With these two operations, the arriving time distribution of each clock cycle is computed, and the overall timing yield of the DFG is obtained using (5).

The total power consumption of a DFG can be computed as the sum of the power consumptions of all the function units used in the DFG. Given a power limitation P_{LMT}, the power yield of the DFG Yield_P is computed as the probability that total power P_{DFG} is less than the requirement, as expressed in (6):

$$\text{Yield}_P = P(P_{DFG} \le P_{LMT}). \tag{6}$$

Since dynamic power is relatively immune to process variations, it is regarded as a constant portion which only affects the mean value of the total power consumption. Therefore, the total power is still normally distributed, although statistical analysis is only applied to the leakage power. As aforementioned in Section 3, our proposed yield-driven statistical framework can be stacked on existing approaches for dynamic power optimization, to further reduce the total power consumption of circuits.

4.3. Voltage Level Conversion in HLS. In designs using multi-V_{dd} resource units, voltage level convertors are required when a low-voltage resource unit is driving a high-voltage resource unit. Level conversion can be performed either synchronously or asynchronously. Synchronous level conversion is usually embedded in flip-flops and occurs at the active clock edge, while asynchronous level converters can be inserted anywhere within the combinational logic block.

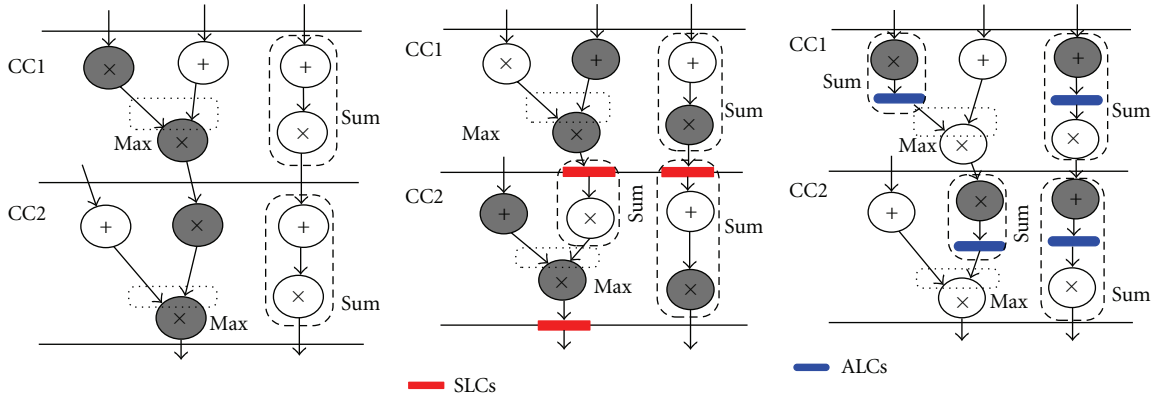
When process variations are considered, asynchronous level converters are even more favorable, because they are not bounded by clock edges, and timing slacks can be passed through the converters. Therefore, time borrowing can happen between low-voltage and high-voltage resource units. As slow function units (due to variations) may get more time to finish execution, the timing yield can be improved, and the impact of process variations is consequently reduced.

While many fast and low-power level conversion circuits have been proposed recently, this paper uses the multi-V_{th} level converter presented in [32], taking the advantage that there is no extra process technology overhead for multi-V_{th} level converters, since multi-V_{th} is already deployed for function units. The proposed level converter is composed of two dual V_{th} cascaded inverters. Its delay and power are then characterized in HSPICE using the listed parameters [32].

The delay penalty of a level converter can be accounted by summing its delay with the delay of the function unit it is associated to. The power penalty can be addressed by counting the level converters used in the DFG and adding the corresponding power to the total power consumption.

5. Yield-Driven Power Optimization Algorithm

In this section, we propose our yield-driven power optimization framework based on the aforementioned statistical



(a) No level conversion (b) synchronous conversion (c) synchronous conversion

FIGURE 6: Timing yield computation in multi-V_{dd} high-level synthesis with different level conversions. Shadowed operations are bound to function units with low-supply voltages, and the bars indicate the insertion of level converters.

timing and power yield analysis. During the high-level synthesis design loop, resource binding selects the optimal resource instances in the resource library and binds them to the scheduled operations at each control step. A variation-aware resource binding algorithm is then proposed to maximize power yield under a preset timing yield constraint, by iteratively searching for the operations with the maximum potential of timing/power yield improvement, and replacing them with better candidates in the multi-V_{th}/V_{dd} resource library.

5.1. Variation-Aware Resource Binding Algorithm Overview. Our variation-aware resource binding algorithm takes a search strategy called variable depth search [19, 33, 34] to iteratively improve the power yield under performance constraints. The outline of the algorithm is shown in Algorithm 1, where a DFG is initially scheduled and bound to resource library with nominal voltages (V_{th}^{L}, V_{dd}^{H}). A lower-bound constraint on the timing yield is set, so that the probability of the design can operate at a given clock frequency, will be larger than or equal to a preset threshold (e.g., 95%). In the algorithm, a move is defined as a local and incremental change on the resource bindings. As shown in the sub routine GENMOVE in Algorithm 1, the algorithm generates a set of moves and finds out a sequence of moves that maximizes the accumulated *gain*, which is defined as $\alpha * \Delta\text{Yield}_D + \Delta\text{Yield}_P$, where α is a weighting factor to balance the weights of timing and power yield improvements. The optimal sequence of moves is then applied to the DFG, and the timing and power yields of the DFG are updated before the next iteration. The iterative search ends when there is no yield improvement or the timing yield constraint is violated.

Note that our worst-case resource binding algorithm uses the same search strategy (variable depth search) [19, 33, 34] as the variation-aware resource binding algorithm. The key difference is that, instead of iteratively improving the power yield under performance constraints, the worst-case resource binding algorithm iteratively reduces the power consumption under specified performance constraints, where both the power consumption calculation and performance constraints are specified as deterministic numbers, rather than using the concept of power yield and performance yield.

5.2. Voltage Level Conversion Strategies. Moves during the iterative search may result in low-voltage resource units driving high-voltage resource units. Therefore, level conversion is needed during resource binding. However, if resources are selected and bound so that low-voltage resource units never drive high-voltage ones, level conversion will not be necessary, and the delay and power overheads brought by level converters can be avoided. This reduces the flexibility of resource binding for multivoltage module combinations, and may consequently decrease the attainable yield improvement. The tradeoff in this conversion-avoidance strategy, can be explored and evaluated within our proposed power optimization algorithm.

We also incorporate other two strategies of level conversions in the power optimization algorithm for comparison. All the three strategies are listed as follows:

(i) level conversion avoidance: resource binding is performed with the objective that low-voltage resources never drive high-voltage ones. As shown in Figure 6(a), no dark-to-light transition between operations is allowed (while dark operations are bound to low-V_{dd} units), so that level conversion is avoided. This is the most conservative strategy;

(ii) synchronous level conversion: voltage level conversion is done synchronously in the level-converting flip-flops (SLCs). As shown in Figure 6(b), the dark-to-light transition only happens at the beginning of each clock cycles. The flip-flop structure proposed in [35] is claimed to have smaller delay than the combination of an asynchronous converter and a conventional flip-flop. However, as discussed previously, synchronous level conversion may reduce the flexibility of resource binding as well as the possibility

```
VABinding(DFG, ResLib, Constraints, LCStrategy)
   ▷ Initialization
(1)   Scheduling using ASAP strategy
(2)   Initial Binding to (V_th^L, V_dd^H) resources
   ▷ Variation-aware resource binding
(3)   while ΔYield_P > 0 AND Yield_D ≥ Constraint
(4)      do for i ← 1 to MAXMOVES
(5)            do Gain_i ←GenMove(DFG, ResLib, LCStrategy)
(6)               Append Gain_i to Gain_List;
(7)         Find subsequence Gain_1, ..., Gain_k in Gain_List
            so that G = Σ_{i=1}^{k} Gain_i is maximized
(8)         if G > 0
(9)            do Accept moves 1 · · · k
(10)                Evaluate ΔYield_P and Yield_D
GenMove(DEG, ResLib, LCStrategy)
(1)   MOVE: Choose a move using steepest descent heuristic [33]
(2)   Check whether and where level conversion is needed
(3)   if LC Strategy = Avoidance AND NeedConversion
(4)      do goto MOVE
(5)   if LC Strategy = Synchronous AND NeedConversion
   ▷Check whether conversion is synchronous or not
(6)      do if  Conversion is inside operation chaining
(7)                  do goto MOVE
(8)   Count the overhead of level conversion
(9)   Evaluate the Gain of this move
(10)  Return Gain
```

Algorithm 1: Outline of the variation-aware resource binding algorithm.

of timing borrowing. The effectiveness of this strategy is to be explored by the optimization algorithm;

(iii) asynchronous level conversionL: asynchronous level converters (ALCs) are inserted wherever level conversion is needed, as dark-to-light transition can happen anywhere in Figure 6. This aggressive strategy provides the maximum flexibility for resource binding and timing borrowing. Although it brings in delay and power overhead, it still has great potential for timing yield improvement.

5.3. Moves Used in the Iterative Search. In order to fully explore the design space, three types of moves are used in the iterative search for resource binding;

(i) resource rebinding: in this move, an operation is assigned to a different function unit in the library with different timing and power characteristics. The key benefit of the multi-V_{th}/V_{dd} techniques is that it provides an enlarged design space for exploration, and optimal improvements are more likely to be obtained;

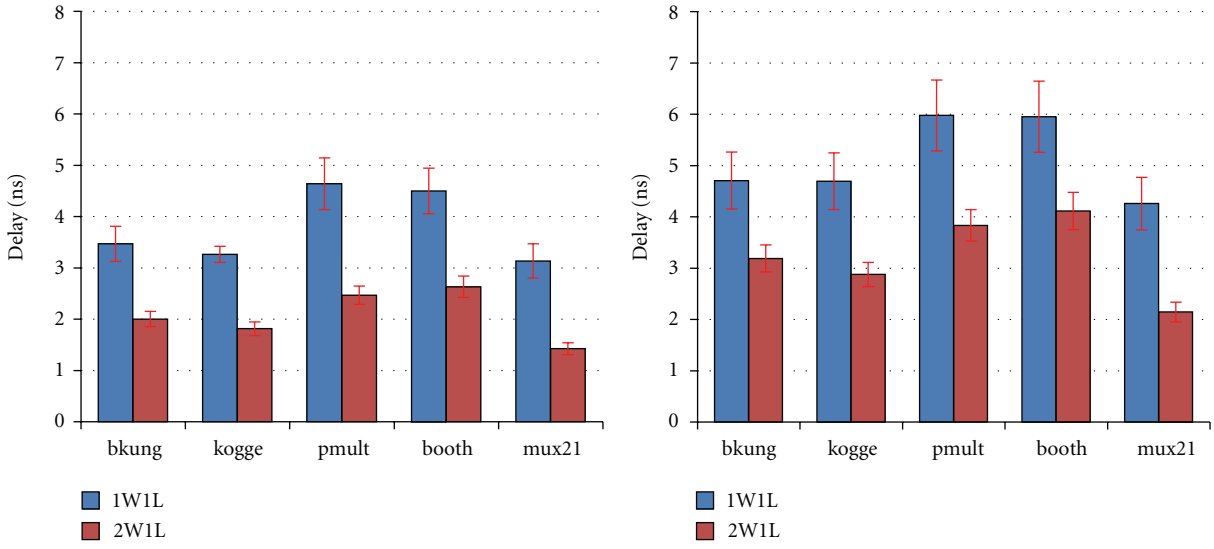
(ii) resource sharing: in this move, two function units that are originally bound to different function units, are now merged to share the same function unit. The type of move reduces the resource usage and consequently improves the power yield;

(iii) resource splitting: in this move, the operation that originally shared function unit with other operations, is split from the shared function unit. This type of move might lead to other moves such as resource rebinding and resource sharing.

After each move, the algorithm checks where the low-supply voltage function units are used and decides whether to insert or remove the level converters, according to the predefined level conversion strategy. If a move is against the strategy, it is revoked, and new moves are generated until a qualifying move is found.

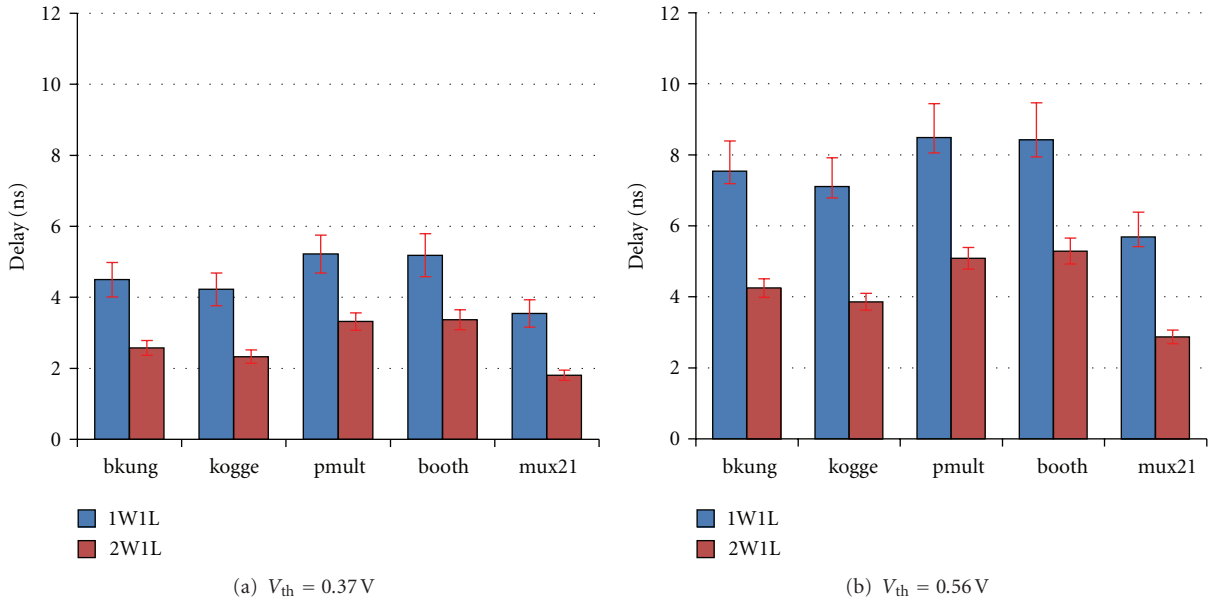
5.4. Algorithm Analysis. It has to be noted that, in the procedure GenMove shown in Algorithm 1, even though the returned Gain might be negative, it still could be accepted. Since the sequence of a cumulative positive gain is considered, the negative gains help the algorithm escape from local minima through hill climbing.

As for the computational complexity, it is generally not possible to give nontrivial upper bounds of run time for local search algorithms [33]. However, for variable depth search in general graph partitioning, Aarts and Lenstra [33] found a near-optimal growth rate of run time to be $O(n \log n)$, where n is the number of nodes in the graph. In our proposed algorithm, the timing and power yield evaluation, as well as the level converter insertion, are performed at each move. Since the yield can be updated using a gradient computation approach [19], the run time for each move is at most $O(n)$. Therefore, the overall run time for the proposed resource binding algorithm is $O(n^2 \log n)$.



(a) $V_{th} = 0.37\,V$ (b) $V_{th} = 0.56\,V$

Figure 7: Delay characterization of function units with multi-V_{th}/V_{dd} and variation awareness.



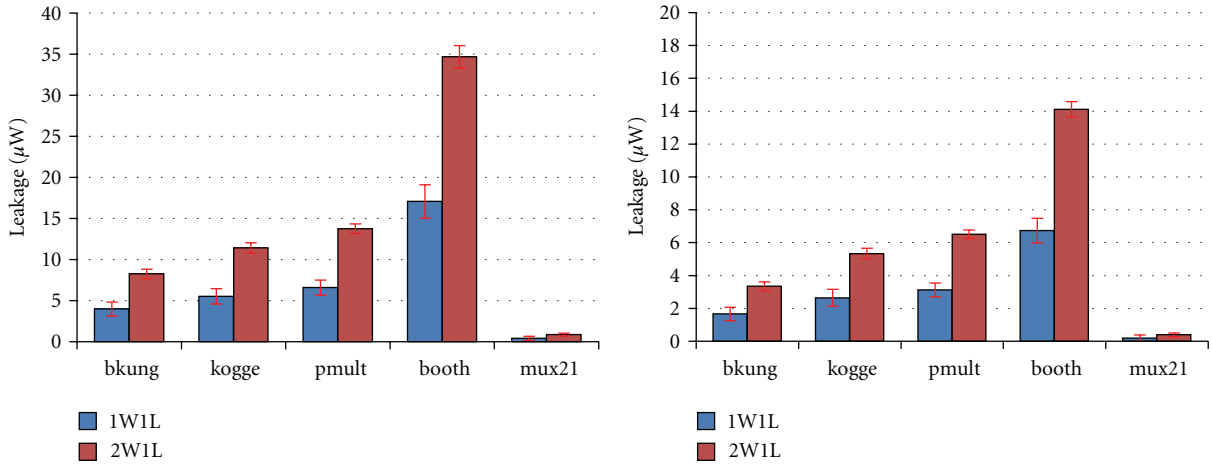
(a) $V_{th} = 0.37\,V$ (b) $V_{th} = 0.56\,V$

Figure 8: Delay characterization of function units with multi-V_{th}/V_{dd} and variation awareness.

6. Experimental Results

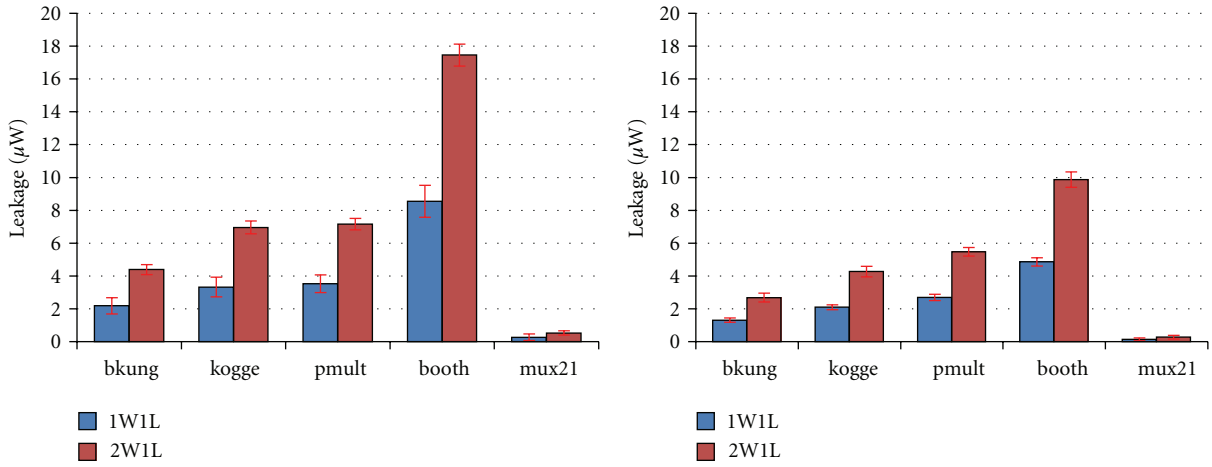
In this section, we present the experimental results of our variation-aware power optimization framework for high-level synthesis. The results show that our method can effectively improve the overall power yield of given designs and reduce the impact of process variations.

We first show the variation-aware delay and power characterization of function units. The characterization is based on NCSU FreePDK 45 nm technology library [23]. The voltage corners for the characterization are set as $V_{th}^{L} = 0.37\,V$, $V_{th}^{H} = 0.56\,V$, $V_{dd}^{L} = 0.9\,V$, and $V_{dd}^{H} = 1.1\,V$. The characterization results for five function units, including two 16-bit adders bkung and kogge, two 8-bit $\times$ 8-bit multipliers pmult and booth, and one 16-bit multiplexer *mux21*, are depicted in Figures 7, 8, 9, 10. In the figures, the color bars show the nominal case values, while the error bars show the deviations. It is clearly shown that with lower V_{dd} and/or higher V_{th}, significant power reductions are achieved at the cost of delay penalty. Meanwhile, up sizing the transistor can improve the circuit performance but also yield to larger power consumption. In terms of variability mitigation, both voltage scaling and device sizing have significant impact on the delay and leakage variations. We can explore this trend



(a) $V_{th} = 0.37$ V (b) $V_{th} = 0.56$ V

Figure 9: Leakage power characterization of function units with multi-V_{th}/V_{dd} and variation awareness.



(a) $V_{th} = 0.37$ V (b) $V_{th} = 0.56$ V

Figure 10: Leakage power characterization of function units with multi-V_{th}/V_{dd} and variation awareness.

further in Figures 11 and 12, where the delay and power distributions of the function unit bkung is sampled at a third V_{th} of 0.45 V. The plotted curves show that the magnitude of delay variation increases for higher V_{th} units, which means larger probabilities of timing violations if these high V_{th} units are placed on near-critical paths. The figures also show that up-sizing the device can effectively reduce the delay and leakage variations, as depicted by the error bars in Figures 11 and 12.

With the variation-aware multi-V_{th}/V_{dd} resource library characterized, our proposed resource binding algorithm is applied on a set of industrial high-level synthesis benchmarks, which are listed in Table 1. A total power limitation P_{LMT} is set for each benchmark to evaluate the power yield improvement. The dynamic power consumption of function units is estimated by Synopsys Design Compiler with multi-V_{th}/V_{dd} technology libraries generated by Liberty NCX. In this work with FreePDK 45 nm technology, the dynamic power is about 2 times the mean leakage power. The power yield before and after the improvement is then computed using (6) in Section 4.2. The proposed resource binding algorithm is implemented in C++, and experiments are conducted on a Linux workstation with Intel Xeon 3.2 GHz processor and 2 GB RAM. All the experiments run in less than 60 s of CPU time.

We compare our variation-aware resource binding algorithm against the traditional deterministic approach, which uses the worst-case ($\mu + 3\sigma$) delay values of function units in the multi-V_{th}/V_{dd} library to guide the resource binding. For deterministic approach, we leverage a commercial HLS tool called Catapult-C to obtain the delay/area/power estimation. The worst-cased based approach will naturally lead to 100% timing yield; however, the power yield is poor as shown in the motivational example in Figure 5. In contrast, our yield-aware statistical optimization algorithm takes the delay and power distributions as inputs, explores the design space

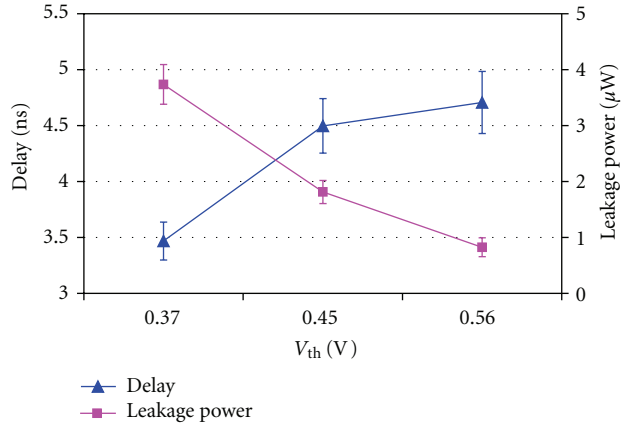


Figure 11: The delay and power tradeoff with increasing V_{th} for bkung with default device sizing (1W1L).

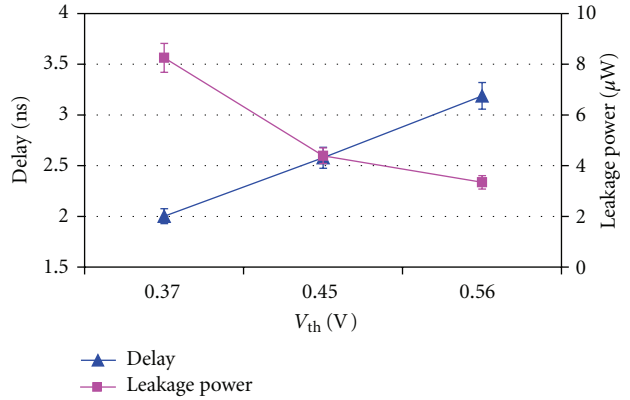


Figure 12: The delay and power tradeoff with increasing V_{th} for bkung with doubled device sizing (2W1L).

Table 1: The profile of test benchmarks.

Name	Nodes number	Edges number	Add number	Mult number
CHEM	33	51	15	18
EWF	34	49	20	12
PR	44	132	26	16
WANG	52	132	26	24
MCM	96	250	64	30
HONDA	99	212	45	52
DIR	150	312	84	64
STEAM	222	470	105	115

with the guidance of YieldGain, and iteratively improves the power yield under a slight timing yield loss. The comparison results are shown in Figures 13, 14, 15 and 16, respectively.

Figure 13 shows the power yield improvement against worst-case delay based approach, with different level conversion strategies. A fixed timing yield constraint of 95% is set for the proposed variation-aware algorithm, using the function units with default device sizes (1W1L). The overheads of the level converters used in this paper are listed in

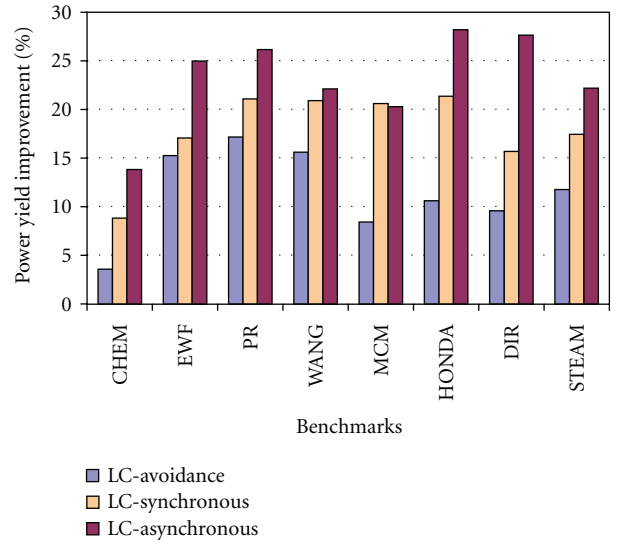


Figure 13: Power yield improvement against deterministic worst-case approach with different level conversion strategies and timing yield constraint of 95%.

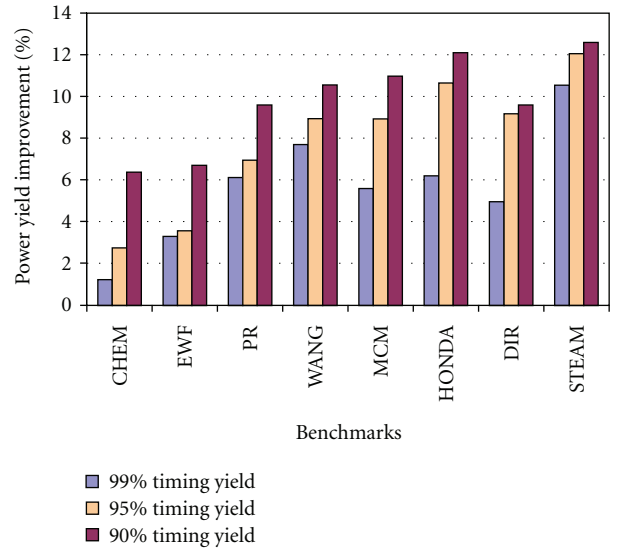


Figure 14: Power yield improvement against deterministic worst-case approach with multi-V_{th} only and different timing yield constraints.

Table 2: The Overheads of voltage level converters.

Type	Delay (ps)	Power (nW)
Synchronous converter	80	Negligible
Asynchronous converter	200	3790

Table 2. The usage of function units and level converters under the three listed conversion strategies (conversion avoidance, synchronous conversion and asynchronous conversion) is listed in Table 3, in which "Vdd-H FUs number" and "Vdd-L FUs number" show the numbers of function units with high/low supply voltages, respectively, and "LCs

Table 3: The usage of function units and level converters with different level conversion strategies.

Bench name	LC-avoidance		LC-synchronous			LC-asynchronous			
	Vdd-H	Vdd-L	Vdd-H	Vdd-L	LCs	Vdd-H	Vdd-L	LCs	Overhead
CHEM	5	1	3	3	1	2	4	2	4.2%
EWF	6	1	4	3	1	3	4	2	3.7%
PR	7	2	6	3	2	4	5	3	4.1%
WANG	9	2	8	3	2	5	6	3	3.4%
MCM	20	4	16	8	4	15	9	5	2.4%
HONDA	22	5	17	10	6	15	12	6	3.9%
DIR	28	4	20	12	8	14	18	12	4.8%
STEAM	34	8	25	19	11	21	23	13	5.0%

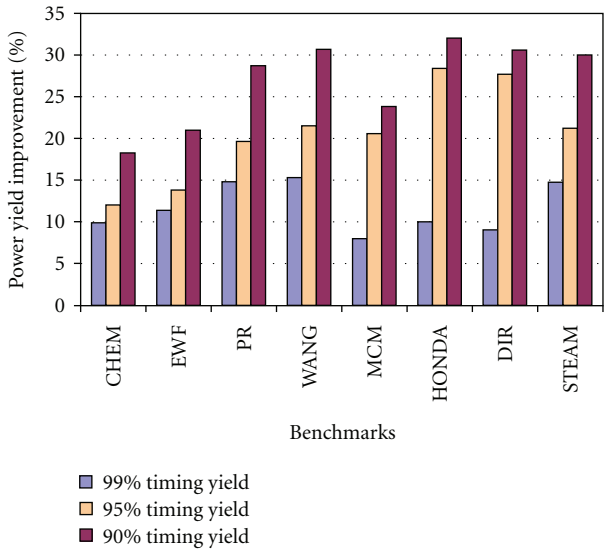


Figure 15: Power yield improvement against deterministic worst-case approach with asynchronous level conversion and different timing yield constraints.

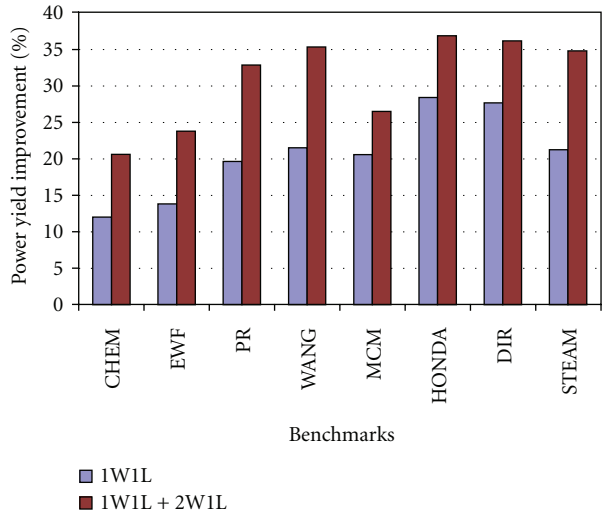


Figure 16: Power yield improvement against deterministic worst-case approach with asynchronous level conversion and different timing yield constraints.

number" counts the number of converters used in the design. The last column counts the total power overhead of the asynchronous level converters. The average power yield improvements for the three strategies are 11.7%, 17.9%, and 22.2%, respectively. From Figure 13 and Table 3 we can see that larger power yield improvements can be achieved when more low-Vdd function units are used in the design. The results also validate our claims in Section 4.3 and Section 5.2 that, asynchronous level conversion is more favorable in statistical optimization because it enables timing borrowing between function units and leads to the timing yield improvement that can compensate the overhead of the converters. Therefore, compared to the synchronous case, more asynchronous converters are used while yielding better results.

Figure 14 shows power yield improvement with multi-V_{th} technique only, which means only the resource units with nominal supply voltage V_{dd}^{H} can be selected. In this case, no level conversion is needed so there is no overhead for level converters. Only function units with default device sizes (1W1L) are used. The average power yield improvements against worst-case delay based approach, under timing yield constraints 99%, 95%, and 90% are 5.7%, 7.9%, and 9.8%, respectively. At timing yield 95%, the average power yield improvement (7.9%) is smaller than the LC-avoidance case (11.5%) in Figure 13, which shows that using multi-V_{dd} resource units can further improve the power yield.

Figure 15 shows the power yield improvement against worst-case delay based approach, under different timing yield constraints. Asynchronous level conversion is chosen in this series of experiments. Only function units with default device sizes (1W1L) are used. The average power yield improvements under timing yield constraints 99%, 95%, and 90% are 11.6%, 20.6%, and 26.9%, respectively. It is clearly shown that, the power yield improvement largely depends on how much timing yield loss is affordable for the design. This will further push forward the design space exploration for a well-balanced timing and power tradeoff.

Figure 16 shows the power yield improvement against worst-case delay-based approach, using function units with different device sizes. Asynchronous level conversion is chosen in this series of experiments, and a fixed timing yield constraint of 95% is set for the proposed variation-aware algorithm. Compared with the average 20.6% yield improvement in the case using default device size (1W1L) only, using

both default-size (1W1L) and double-size (2W1L) resources can lead to an average power yield improvement of 30.9%. Obviously, upsized device with higher performance and smaller variability provide additional flexibility for design space exploration; however, this is achieved at the cost of larger silicon area.

7. Conclusions

In this paper, we investigate the impact of process variations on multi-V_{th}/V_{dd} and device sizing techniques for low-power-high-level synthesis. We characterize delay and power variations of function units under different threshold and supply voltages, and feed the variation-characterized resource library to the HLS design loop. Statistical timing and power analysis for high-level synthesis is then introduced, to help our proposed resource binding algorithm explore the design space and maximize the power yield of designs under given timing yield constraints. Experimental results show that significant power reduction can be achieved with the proposed variation-aware framework, compared with traditional worst-case based deterministic approaches.

Acknowledgment

This work was supported in part by NSF 0643902,0903432, and 1017277, NSFC 60870001/61028006 and a grant from SRC.

References

[1] P. Coussy and A. Morawiec, *High-Level Synthesis: From Algorithm to Digital Circuit*, Springer, 2009.

[2] W. T. Shiue, "High level synthesis for peak power minimization using ILP," in *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, pp. 103–112, July 2000.

[3] K. S. Khouri and N. K. Jha, "Leakage power analysis and reduction during behavioral synthesis," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 10, no. 6, pp. 876–885, 2002.

[4] X. Tang, H. Zhou, and P. Banerje, "Leakage power optimization with dual-Vth library in high-level synthesis," in *Proceedings of the 42nd Design Automation Conference (DAC '05)*, pp. 202–207, June 2005.

[5] W. L. Hung, X. Wu, and Y. Xie, "Guaranteeing performance yield in high-level synthesis," in *Proceedings of the International Conference on Computer-Aided Design (ICCAD '06)*, pp. 303–309, November 2006.

[6] J. Jung and T. Kim, "Timing variation-aware high-level synthesis," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '07)*, pp. 424–428, November 2007.

[7] F. Wang, G. Sun, and Y. Xie, "A variation aware high level synthesis framework," in *Proceedings of the Design, Automation and Test in Europe (DATE '08)*, pp. 1063–1068, March 2008.

[8] G. Lucas, S. Cromar, and D. Chen, "FastYield: Variation-aware, layout-driven simultaneous binding and module selection for performance yield optimization," in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '09)*, pp. 61–66, January 2009.

[9] A. Srivastava, T. Kachru, and D. Sylvester, "Low-power-design space exploration considering process variation using robust optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 1, pp. 67–78, 2007.

[10] K. Usami and M. Igarashi, "Low-power design methodology and applications utilizing dual supply voltages," in *Proceedings of the Design Automation Conference (ASPDAC '00)*, pp. 123–128, Yokohama, Japan, 2000.

[11] A. Srivastava and D. Sylvester, "Minimizing total power by simultaneous Vdd/Vth assignment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 665–677, 2004.

[12] C. P. Chen, C. C. N. Chu, and D. F. Wong, "Fast and exact simultaneous gate and wire sizing by lagrangian relaxation," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '98)*, pp. 617–624, ACM, New York, NY, USA, 1998.

[13] S. Sirichotiyakul, T. Edwards, C. Oh et al., "Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing," in *Proceedings of the 1999 36th Annual Design Automation Conference (DAC '99)*, pp. 436–441, June 1999.

[14] L. Wei, K. Roy, and C. K. Koh, "Power minimization by simultaneous dual-Vth assignment and gate-sizing," in *Proceedings of the 22nd Annual Custom Integrated Circuits Conference (CICC '00)*, pp. 413–416, May 2000.

[15] P. Pant, R. K. Roy, and A. Chatterjee, "Dual-threshold voltage assignment with transistor sizing for low power CMOS circuits," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 9, no. 2, pp. 390–394, 2001.

[16] T. Karnik, Y. Ye, J. Tschanz et al., "Total power optimization by simultaneous dual-Vt allocation and device sizing in high performance microprocessors," in *Proceedings of the 39th Annual Design Automation Conference (DAC '02)*, pp. 486–491, June 2002.

[17] S. Insup, P. Seungwhun, and S. Youngsoo, "Register allocation for high-level synthesis using dual supply voltages," in *Proceedings of the 46th ACM/IEEE Design Automation Conference (DAC '09)*, pp. 937–942, July 2009.

[18] S. P. Mohanty and E. Kougianos, "Simultaneous power fluctuation and average power minimization during nano-CMOS behavioral synthesis," in *Proceedings of the 20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID '07)*, pp. 577–582, January 2007.

[19] F. Wang, X. Wu, and Y. Xie, "Variability-driven module selection with joint design time optimization and post-silicon tuning," in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '08)*, pp. 2–9, March 2008.

[20] R. W. Keyes, "Physical limits in digital electronics," *Proceedings of the IEEE*, vol. 63, no. 5, pp. 740–767, 1975.

[21] D. S. Boning and S. Nassif, "Models of process variations in device and interconnect," in *Design of High Performance Microprocessor Circuits*, IEEE Press, 2000.

[22] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "Analysis and mitigation of variability in subthreshold design," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 20–25, August 2005.

[23] NCSU, "45 nm FreePDK," http://www.eda.ncsu.edu/wiki/FreePDK.

[24] M. Meterelliyoz, A. Goel, J. P. Kulkarni, and K. Roy, "Accurate characterization of random process variations using a robust

low-voltage high-sensitivity sensor featuring replica-bias circuit," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC '10)*, pp. 186–187, February 2010.

[25] C. Jacoboni and P. Lugli, *The Monte Carlo Method for Semiconductor Device Simulation*, Springer, 1990.

[26] N. S. Kim, T. Austin, D. Blaauw et al., "Leakage current: Moore's law meets static power," *Computer*, vol. 36, no. 12, pp. 68–64, 2003.

[27] K. M. Cao, W. C. Lee, W. Liu et al., "BSIM4 gate leakage model including source-drain partition," in *Proceedings of the IEEE International Electron Devices Meeting*, pp. 815–818, December 2000.

[28] N. C. Beaulieu, A. A. Abu-Dayya, and P. J. McLane, "Comparison of methods of computing lognormal sum distributions and outages for digital wireless applications," in *Proceedings of the IEEE International Conference on Communications*, pp. 1270–1275, May 1994.

[29] S. H. Kulkarni, A. N. Srivastava, and D. Sylvester, "A new algorithm for improved VDD assignment in low power dual VDD systems," in *Proceedings of the 2004 International Symposium on Lower Power Electronics and Design (ISLPED '04)*, pp. 200–205, August 2004.

[30] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1440, 1989.

[31] J. Kwong and A. P. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," in *Proceedings of the 11th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '06)*, pp. 8–13, October 2006.

[32] S. A. Tawfik and V. Kursun, "Multi-Vth level conversion circuits for multi-VDD systems," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 1397–1400, May 2007.

[33] E. Aarts and J. K. Lenstra, *Local Search in Combinatorial Optimization*, Princeton University Press, 2003.

[34] A. Raghunathan and N. K. Jha, "Iterative improvement algorithm for low power data path synthesis," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 597–602, November 1995.

[35] F. Ishihara, F. Sheikh, and B. Nikolić, "Level conversion for dual-supply systems," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 12, no. 2, pp. 185–195, 2004.