PAPER Special Section on VLSI Design and CAD Algorithms

Low-Power Design Methodology of Voltage Over-Scalable Circuit with Critical Path Isolation and Bit-Width Scaling

Yutaka MASUDA[†a)], ***Member***, **Jun NAGAYAMA**[††], **TaiYu CHENG**[†††], ***Nonmembers***, **Tohru ISHIHARA**[†], **Yoichi MOMIYAMA**[††], ***and*** **Masanori HASHIMOTO**[†††], ***Members***

SUMMARY This work proposes a design methodology that saves the power dissipation under voltage over-scaling (VOS) operation. The key idea of the proposed design methodology is to combine critical path isolation (CPI) and bit-width scaling (BWS) under the constraint of computational quality, e.g., Peak Signal-to-Noise Ratio (PSNR) in the image processing domain. Conventional CPI inherently cannot reduce the delay of intrinsic critical paths (CPs), which may significantly restrict the power saving effect. On the other hand, the proposed methodology tries to reduce both intrinsic and non-intrinsic CPs. Therefore, our design dramatically reduces the supply voltage and power dissipation while satisfying the quality constraint. Moreover, for reducing co-design exploration space, the proposed methodology utilizes the exclusiveness of the paths targeted by CPI and BWS, where CPI aims at reducing the minimum supply voltage of non-intrinsic CP, and BWS focuses on intrinsic CPs in arithmetic units. From this key exclusiveness, the proposed design splits the simultaneous optimization problem into three sub-problems; (1) the determination of bit-width reduction, (2) the timing optimization for non-intrinsic CPs, and (3) investigating the minimum supply voltage of the BWS and CPI-applied circuit under quality constraint, for reducing power dissipation. Thanks to the problem splitting, the proposed methodology can efficiently find quality-constrained minimum-power design. Evaluation results show that CPI and BWS are highly compatible, and they significantly enhance the efficacy of VOS. In a case study of a GPGPU processor, the proposed design saves the power dissipation by 42.7% with an image processing workload and by 51.2% with a neural network inference workload.

key words: *critical path isolation, bit-width scaling, voltage over-scaling, approximate computing*

1. Introduction

Approximate computing has recently emerged as a promising approach to energy-efficient design of digital systems [1]–[3]. While the conventional systems require fully precise and completely deterministic computation, approximate computing allows some loss of quality or optimality in the computed result. This concept is suitable for a wide range of applications such as digital signal processing, image, audio, and video processing, graphics, wireless communications, and machine learning. By exploiting the inherent resilience of those applications, approximate computing techniques substantially improve energy efficiency (e.g., [1]).

As one of the approximate computing techniques aiming at low-power design, voltage over-scaling (VOS) has been widely studied [4]–[10]. VOS aggressively reduces the supply voltage and thus dramatically saves the dynamic power dissipation. Since VOS may cause timing errors in the circuit due to the supply voltage reduction, designers should carefully investigate whether these potential timing errors cause fatal system failures or not. For keeping correct operations under VOS, two types of approaches have been proposed; (1) add error-resilient mechanisms, and (2) optimize the timing design.

The first approach introduces error-tolerant mechanisms to the original circuit for recovering occurred errors (e.g., [4], [11]). Although the first approach could satisfy the constraint of computational quality thanks to the recovery mechanisms, such additional circuits may induce large area overhead, e.g., 20% in [11]. The second approach manipulates the timing design for reducing the number of critical paths (e.g., [5], [12]). Masuda *et al.* proposed a critical path isolation (CPI) method that gives timing slacks to active critical paths (CPs) and reduces the number of CPs for lowering the supply voltage [12]. They report that CPI reduced the supply voltage by 25% with 1.4% area overhead. However, we found that inherently CPI cannot reduce the delay of intrinsic CPs, which already consist of wide cells and low-Vth cells. In other words, if these intrinsic CPs affect the computational quality, CPI may not reduce the supply voltage, which severely restricts the power saving effect under VOS. To the contrary, if we could reduce intrinsic CPs, we can expect further supply voltage reduction.

This work proposes a design methodology that saves the power dissipation under VOS operation. The key idea of the proposed design methodology is to combine CPI [12] and bit-width scaling (BWS) [13]–[16]. BWS reduces the bit-width of data representation and thus reduces the delay of the target data paths. Figure 1 illustrates the expected power saving effects of the proposed design under the constraint of computation quality, e.g., Peak Signal-to-Noise Ratio (PSNR) in the image process domain. In the naive VOS without any timing optimization, the computational quality degrades sharply due to a number of non-intrinsic CPs. In Fig. 1(a), conventional CPI mitigates the quality degradation by VOS thanks to the delay reduction of the non-intrinsic CPs. On the other hand, the proposed design tries to reduce the delay of both intrinsic and non-intrinsic CPs by incorpo-

Manuscript received February 27, 2021.
Manuscript revised July 10, 2021.
Manuscript publicized August 31, 2021.

[†]The authors are with Graduate School of Informatics, Nagoya University, Nagoya-shi, 464-0814 Japan.
[††]The authors are with the Socionext Inc., Yokohama-shi, 222-0033 Japan.
[†††]The authors are with Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.
a) E-mail: masuda@ertl.jp

DOI: 10.1587/transfun.2021VLP0002

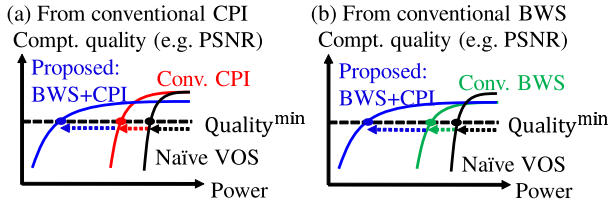

Fig. 1 Expected power savings thanks to the proposed design methodology. (a) From conventional CPI, the proposed design reduces the power dissipation thanks to BWS. (b) From conventional BWS, the proposed design achieves graceful degradation thanks to CPI.

rating BWS and CPI. Then, the proposed design is expected to achieve graceful degradation drawn in blue and attain a better trade-off regarding power dissipation under the quality constraint. Similarly, compared with conventional BWS in Fig. 1(b), the proposed design further reduces the supply voltage and power dissipation since CPI increases the timing slack of non-intrinsic CPs and reduces timing errors in them.

Here, BWS is one of the approximate computing techniques, and it sacrifices a certain amount of computation quality by degrading the precision. Thus, for minimizing the power dissipation under the quality constraint, a designer needs to carefully determine the design parameters, including the amount of bit-width reduction, the CPI methodology, and the supply voltage. On the other hand, an explicit simultaneous optimization is difficult concerning computational time since the computation quality has a non-linear relationship to the circuit structure and supply voltage, and then estimating the minimum supply voltage under VOS requires a long computational time.

For efficiently solving the co-design optimization problem, the proposed design methodology utilizes the exclusiveness of the paths targeted by CPI and BWS, where CPI aims at non-intrinsic CPs and BWS focuses on intrinsic CPs in arithmetic units. Thanks to this exclusiveness of the target paths, we can individually explore the design space of CPI and BWS, which leads us to a lightweight co-design optimization methodology. The proposed methodology solves the co-design optimization problem with three steps; (1) the determination of bit-width reduction (BWS), (2) the timing optimization for non-intrinsic CPs (CPI), and (3) investigating the minimum supply voltage of the BWS and CPI-applied circuit under quality constraint, for reducing power dissipation. Based on this approach, we can efficiently solve the problem of finding quality-constrained minimum-power design.

The main contributions of this work include (1) the design methodology using CPI and BWS toward quality-aware minimum-power VOS, and (2) quantitative evaluation of the power saving effects thanks to the proposed design under several PVTA corners. To the best of our knowledge, this is the first work that optimizes the design by incorporating CPI and BWS for overcoming the limitation of conventional CPI[†]. Moreover, for efficiently finding the low-power design under VOS, the proposed design methodology exploits the exclusiveness of target CPs between CPI and BWS to reduce co-design exploration space. Experimental results show that BWS and CPI are highly compatible and they enhance and provide significant power saving effects. More specifically, in a case study of a GPGPU processor, we demonstrate that the proposed design saves the power dissipation by 42.7% with an image processing workload and by 51.2% with a neural network inference workload.

The rest of this paper is organized as follows. Section 2 explains assumed BWS and CPI and formulates the design optimization problem. Section 3 proposes the design methodology which incorporates CPI and BWS. Section 4 demonstrates power saving effects thanks to the proposed design methodology. Lastly, concluding remarks are given in Sect. 5.

2. Assumption and Problem Formulation

The proposed design methodology for VOS consists of BWS and CPI. Section 2.1 explains the assumed BWS and CPI. Then, Sect. 2.2 formulates the design optimization problem.

2.1 Assumed BWS and CPI

First, we explain the assumed BWS. BWS reduces the bit-width of data representation and thus reduces the path delay, which contributes to not only power saving but the delay reduction of CPs. Here, the allowable bit-width reduction varies depending on the application program and quality constraint. In this work, we assume to run various workloads and thus adopt the bit-width tunable methodology as shown in Fig. 2. Also, we assume that the supply voltage and bit-width can be dynamically tuned for each chip and running workload individually to exploit the chip and workload-dependent margin for power minimization under quality constraint.

Let us explain Fig. 2. The number of bit-width reduction N_{red} is specified by a control signal. Then, the bottom N_{red} bits are replaced with a user-defined specific value, e.g., 0000, and the replaced bits are given to the arithmetic unit. In this case, paths starting from the replaced bits become false paths and do not toggle in the arithmetic unit. Hence, the dynamic power dissipation and delay of CPs could be reduced. Note that if we increase N_{red}, the larger power/delay reduction can be obtained, but loss of computational quality also enlarges. Therefore, we need to choose appropriate N_{red} from the trade-off between computational quality and the power/delay reduction. In Sect. 3.2, we will explain how to determine N_{red}.

Figure 3 shows the assumed CPI. In a conventional design, negative timing slack is cured by up-sizing and buffering, while positive timing slack is traded off for area and power reduction. As a result, many non-CPs become CPs, which are called non-intrinsic CPs. However, such a timing optimization decreases the timing margin of the paths that go through the replaced instances and may increase the timing error occurrence under VOS. On the other hand, CPI

[†]A preliminary version of this work is presented in [17]. This paper extends [17] and adds discussion in terms of power reduction thanks to BWS.

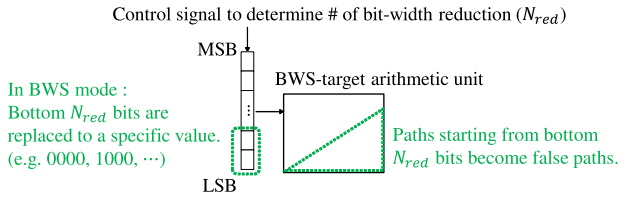

Fig. 2 Assumed BWS. In BWS operation, bottom bits will be replaced to the user-defined value. Paths starting from replaced bits become false paths and will not toggle in the arithmetic unit.

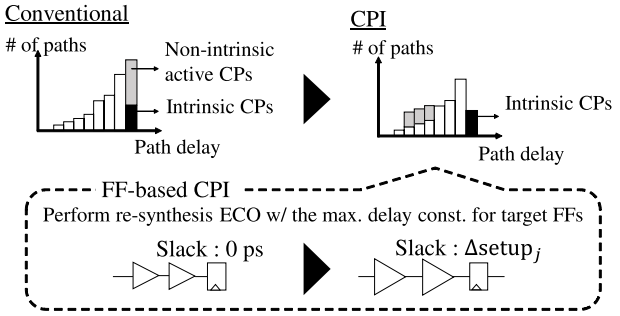

Fig. 3 Assumed FF-based CPI with re-synthesis. CPI tries to increase the timing slack of target FFs, e.g., by increasing the setup time artificially or reducing the maximum delay constraint. After CPI, we expect that the circuit has fewer paths where timing error is likely to occur.

increases the timing slack of non-intrinsic CPs. In this case, timing error occurrence in these paths is dramatically reduced compared to the conventional circuit, which is the main advantage of CPI. Note that this work assumes the logic synthesis based CPI since this design phase allows CPI to change the circuit structure flexibly.

Here, it should be noted that CPI partially loses the power and area reduction acquired by the conventional design optimization. From this sense, we need to find a better trade-off relation regarding the minimum supply voltage under VOS, power, and area. For pursuing the better trade-off, this work targets active CPs, i.e., paths actually causing timing errors, where a similar consideration is found in literature [5], [12], [18], [19]. Also, we refer to [12] and adopt FF-based CPI; assigns manipulated setup delay constraints to FFs, and re-synthesizes the design as an engineering change order (ECO) process. Note that path-based CPI is not efficient since the number of paths in a circuit is huge. Section 3.3 will explain how to determine the delay constraint for target FFs.

2.2 Problem Formulation for Voltage Over-Scaled Design

Based on the discussion in Sect. 2.1, we formulate the design optimization of the main logic circuit under VOS.

- Input
  - one pre-CPI circuit
  - N_W workloads
- Output
  - one circuit to which CPI and BWS are applied
- Objective
  - Minimize: V_{dd_i} for each i $(1 \le i \le N_W)$
- Constraints
  - $Quality_i \ge \text{Quality}_i^{\min}$ $(1 \le i \le N_W)$
  - $Area \le \text{Area}^{\max}$
  - $N_{LowVth} \le N_{\text{LowVth}}^{\max}$
- Variables
  - N_{red_i} $(1 \le i \le N_W)$
  - D_{FF_j} $(1 \le j \le N_{FF})$

The input of this problem is one pre-CPI circuit and N_W workloads, and the output is one synthesized design incorporating CPI and BWS. The objective of this problem is to minimize the supply voltage for each assumed workload aiming at the power minimization. The design constraints are computational quality ($\text{Quality}_i^{\min}$), total cell area ($\text{Area}^{\max}$), and the number of low-Vth cells ($N_{\text{LowVth}}^{\max}$). In other words, we aim to implement BWS and CPI so that the designed circuit can achieve the target quality for assumed N_W workloads under area and power constraints. Note that we assume these constraints are given to the designer according to the requirement for the target circuit. The variable N_{red_i} means the number of bit-width reduction for the i-th workload. The variable D_{FF_j} is given to the re-synthesis ECO as a maximum delay constraint for j-th FF_j. N_{FF} is the number of FFs in the circuit. In summary, the pair of N_{red_i} and V_{dd_i} needs to be examined for each workload, and D_{FF_j} should be carefully tuned taking into account N_W workloads.

3. Proposed Design Methodology

In this section, we propose a design methodology to solve the problem described in the previous section.

3.1 Overview

A difficulty in solving the formulated problem is the non-linear relationships among $Area$, V_{dd_i}, $Quality_i$, N_{LowVth}, N_{red_i}, and D_{FF_j}. Also, the evaluations of $Area$, V_{dd_i}, and $Quality_i$ need relatively long computational time, and hence an explicit optimization is difficult concerning computational time. Here, please remind that CPI and BWS are expected to reduce the minimum supply voltage for different parts of the circuit, i.e., CPI targets non-intrinsic CPs and BWS focuses on intrinsic CPs in arithmetic units. This target exclusiveness guides us a lightweight design methodology that efficiently reduces the co-design exploration space. Namely, the simultaneous optimization of CPI and BWS for the supply voltage minimization could be split into sub-problems.

Based on the above consideration, this work proposes a three-stage design flow. The first stage finds the maximum number of N_{red_i} under the quality constraint. The second stage determines the set of D_{FF_j} for successive CPI under area and leakage power constraints. After BWS and CPI are implemented, in the third stage, the supply voltage of the

designed circuit is swept for minimizing power dissipation of each workload. We highlight that the simpleness of the proposed flow originates from the exclusiveness of target paths between CPI and BWS. Note that BWS may change the number of non-intrinsic CPs. In this case, the successive CPI can be utilized to eliminate remaining non-intrinsic CPs.

Here, the first stage of N_{red_i} determination does not take into account the constraints of area and the number of low-Vth cells. This is because we assume the tunable BWS. More specifically, even if we vary N_{red_i} via the control signal, the circuit area and the number of low-Vth cells do not change. Based on the above consideration, the N_{red_i} determination takes into account the quality constraint. Similarly, the second stage does not consider the quality constraint. Since we assume to use CPI for increasing the timing slack of selected FFs, we expect that CPI does not degrade the computational quality. Therefore, the second stage focuses on the constraints of area and the number of low-Vth cells. The following subsections explain the first and second stages.

3.2 N_{red_i} Determination

First, N_{red_i} is determined for each i-th workload. This stage aims to reduce the intrinsic CP delay and dynamic power dissipation in target arithmetic units as much as possible. Such a reduction enhances the supply voltage reduction effect by CPI. Therefore, the proposed flow investigates the maximum N_{red_i} for improving the effectiveness of CPI.

An important consideration here is that the maximum N_{red_i} varies depending on the quality constraint and workloads. In addition, this N_{red_i} can be derived without timing analysis since functional simulation, e.g., instruction set simulation or register transfer level (RTL) simulation, provides the upper bound. Therefore, the first stage runs functional simulation, evaluates the trade-off curve between N_{red_i} and the computational quality, and determines the maximum N_{red_i} where the target quality is satisfied. We note that this stage can be extended similarly to take into account the multiple quality constraints in each workload.

3.3 D_{FF_j} Determination

Then, the proposed design methodology determines the set of D_{FF_j}. Figure 4 shows a CPI flow, which is adopted in this work. Note that other strategies for D_{FF_j} determination, e.g., the methodology proposed in [12], can be similarly utilized.

First, let us explain CPI-target FFs. In Fig. 4, after extracting active FFs from the given BWS circuit and N_W workloads, the proposed methodology takes two types of constraints for ECO re-synthesis. The first constraint targets active-endpoint FFs, which is similar to the conventional CPI [12]. In addition, for eliminating the non-intrinsic CPs, the second constraint focuses on top k bits of the input to the BWS-target arithmetic unit, where how to determine k will be discussed below.

Figure 5 depicts the motivation of the second constraint,

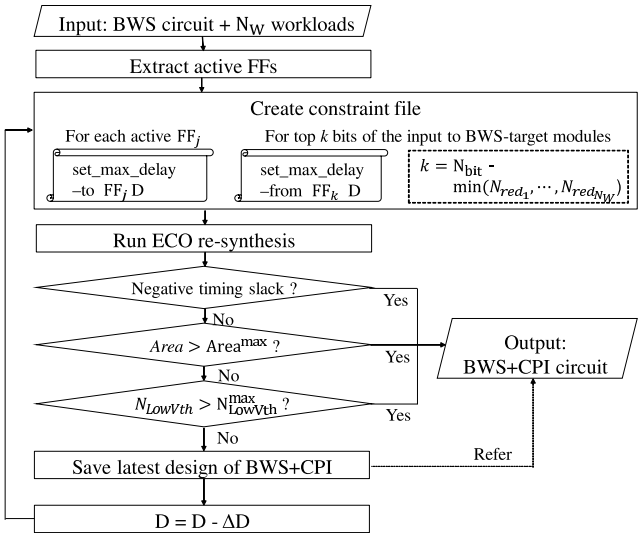

Fig. 4 Overview of the CPI flow. The flow repeatedly performs the ECO re-synthesis with updating the delay constraint file. As for the detail of the constraint file, please refer to Sect. 3.3. When the synthesized design violates the constraints of setup timing, $\text{Area}^{\max}$, or $\text{N}^{\max}_{\text{LowVth}}$, the flow finishes the iteration of CPI and outputs the latest saved design.

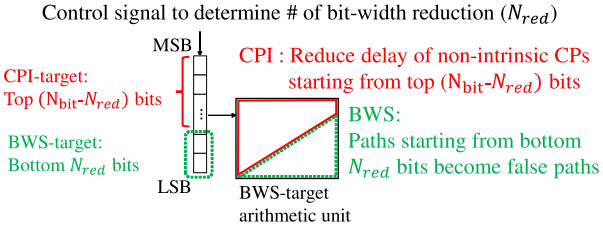

Fig. 5 Startpoint-aware CPI for reducing non-intrinsic CPs in the BWS-target arithmetic unit.

i.e., startpoint-aware CPI. Let us suppose that BWS targets the bottom N_{red} bits of the input as an example. Since the assumed BWS is tunable, the BWS module may have intrinsic CPs for realizing an exact calculation, e.g., from the least significant bit (LSB) of the input to the most significant bit (MSB) of the output. In this case, the paths starting from top (N_{bit}-N_{red}) bits of the input may become non-intrinsic CPs due to the timing optimization. Note that N_{bit} is the bit-width of the input. If the MSB of the output is an endpoint of these non-intrinsic CPs, conventional endpoint-based CPI could not reduce the delay of these CPs well since the MSB is also an endpoint FF of intrinsic CPs. On the other hand, with our strategy, CPI can reduce non-intrinsic CPs starting from top (N_{bit}-N_{red}) bits thanks to timing constraints for startpoint FFs. Therefore, the supply voltage and thus power dissipation can be further reduced. Section 4.2.2 will examine the effectiveness of the second startpoint-aware constraint in terms of the supply voltage reduction. Note that, for taking into account different N_{red_i} under various workloads, we set k with the following equation.

$$k = \text{N}_{\text{bit}} - \min(N_{red_i}, \cdots, N_{red_{N_W}}). \quad (1)$$

In Eq. (1), $\min(N_{red_i}, \cdots, N_{red_{N_W}})$ means the minimum

number of bit-width reduction derived from assumed workloads. Namely, we conservatively determine k so that CPI can eliminate non-intrinsic CPs in the BWS-target unit even for the workload with the minimum value of N_{red}. We note that when intrinsic CPs do not start from lower bits of inputs, the truncation of these lower bits and successive CPI may not contribute to reducing the delay of intrinsic CPs well.

Next, the value of maximum delay constraint, D_{FF_j}, is discussed. When we continue to reduce the worst delay of the target FFs, some FFs will not satisfy the specified delay constraint at a certain stage. In this case, such delay-limiting FFs determine the maximum delay of the circuit. In other words, once we find these FFs and derive the achievable delay reduction, we could skip the further ECO re-synthesis, which contributes to eliminating the redundant ECO process.

Taking into account the above discussion, our flow gives an identical constraint regarding the maximum delay, i.e., D, for each CPI-target FF as described in Fig. 4. This approach finds the delay-limiting FF and thus generates the CPI circuit whose worst delay is decreased as much as possible. We repeatedly update D with reducing by ΔD and run ECO re-synthesis with checking the timing slack, the number of low-Vth cells, and the area. Note that the amount of the delay reduction step, i.e., ΔD, can be tuned by designers under given design time. Once the ECO re-synthesis generates the circuit having negative timing slack, we can exit from CPI since the delay-limiting FFs reject further delay reduction. Note that our flow checks the area and the number of low-Vth cells of the synthesized circuit for satisfying constraints of the design optimization problem discussed in Sect. 2.2.

4. Experimental Evaluation

This section experimentally evaluates the power saving of the proposed design. Section 4.1 explains the evaluation setup. Section 4.2 shows the power saving effects and demonstrates the proposed design achieves the lower power dissipation compared with naive VOS and conventional CPI and BWS.

4.1 Evaluation Setup

In this work, we used a Nyuzi processor, which is an open-source processor for GPGPU applications [20]. This circuit was designed by a commercial logic synthesis tool with a 45 nm Nangate standard cell library. The minimum clock period of the synthesized circuit at the worst corner is 1.24 ns. The synthesized circuit includes 184,243 combinational logic cells and 29,456 FFs.

As for workloads, we selected two programs. The first program is Mandelbrot set drawing. The Mandelbrot set is a set of points in the complex plane derived from the specific recurrence formula forming a fractal. Since the fractal calculation has a high degree of parallelism and a simple control structure [21], this workload can be a good candidate as the benchmark for parallel computing, e.g., GPGPU programming [22]. The second program is neural network inference with Fourclass dataset [23], which is a simple 2-D classification problem. Here, it should be noted that the VOS logic simulation to evaluate the inference accuracy is very time-consuming. To make the CPU time needed for the entire evaluation in this work acceptable, we prepared a simple 3-layer neural network structure that includes one hidden layer. The numbers of neurons in the input layer, hidden layer, and output layer are set to 2, 8, and 2, respectively. We adopt Rectified Linear Unit (ReLU) as the activation function of hidden layer. Note that one VOS logic simulation for Mandelbrot set and Fourclass case took 6 hours and 8 hours, respectively. We note that the required computational time for the proposed design methodology depends on not only the CPU machine and the circuit size but also a set of workloads. Therefore, the scalability of the proposed design, e.g., the number of active FFs that we can consider, is also depending on the set of workloads, the spec of CPU machine, and the target circuit.

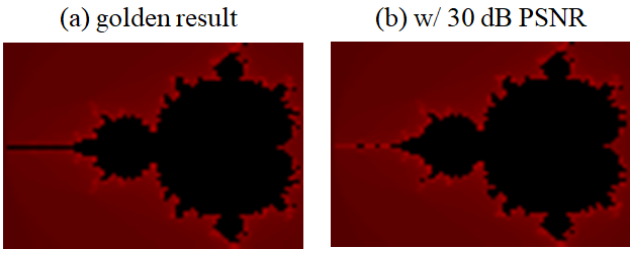

Fig. 6 The Manldelbrot set with (a) no quality loss and (b) 30 dB PSNR.

In our experiment, as quality constraints ($\text{Quality}^{\text{min}}$), we set 30 dB of PSNR for Mandelbrot and 98% test accuracy for Fourclass. Figure 6 shows the Mandelbrot set whose PSNR is 30 dB and without precision loss. Note that the above $\text{Quality}^{\text{min}}$ is just an example, and the proposed design methodology can cope with other constraints similarly.

We incorporated BWS and CPI into the baseline Nyuzi processor. This paper focuses on floating-point units (FPUs) since FPUs are known as the power-hungry unit [24], and they often include intrinsic CPs. Note that floating-point numbers are expressed by 32 bits in the Nyuzi processor. Then, we reduced the bit-width of the mantissa in the FPUs and performed the RTL simulation for determining N_{red}. Figure 7 shows the simulation result when replacing the bottom N_{red} bits with the value "0". From Fig. 7, we can see that the Nyuzi satisfies $\text{Quality}^{\text{min}}$ under condition that $N_{red} \leq 13$ in Mandelbrot and $N_{red} \leq 20$ in Fourclass case. From these results, we set N_{red} to 13 for Mandelbrot and 20 for Fourclass, respectively. We should note that PSNR becomes ∞ when the image has an identical set of pixel values with golden results. For visualizing such a case, we set the upper bound of PSNR to 50 dB in Fig. 7(a).

Then, CPI was performed to the synthesized circuit. For determining target FFs, we performed logic simulation with a commercial tool and extracted the active FFs. The number of active FFs was 19,734. We repeatedly performed ECO synthesis by decreasing the maximum delay of target FFs according to the design flow in Sect. 3.3. Constraints of total cell area (Area^{max}) and the number of low-Vth cells ($\text{N}^{\text{max}}_{\text{LowVth}}$)

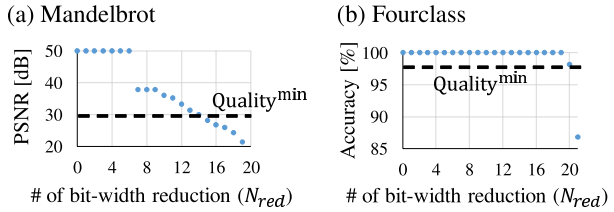

Fig. 7 The relation between N_{red} and Quality$^{\min}$ in (a) Mandelbrot and (b) Fourclass.

were set to 101.0% and 103.0% of the baseline circuit. Note that the above constraints are prepared as just an example, and the proposed design methodology can cope with other constraints similarly. To make the CPU time required for iterating the re-synthesis ECO acceptable, we set the delay update for re-synthesis constraint (ΔD) to 10 ps. Note that we used a 2.2 GHz Xeon CPU machine with 93 GB memory for the entire evaluation, and one re-synthesis ECO took 10 minutes.

In addition, we set k to 9, which is derived by subtracting the minimum $N_{\mathrm{red_i}}$ for assumed workloads, i.e., 14, from the bit-width of fraction bit, i.e., 23. Note that the proposed design methodology can work similarly with other settings. On the other hand, since k depends on given workloads and quality constraints, we expect that the proposed design methodology is applied to the circuit where the allowable values of N_{red_i} are relatively large for given workloads and quality constraints, e.g., approximate computing circuit where the inherent resilience of applications allows a large amount of functional and timing modification. For the baseline design, BWS design, CPI design, and the proposed design, we fixed the clock period to 1.24 ns, swept the supply voltage from 1.1 V to 0.9 V with 10 mV interval, and performed the VOS logic simulation at each supply voltage. Thus, the trade-off relationship between the supply voltage and computational quality can be obtained. Then, we analyzed power dissipation for each pair of the circuit and the supply voltage, and obtained the trade-off relationship between the power dissipation and quality.

4.2 Evaluation Results

This subsection first shows power savings thanks to the proposed design, and then discusses the effectiveness of CPI and BWS, respectively.

4.2.1 Power Saving Effects

Figure 8 shows the trade-off curves between the power dissipation and constraints, i.e., PSNR and test accuracy. Remind that the quality constraint is set to 30 dB PSNR for Mandelbrot and 98% test accuracy for Fourclass. In this figure, the black plots represent the naive VOS which reduces the supply voltage without any timing optimization. Also, red, green, and blue plots correspond to the conventional CPI, the conventional BWS, and the proposed design, respectively. As a baseline, we note that the conventional worst-case design without VOS consumes 450.0 mW in Mandelbrot and 567.5 mW in Fourclass. Here, it should be noted that when we set the lower supply voltage, we observed several cases where the VOS simulation did not finish correctly due to fatal timing errors, e.g., errors in program counters. For these cases, since we could not obtain the computational results such as PSNR and inference accuracy, we did not plot these results to the figure. In this section, we examine our evaluation results from the following two aspects; (1) overall power saving effect thanks to the proposed design, and (2) difference of the power dissipation between the proposed, conventional CPI, and conventional BWS design.

First, we compare the black and blue plots for clarifying the overall power saving effects. Figure 8 shows that the proposed design saves power dissipation while keeping the quality constraint. For example, in Fig. 8(a-1), the proposed design achieves the quality constraint of 30 dB at the power of 257.8 mW, whereas the conventional VOS design consumed 427.2 mW. In other words, the proposed design achieved 42.7% power savings from 450.0 mW to 257.8 mW whereas the naive VOS achieved only 5.1% power savings from 450.0 mW to 427.2 mW. Similarly, in Fourclass case, the proposed design saves the power dissipation by 51.2% from 567.5 mW to 277.4 mW as shown in Fig. 8(a-2). Compared with the baseline circuit, the proposed design increased the number of low-Vth cells by 0.11% but decreased the total cell area by 0.58%.

Next, we compare the conventional CPI, conventional BWS, and the proposed design. Figure 8 shows that the proposed design further improves power dissipation from the conventional CPI and BWS. For example, from Fig. 8(a), the proposed design achieved 22.3% and 38.7% power savings compared with the conventional CPI. Similarly, compared with the conventional BWS, we can see that the proposed design saves the power dissipation by 31.0% in Mandelbrot and 35.9% in Fourclass case as shown in Fig. 8(b). These power saving effects reveal that BWS and CPI are highly compatible and the optimization of BWS and CPI significantly enhances the efficacy of VOS.

4.2.2 Discussion

The evaluation results for power dissipation in Sect. 4.2.1 showed that the proposed design saved power significantly. Let us investigate the results in detail.

First, we examine the power saving effects by the proposed design in terms of the supply voltage reduction thanks to the proposed design and the dynamic power savings thanks to BWS. Figure 9 shows the trade-off curves between the supply voltage and the computational quality. We can see that the proposed design achieves the target quality at a lower supply voltage compared with the naive VOS, conventional CPI, and conventional BWS. For example, in Fig. 9(a), the proposed design achieves the target quality at a supply voltage of 0.93 V, whereas the naive VOS requires 1.07 V operation, which means the proposed design achieves 13.0% supply voltage reduction from the naive VOS. Thanks to the supply voltage

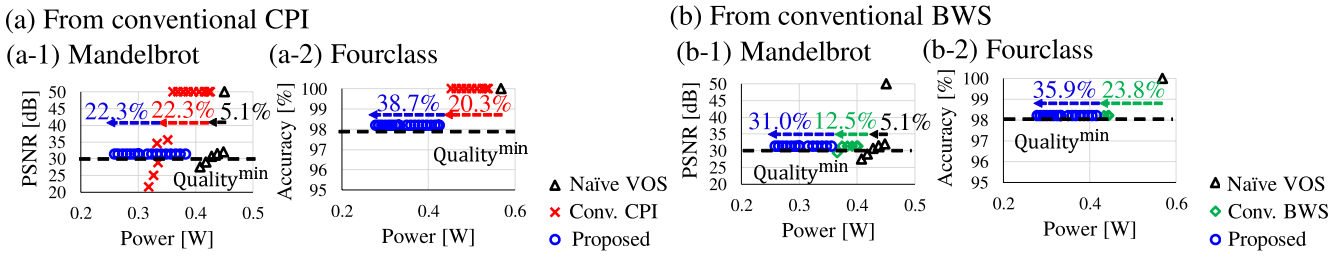

Fig. 8 Power saving effects of the proposed design from the conventional CPI and BWS. (a) From conventional CPI, the proposed design reduces the power dissipation thanks to BWS. (b) From conventional BWS, the proposed design mitigates the quality degradation slope thanks to CPI.

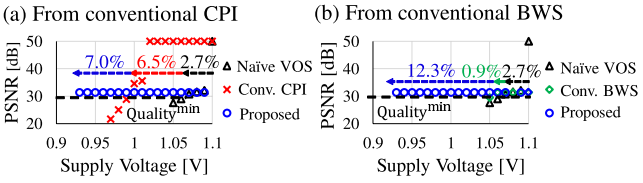

Fig. 9 The supply voltage reduction thanks to the proposed design in Mandelbrot case (a) from conventional CPI and (b) from conventional BWS.

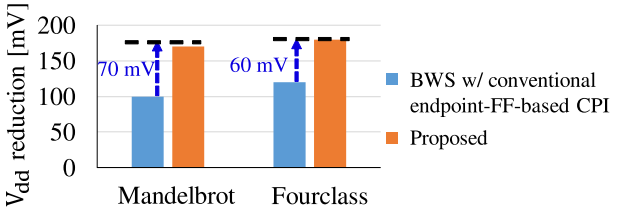

Fig. 10 Comparison of supply voltage reduction between conventional endpoint-based CPI with BWS and the proposed design. The proposed design further improves the supply voltage reduction effects.

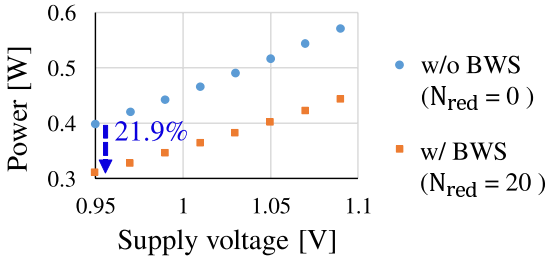

Fig. 11 Power comparison between with and without BWS in Fourclass case. BWS saves the power dissipation even at the identical supply voltage thanks to the dynamic power reduction.

reduction, the circuit power dissipation is dramatically reduced as shown in Fig. 8. Also, these results suggest that the reduction of both intrinsic and non-intrinsic CPs is crucially important for enhancing the efficacy of quality-aware VOS. We note that Fig. 9 does not plot the computational quality for cases where the VOS logic simulation did not finish correctly due to fatal timing errors, which is similar to Fig. 8.

Figure 10 compares the supply voltage reduction of BWS circuit with conventional endpoint-based CPI and with our CPI strategy in Sect. 3.3. From Fig. 10, we can see that our startpoint-aware CPI further contributes to reducing the supply voltage compared with the conventional CPI. For example, in the Mandelbrot case, our proposed design increases the supply voltage reduction effects by 70 mV from 100 mV to 170 mV. This result indicates that our CPI cooperates with the tunable BWS well and hence their co-design optimization dramatically reduces CPs and thus the supply voltage.

Figure 11 compares the power dissipation of Nyuzi processor with and without BWS in the Fourclass case. From Fig. 11, we can see that BWS reduces the power dissipation even at the identical supply voltage. For example, in the case where the supply voltage is set to 0.95 V, BWS reduces the power dissipation by 21.9% from 398.6 mW to 311.5 mW. This power reduction originates from the dynamic power savings, as previously explained with Fig. 2 in Sect. 2.1. Such a dynamic power reduction also contributes to enhancing the power saving effects of the proposed design.

Lastly, this section examines the power saving effect thanks to CPI at different PVTA corners. We replaced the corner information in liberty files from the worst corner to the typical corner, and then swept the supply voltage for finding the minimum supply voltage, where the optimized circuit design was unchanged. Note that the clock period was fixed to 1.24 ns and only the corner information was substituted. By changing the corner information, the delay slope and sensitivity of each gate in the circuit dramatically vary. Therefore, the dependability of the proposed design against delay variability can be experimentally evaluated.

Figure 12 shows the comparison results between the proposed design and the conventional BWS. Similar to Figs. 8 and 9, Fig. 12 does not include the computational quality results for cases where the VOS logic simulation did not finish correctly. From Fig. 12, we can see that the proposed design saves power dissipation thanks to CPI even at the typical corner. For example, from Fig. 12(a), proposed design saves power dissipation by 21.1% from 168.5 mW to 133.0 mW. From the above, we experimentally confirmed that the proposed design made the significant power savings even when operating at different PVTA corners. We expect that such a variation-tolerant design is useful for self-tuning design such as dynamic frequency voltage scaling (DVFS) [25], [26], minimum energy point tracking (MEPT) [27], [28], and adaptive voltage scaling (AVS) [29]–[32].

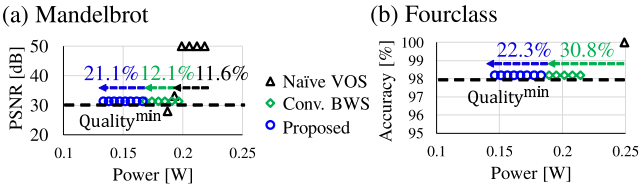

Fig. 12 Power saving effects thanks to the proposed design at a typical corner in (a) Mandelbrot and (b) Fourclass case.

5. Conclusion

This work proposed a design methodology that saves the power under VOS operation. The key idea of the proposed design is to combine CPI and BWS. Evaluation results show that BWS is highly compatible with CPI and they dramatically enhance the efficacy of VOS. In the case study of the GPGPU processor, the proposed design saves the power dissipation by 42.7% with the image processing workload and 51.2% with the neural network inference workload.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number JP19K24341 and JP20K19767.

References

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," Proc. ETS, pp.1–6, 2013.
[2] V.K. Chippa, S.T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," Proc. DAC, pp.1–9, 2013.
[3] Q. Xu, T. Mytkowicz, and N.S. Kim, "Approximate computing: A survey," IEEE Des. Test, vol.33, no.1, pp.8–22, 2016.
[4] R. Hegde and N.R. Shanbhag, "Soft digital signal processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.9, no.6, pp.813–823, 2001.
[5] A.B. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," Proc. ASP-DAC, pp.825–831, 2010.
[6] R. Venkatesan, A. Agarwal, K. Roy and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," Proc. ICCAD, pp.667–673, 2011.
[7] V. Gupta, D. Mohapatra, S.P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," Proc. ISLPED, pp.409–414, 2011.
[8] P.K. Krause and I. Polian, "Adaptive voltage over-scaling for resilient applications," Proc. DATE, pp.1–6, 2011.
[9] R. Ragavan, B. Barrois, C. Killian, and O. Sentieys, "Pushing the limits of voltage over-scaling for error-resilient applications," Proc. DATE, pp.476–481, 2017.
[10] R. Li, R. Naous, H. Fariborzi, and K.N. Salama, "Approximate computing with stochastic transistors' voltage over-scaling," IEEE Access, vol.7, pp.6373–6385, 2019.
[11] B. Shim, S.R. Sridhara, and N.R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.12, no.5, pp.497–510, 2004.
[12] Y. Masuda, M. Hashimoto, and T. Onoye, "Critical path isolation for time-to-failure extension and lower voltage operation," Proc. ICCAD, pp.1–8, 2016.
[13] J.Y.F. Tong, D. Nagle, and R.A. Rutenbar, "Reducing power by optimizing the necessary precision/range of floating-point arithmetic," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.8, no.3, pp.273–286, 2000.
[14] K. Kunaparaju, S. Narasimhan, and S. Bhunia, "VaROT: Methodology for variation-tolerant DSP hardware design using post-silicon truncation of operand width," Proc. VLSID, pp.310–315, 2011.
[15] D. Kim, J. Kung, and S. Mukhopadhyay, "A power-aware digital multilayer perceptron accelerator with on-chip training based on approximate computing," IEEE Trans. Emerg. Topics Comput., vol.5, no.2, pp.164–178, 2017.
[16] I. Tsiokanos, L. Mukhanov, and G. Karakonstantis, "Low-power variation-aware cores based on dynamic data-dependent bitwidth truncation," Proc. DATE, pp.698–703, 2019.
[17] Y. Masuda, J. Nagayama, T. Cheng, T. Ishihara, Y. Momiyama, and M. Hashimoto, "Critical path isolation and bit-width scaling are highly compatible for voltage over-scalable design," Proc. DATE, pp.1260–1265, 2021.
[18] S. Ghosh, S. Bhunia, and K. Roy, "CRISTA: A new paradigm for low-power, variation-tolerant, and adaptive circuit synthesis using critical path isolation," Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.26, no.11, pp.1947–1956, 2007.
[19] B. Greskamp, L. Wan, U.R. Karpuzcu, J.J. Cook, J. Torrellas, D. Chen, and C. Zilles, "Blueshift: Designing processors for timing speculation from the ground up," Proc. HPCA, pp.213–224., 2009.
[20] J. Bush, NyuziProcessor Source code, https://github.com/jbush001/NyuziProcessor, 2015.
[21] M.J. Quinn and P.J. Hatcher, "Data-parallel programming on multicomputers," IEEE Softw., vol.7, no.5, pp.69–76, Sept. 1990.
[22] M. Steuwer, P. Kegel, and S. Gorlatch, "SkelCL - A portable skeleton library for high-level GPU programming," IPDPS PhD Forum, pp.1176–1182, 2011.
[23] C. Chang and C. Lin, Fourclass, 1996. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html
[24] T. Cheng, J. Yu, and M. Hashimoto, "Minimizing power for neural network training with logarithm-approximate floating-point multiplier," Proc. PATMOS, pp.91–96, 2019.
[25] T.D. Burd, T.A. Pering, A.J. Stratakos, and R.W. Brodersen, "A dynamic voltage scaled microprocessor system," IEEE J. Solid-State Circuits, vol.35, no.11, pp.1571–1580, 2000.
[26] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "A self-tuning DVS processor using delay-error detection and correction," IEEE J. Solid-State Circuits, vol.41, no.4, pp.792–804, 2006.
[27] S. Hokimoto, T. Ishihara, and H. Onodera, "Minimum energy point tracking using combined dynamic voltage scaling and adaptive body biasing," Proc. SOCC, pp.1–6, 2016.
[28] N. Mehta and K.A.A. Makinwa, "Minimum energy point tracking for sub-threshold digital CMOS circuits using an in-situ energy sensor," Proc. ISCAS, pp.570–573, 2013.
[29] K.A. Bowman, J.W. Tschanz, S.-L.L. Lu, P.A. Aseron, M.M. Khellah, A. Raychowdhury, B.M. Geuskens, C. Tokunaga, C.B. Wilkerson, T. Karnik, and V.K. De, "A 45 nm resilient microprocessor core for dynamic variation tolerance," IEEE J. Solid-State Circuits, vol.46, no.1, pp.194–208, 2011.
[30] H. Fuketa, M. Hashimoto, Y. Mitsuyama, and T. Onoye, "Adaptive performance compensation with in-situ timing error predictive sensors for subthreshold circuits," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.20, no.2, pp.333–343, 2012.
[31] S. Kim and M. Seok, "Variation-tolerant, ultra-low-voltage microprocessor with a low-overhead, within-a-cycle in-situ timing-error detection and correction technique," IEEE J. Solid-State Circuits, vol.50, no.6, pp.1478–1490, 2015.
[32] M. Cho, S.T. Kim, C. Tokunaga, C. Augustine, J.P. Kulkarni, K. Ravichandran, J.W. Tschanz, M.M. Khellah, and V. De, "Postsilicon voltage guard-band reduction in a 22 nm graphics execution core using adaptive voltage scaling and dynamic power gating," IEEE J.

Solid-State Circuits, vol.52, no.1, pp.50–63, 2017.

Yutaka Masuda received the B.E., M.E., and Ph.D. degrees in Information Systems Engineering from the Osaka University, Osaka, Japan, in 2014, 2016, and 2019, respectively. He is currently an Assistant Professor in Center for Embedded Computing Systems, Graduate School of Informatics, Nagoya University. His research interests include low-power circuit design. He serves on the Technical Program Committee of international conferences including ASP-DAC. He is a member of IEEE, IEICE, and IPSJ.

Jun Nagayama received the B.S. and M.S. degrees in physics from Sophia University, Tokyo, Japan, in 2002 and 2004, respectively. In 2004, he joined the Fujitsu Ltd., Kawasaki, Japan. In 2015, he moved the Socionext Inc., Yokohama, Japan. He has been engaged in research and development of low-power CMOS design.

TaiYu Cheng received the B.E. and M.E. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2010 and 2012, respectively. From 2012 to 2018, he was with Taiwan Semiconductor Manufacturing Company, Hsinchu, Taiwan, where he has been engaged in design flow of timing closure. Since 2018, he has been a Ph.D. student with the Department of Information Systems Engineering, Osaka University, Osaka, Japan. His research interests include low power circuit design.

Tohru Ishihara received his Dr. Eng. degree in computer science from Kyushu University in 2000. For the next three years, he was a Research Associate in the University of Tokyo. From 2003 to 2005, he was with Fujitsu Laboratories of America as a Research Staff of an Advanced CAD Technology Group. From 2005 to 2011, he was with Kyushu University and for the next seven years he was with Kyoto University as an Associate Professor. In October 2018, he joined Nagoya University where he is currently a Professor in the Department of Computing and Software Systems. His research interests include low-power design methodologies and power management techniques for embedded systems. Dr. Ishihara is a member of the IEEE, ACM and IPSJ.

Yoichi Momiyama received the B.S. and M.S. degrees in electronics engineering from Niigata University, Niigata, Japan, in 1990 and 1992, respectively. In 1992, he joined Fujitsu Laboratories Ltd., Atsugi, Japan, where he has been engaged in research and development of low-power and high-speed CMOS devices. He moved to the Socionext Inc. at 2015, where he has been investigating CMOS low-power design. His research interests include CMOS low-power enablement. He received the Best Paper Award at the 1st conference of IEEE IWJT. He has been serving as editor of *IEEE Transaction on Electron Devices* since 2011. Mr. Momiyama is a member of the IEEE electron devices society, solid-state circuits society and IEICE.

Masanori Hashimoto received the B.E., M.E., and Ph.D. degrees in communications and computer engineering from Kyoto University, Kyoto, Japan, in 1997, 1999, and 2001, respectively. He is currently a Professor with the Department of Information Systems Engineering, Graduate School of Information Science and Technology, Osaka University, Suita, Japan. His current research interests include the design for manufacturability and reliability, timing and power integrity analysis, reconfigurable computing, soft error characterization, and low-power circuit design. He was a recipient of the Best Paper Award from Asia and South Pacific Design Automation Conference in 2004 and the Best Paper Award of the *IEICE Transactions* in 2016. He was on the Technical Program Committee of international conferences, including the Design Automation Conference, the International Conference on Computer Aided Design, the International Test Conference, the Symposium on VLSI Circuits, ASP-DAC, and DATE. He serves/served as the Editor-in-Chief for *Microelectronics Reliability* (Elsevier) and an Associate Editor for the *IEEE Transactions on Very Large Scale Integration*, *IEEE Transactions on Circuits and Systems—I: Regular Papers*, and *ACM Transactions on Design Automation of Electronic Systems*.