High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels

Milad Bahadori, Mehdi Kamal, Ali Afzali-Kusha, *Senior Member, IEEE*, and Massoud Pedram, *Fellow, IEEE*

Abstract— **In this paper, we present a carry skip adder (CSKA) structure that has a higher speed yet lower energy consumption compared with the conventional one. The speed enhancement is achieved by applying concatenation and incrementation schemes to improve the efficiency of the conventional CSKA (Conv-CSKA) structure. In addition, instead of utilizing multiplexer logic, the proposed structure makes use of AND-OR-Invert (AOI) and OR-AND-Invert (OAI) compound gates for the skip logic. The structure may be realized with both fixed stage size and variable stage size styles, wherein the latter further improves the speed and energy parameters of the adder. Finally, a hybrid variable latency extension of the proposed structure, which lowers the power consumption without considerably impacting the speed, is presented. This extension utilizes a modified parallel structure for increasing the slack time, and hence, enabling further voltage reduction. The proposed structures are assessed by comparing their speed, power, and energy parameters with those of other adders using a 45-nm static CMOS technology for a wide range of supply voltages. The results that are obtained using HSPICE simulations reveal, on average, 44% and 38% improvements in the delay and energy, respectively, compared with those of the Conv-CSKA. In addition, the power–delay product was the lowest among the structures considered in this paper, while its energy–delay product was almost the same as that of the Kogge–Stone parallel prefix adder with considerably smaller area and power consumption. Simulations on the proposed hybrid variable latency CSKA reveal reduction in the power consumption compared with the latest works in this field while having a reasonably high speed.**

Index Terms— **Carry skip adder (CSKA), energy efficient, high performance, hybrid variable latency adders, voltage scaling.**

I. Introduction

ADDERS are a key building block in arithmetic and logic units (ALUs) [1] and hence increasing their speed and reducing their power/energy consumption strongly affect the speed and power consumption of processors. There are many works on the subject of optimizing the speed and power of these units, which have been reported in [2]–[9]. Obviously, it is highly desirable to achieve higher speeds at low-power/energy consumptions, which is a challenge for the designers of general purpose processors.

One of the effective techniques to lower the power consumption of digital circuits is to reduce the supply voltage due to quadratic dependence of the switching energy on the voltage. Moreover, the subthreshold current, which is the main leakage component in OFF devices, has an exponential dependence on the supply voltage level through the drain-induced barrier lowering effect [10]. Depending on the amount of the supply voltage reduction, the operation of ON devices may reside in the superthreshold, near-threshold, or subthreshold regions. Working in the superthreshold region provides us with lower delay and higher switching and leakage powers compared with the near/subthreshold regions. In the subthreshold region, the logic gate delay and leakage power exhibit exponential dependences on the supply and threshold voltages. Moreover, these voltages are (potentially) subject to process and environmental variations in the nanoscale technologies. The variations increase uncertainties in the aforesaid performance parameters. In addition, the small subthreshold current causes a large delay for the circuits operating in the subthreshold region [10].

Recently, the near-threshold region has been considered as a region that provides a more desirable tradeoff point between delay and power dissipation compared with that of the subthreshold one, because it results in lower delay compared with the subthreshold region and significantly lowers switching and leakage powers compared with the superthreshold region. In addition, near-threshold operation, which uses supply voltage levels near the threshold voltage of transistors [11], suffers considerably less from the process and environmental variations compared with the subthreshold region.

The dependence of the power (and performance) on the supply voltage has been the motivation for design of circuits with the feature of dynamic voltage and frequency scaling. In these circuits, to reduce the energy consumption, the system may change the voltage (and frequency) of the circuit based on the workload requirement [12]. For these systems, the circuit should be able to operate under a wide range of supply voltage levels. Of course, achieving higher speeds at lower supply voltages for the computational blocks, with the adder as one the main components, could be crucial in the design of high-speed, yet energy efficient, processors.

In addition to the knob of the supply voltage, one may choose between different adder structures/families for

Manuscript received October 8, 2014; revised January 16, 2015; accepted February 13, 2015. The work of M. Bahadori, M. Kamal, and A. Afzali-Kusha was supported by the Iranian National Science Foundation. The work of M. Pedram was supported by the U.S. National Science Foundation.

M. Bahadori, M. Kamal, A. Afzali-Kusha are with the School of Electrical and Computer Engineering, University of Tehran, Tehran 19967-15433, Iran (e-mail: milad.bahadori@ut.ac.ir; mehdikamal@ut.ac.ir; afzali@ut.ac.ir).

M. Pedram is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: pedram@usc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2015.2405133

optimizing power and speed. There are many adder families with different delays, power consumptions, and area usages. Examples include ripple carry adder (RCA), carry increment adder (CIA), carry skip adder (CSKA), carry select adder (CSLA), and parallel prefix adders (PPAs). The descriptions of each of these adder architectures along with their characteristics may be found in [1] and [13]. The RCA has the simplest structure with the smallest area and power consumption but with the worst critical path delay. In the CSLA, the speed, power consumption, and area usages are considerably larger than those of the RCA. The PPAs, which are also called carry look-ahead adders, exploit direct parallel prefix structures to generate the carry as fast as possible [14]. There are different types of the parallel prefix algorithms that lead to different PPA structures with different performances. As an example, the Kogge–Stone adder (KSA) [15] is one of the fastest structures but results in large power consumption and area usage. It should be noted that the structure complexities of PPAs are more than those of other adder schemes [13], [16].

The CSKA, which is an efficient adder in terms of power consumption and area usage, was introduced in [17]. The critical path delay of the CSKA is much smaller than the one in the RCA, whereas its area and power consumption are similar to those of the RCA. In addition, the power-delay product (PDP) of the CSKA is smaller than those of the CSLA and PPA structures [19]. In addition, due to the small number of transistors, the CSKA benefits from relatively short wiring lengths as well as a regular and simple layout [18]. The comparatively lower speed of this adder structure, however, limits its use for high-speed applications.

In this paper, given the attractive features of the CSKA structure, we have focused on reducing its delay by modifying its implementation based on the static CMOS logic. The concentration on the static CMOS originates from the desire to have a reliably operating circuit under a wide range of supply voltages in highly scaled technologies [10]. The proposed modification increases the speed considerably while maintaining the low area and power consumption features of the CSKA. In addition, an adjustment of the structure, based on the variable latency technique, which in turn lowers the power consumption without considerably impacting the CSKA speed, is also presented. To the best of our knowledge, no work concentrating on design of CSKAs operating from the superthreshold region down to near-threshold region and also, the design of (hybrid) variable latency CSKA structures have been reported in the literature. Hence, the contributions of this paper can be summarized as follows.

1) Proposing a modified CSKA structure by combining the concatenation and the incrementation schemes to the conventional CSKA (Conv-CSKA) structure for enhancing the speed and energy efficiency of the adder. The modification provides us with the ability to use simpler carry skip logics based on the AOI/OAI compound gates instead of the multiplexer.
2) Providing a design strategy for constructing an efficient CSKA structure based on analytically expressions presented for the critical path delay.
3) Investigating the impact of voltage scaling on the efficiency of the proposed CSKA structure (from the nominal supply voltage to the near-threshold voltage).
4) Proposing a hybrid variable latency CSKA structure based on the extension of the suggested CSKA, by replacing some of the middle stages in its structure with a PPA, which is modified in this paper.

The rest of this paper is organized as follows. Section II discusses related work on modifying the CSKA structure for improving the speed as well as prior work that use variable latency structures for increasing the efficiency of adders at low supply voltages. In Section III, the Conv-CSKA with fixed stage size (FSS) and variable stage size (VSS) is explained, while Section IV describes the proposed static CSKA structure. The hybrid variable latency CSKA structure is suggested in Section V. The results of comparing the characteristics of the proposed structures with those of other adders are discussed in Section VI. Finally, the conclusion is drawn in Section VII.

II. Prior Work

Since the focus of this paper is on the CSKA structure, first the related work to this adder are reviewed and then the variable latency adder structures are discussed.

A. *Modifying CSKAs for Improving Speed*

The conventional structure of the CSKA consists of stages containing chain of full adders (FAs) (RCA block) and 2:1 multiplexer (carry skip logic). The RCA blocks are connected to each other through 2:1 multiplexers, which can be placed into one or more level structures [19]. The CSKA configuration (i.e., the number of the FAs per stage) has a great impact on the speed of this type of adder [23]. Many methods have been suggested for finding the optimum number of the FAs [18]–[26]. The techniques presented in [19]–[24] make use of VSSs to minimize the delay of adders based on a single-level carry skip logic. In [25], some methods to increase the speed of the multilevel CSKAs are proposed. The techniques, however, cause area and power increase considerably and less regular layout. The design of a static CMOS CSKA where the stages of the CSKA have a variable sizes was suggested in [18]. In addition, to lower the propagation delay of the adder, in each stage, the carry look-ahead logics were utilized. Again, it had a complex layout as well as large power consumption and area usage. In addition, the design approach, which was presented only for the 32-bit adder, was not general to be applied for structures with different bits lengths.

Alioto and Palumbo [19] propose a simple strategy for the design of a single-level CSKA. The method is based on the VSS technique where the near-optimal numbers of the FAs are determined based on the skip time (delay of the multiplexer), and the ripple time (the time required by a carry to ripple through a FA). The goal of this method is to decrease the critical path delay by considering a noninteger ratio of the skip time to the ripple time on contrary to most of the previous works, which considered an integer ratio [17], [20]. In all of the works reviewed so far, the focus was on the speed, while the power consumption and area usage of the CSKAs were

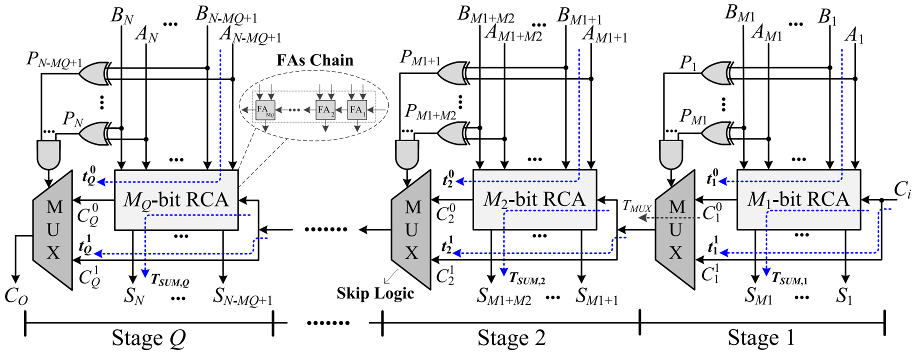


Fig. 1. Conventional structure of the CSKA [19].

not considered. Even for the speed, the delay of skip logics, which are based on multiplexers and form a large part of the adder critical path delay [19], has not been reduced.

B. Improving Efficiency of Adders at Low Supply Voltages

To improve the performance of the adder structures at low supply voltage levels, some methods have been proposed in [27]–[36]. In [27]–[29], an adaptive clock stretching operation has been suggested. The method is based on the observation that the critical paths in adder units are rarely activated. Therefore, the slack time between the critical paths and the off-critical paths may be used to reduce the supply voltage. Notice that the voltage reduction must not increase the delays of the noncritical timing paths to become larger than the period of the clock allowing us to keep the original clock frequency at a reduced supply voltage level. When the critical timing paths in the adder are activated, the structure uses two clock cycles to complete the operation. This way the power consumption reduces considerably at the cost of rather small throughput degradation. In [27], the efficiency of this method for reducing the power consumption of the RCA structure has been demonstrated.

The CSLA structure in [28] was enhanced to use adaptive clock stretching operation where the enhanced structure was called cascade CSLA (C^2SLA). Compared with the common CSLA structure, C^2SLA uses more and different sizes of RCA blocks. Since the slack time between the critical timing paths and the longest off-critical path was small, the supply voltage scaling, and hence, the power reduction were limited.

Finally, using the hybrid structure to improve the effectiveness of the adaptive clock stretching operation has been investigated in [31] and [33]. In the proposed hybrid structure, the KSA has been used in the middle part of the C^2SLA where this combination leads to the positive slack time increase. However, the C^2SLA and its hybrid version are not good candidates for low-power ALUs. This statement originates from the fact that due to the logic duplication in this type of adders, the power consumption and also the PDP are still high even at low supply voltages [33].

III. Conventional Carry Skip Adder

The structure of an N-bit Conv-CSKA, which is based on blocks of the RCA (RCA blocks), is shown in Fig. 1. In addition to the chain of FAs in each stage, there is a carry skip logic. For an RCA that contains N cascaded FAs, the worst propagation delay of the summation of two N-bit numbers, A and B, belongs to the case where all the FAs are in the propagation mode. It means that the worst case delay belongs to the case where

$$P_i = A_i \oplus B_i = 1 \quad \text{for } i = 1, \ldots, N$$

where P_i is the propagation signal related to A_i and B_i. This shows that the delay of the RCA is linearly related to N [1]. In the case, where a group of cascaded FAs are in the propagate mode, the carry output of the chain is equal to the carry input. In the CSKA, the carry skip logic detects this situation, and makes the carry ready for the next stage without waiting for the operation of the FA chain to be completed. The skip operation is performed using the gates and the multiplexer shown in the figure. Based on this explanation, the N FAs of the CSKA are grouped in Q stages. Each stage contains an RCA block with M_j FAs ($j = 1, \ldots, Q$) and a skip logic. In each stage, the inputs of the multiplexer (skip logic) are the carry input of the stage and the carry output of its RCA block (FA chain). In addition, the product of the propagation signals (P) of the stage is used as the selector signal of the multiplexer.

The CSKA may be implemented using FSS and VSS where the highest speed may be obtained for the VSS structure [19], [22]. Here, the stage size is the same as the RCA block size. In Sections III-A and III-B, these two different implementations of the CSKA adder are described in more detail.

A. Fixed Stage Size CSKA

By assuming that each stage of the CSKA contains M FAs, there are $Q = N/M$ stages where for the sake of simplicity, we assume Q is an integer. The input signals of the jth multiplexer are the carry output of the FAs chain in the jth stage denoted by C_j^0, the carry output of the previous stage (carry input of the jth stage) denoted by C_j^1 (Fig. 1).

The critical path of the CSKA contains three parts: 1) the path of the FA chain of the first stage whose delay is equal to $M \times T_{\text{CARRY}}$; 2) the path of the intermediate carry skip multiplexer whose delay is equal to the $(Q - 1) \times T_{\text{MUX}}$; and 3) the path of the FA chain in the last stage whose its delay is equal to the $(M - 1) \times T_{\text{CARRY}} + T_{\text{SUM}}$. Note that T_{CARRY},

T_{SUM}, and T_{MUX} are the propagation delays of the carry output of an FA, the sum output of an FA, and the output delay of a 2:1 multiplexer, respectively. Hence, the critical path delay of a FSS CSKA is formulated by

$$T_D = [M \times T_{\mathrm{CARRY}}] + \left[\left(\frac{N}{M} - 1\right) \times T_{\mathrm{MUX}}\right] + [(M-1) \times T_{\mathrm{CARRY}} + T_{\mathrm{SUM}}]. \qquad (1)$$

Based on (1), the optimum value of M (M_{opt}) that leads to optimum propagation delay may be calculated as $(0.5N\alpha)^{1/2}$ where α is equal to $T_{\mathrm{MUX}}/T_{\mathrm{CARRY}}$. Therefore, the optimum propagation delay ($T_{D,\mathrm{opt}}$) is obtained from

$$\begin{aligned} T_{D,\mathrm{opt}} &= 2\sqrt{2NT_{\mathrm{CARRY}}T_{\mathrm{MUX}}} + (T_{\mathrm{SUM}} - T_{\mathrm{CARRY}} - T_{\mathrm{MUX}}) \\ &= T_{\mathrm{SUM}} + (2\sqrt{2N\alpha} - 1 - \alpha) \times T_{\mathrm{CARRY}}. \end{aligned} \qquad (2)$$

Thus, the optimum delay of the FSS CSKA is almost proportional to the square root of the product of N and α [19].

B. Variable Stage Size CSKA

As mentioned before, by assigning variable sizes to the stages, the speed of the CSKA may be improved. The speed improvement in this type is achieved by lowering the delays of the first and third terms in (1). These delays are minimized by lowering sizes of first and last RCA blocks. For instance, the first RCA block size may be set to one, whereas sizes of the following blocks may increase. To determine the rate of increase, let us express the propagation delay of the C_j^1 (t_j^1) by

$$t_j^1 = \max\left(t_{j-1}^0, t_{j-1}^1\right) + T_{\mathrm{MUX}} \qquad (3)$$

where t_{j-1}^0 (t_{j-1}^1) shows the calculating delay of $C_{j-1}^0(C_{j-1}^1)$ signal in the $(j-1)$th stage. In a FSS CSKA, except in the first stage, t_j^0 is smaller than t_j^1. Hence, based on (3), the delay of t_{j-1}^0 may be increased from t_1^0 to t_{j-1}^1 without increasing the delay of C_j^1 signal. This means that one could increase the size of the $(j-1)$th stage (i.e., M_{j-1}) without increasing the propagation delay of the CSKA. Therefore, increasing the size of M_j for the jth stage should be bounded by

$$t_j^0 \le t_j^1 = t_1^0 + (j-1)T_{\mathrm{MUX}}. \qquad (4)$$

Since the last RCA block size also should be minimized, the increase in the stage size may not be continued to the last RCA block. Thus, we justify the decrease in the RCA block sizes toward the last stage. First, note that based on Fig. 1, the output of the jth stage is, in the worst case, accessible after $t_j^1 + T_{\mathrm{SUM},j}$. Assuming that the pth stage has the maximum RCA block size, we wish to keep the delay of the outputs of the following stages to be equal to the delay of the output of the pth stage. To keep the same worst case delay for the critical path, we should reduce the size of the following RCA blocks. For example, when $i \ge p$, for the $(i+1)$th stage, the output delay is $t_i^1 + T_{\mathrm{MUX}} + T_{\mathrm{SUM},i+1}$, where $T_{\mathrm{SUM},i+1}$ is the delay of the $(i+1)$th RCA block for calculating all of its sum outputs when its carry input is ready. Therefore, the size of the $(i+1)$th stage should be reduced to decrease $T_{\mathrm{SUM},i+1}$ preventing the increase in the worst case delay (T_D) of the adder. In other words, we eliminate the increase in the delay of the next stage due to the additional multiplexer by reducing the sum delay of the RCA block. This may be analytically expressed as

$$T_{\mathrm{SUM},i+1} \le T_{\mathrm{SUM},i} - T_{\mathrm{MUX}}; \quad \text{for } i \ge p. \qquad (5)$$

The trend of decreasing the stage size should be continued until we produce the required number of adder bits.

Note that, in this case, the size of the last RCA block may only be one (i.e., one FA). Hence, to reach the highest number of input bits under a constant propagation delay, both (4) and (5) should be satisfied. Having these constraints, we can minimize the delay of the CSKA for a given number of input bits to find the stages sizes for an optimal structure. In this optimal CSKA, the size of first p stages is increased, while the size of the last $(Q-p)$ stages is decreased. For this structure, the pth stage, which is called nucleus of the adder, has the maximum size [24].

Now, let us find the constraints used for determining the optimum structure in this case. As mentioned before, when the jth stage is not in the propagate mode, the carry output of the stage is C_j^0. In this case, the maximum of t_j^0 is equal to $M_j \times T_{\mathrm{CARRY}}$. To satisfy (4), we increase the size of the first p stages up to the nucleus using [19]

$$M_j \le M_1 + (j-1)\alpha; \quad \text{for } 1 \le j \le p. \qquad (6)$$

In addition, the maximum of $T_{\mathrm{SUM},i}$ is equal to $(M_i - 1) \times T_{\mathrm{CARRY}} + T_{\mathrm{SUM}}$. To satisfy (5), the size of the last $(Q-p)$ stages from the nucleus to the last stage should decrease based on [19]

$$M_i \ge M_Q + (Q-i)\alpha; \quad \text{for } p \le i \le Q. \qquad (7)$$

In the case, where α is an integer value, the exact sizes of stages for the optimal structure can be determined. Subsequently, the optimal values of M_1, M_Q, and Q as well as the delay of the optimal CSKA may be calculated [19]. In the case, where α is a noninteger value, one may realize only a near-optimal structure, as detailed in [19] and [21]. In this case, most of the time, by setting M_1 to 1 and using (6) and (7), the near-optimal structure is determined. It should be noted that, in practice, α is noninteger whose value is smaller than one. This is the case that has been studied in [19], where the estimation of the near-optimal propagation delay of the CSKA is given by [19]

$$T_{D,\mathrm{opt}} = \left(2\left\lceil\frac{\alpha}{2}\right\rceil - 1\right)T_{\mathrm{CARRY}} + \left(2\sqrt{\frac{N}{\alpha}} - 1\right)T_{\mathrm{MUX}} + T_{\mathrm{SUM}}. \qquad (8)$$

This equation may be written in a more general form by replacing T_{MUX} by T_{SKIP} to allow for other logic types instead of the multiplexer. For this form, α becomes equal to $T_{\mathrm{SKIP}}/T_{\mathrm{CARRY}}$. Finally, note that in real implementations, $T_{\mathrm{SKIP}} < T_{\mathrm{CARRY}}$, and hence, $\lceil\alpha/2\rceil$ becomes equal to one. Thus, (8) may be written as

$$T_{\mathrm{PD_{opt}}} = T_{\mathrm{CARRY}} + \left(2\sqrt{\frac{N}{\alpha}} - 1\right)T_{\mathrm{SKIP}} + T_{\mathrm{SUM}}. \qquad (9)$$

Note that, as (9) reveals that a large portion of the critical path delay is due to the carry skip logics.

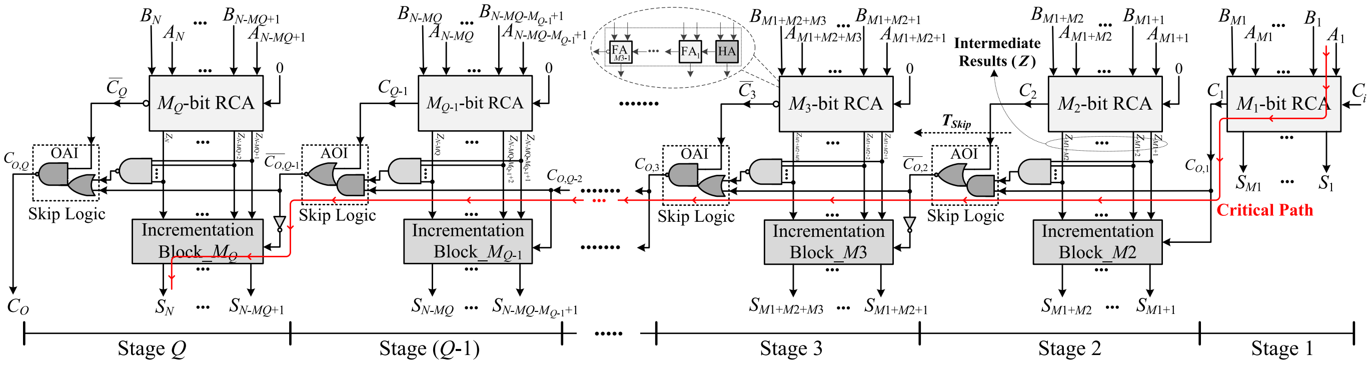


Fig. 2. Proposed CI-CSKA structure.

IV. Proposed CSKA Structure

Based on the discussion presented in Section III, it is concluded that by reducing the delay of the skip logic, one may lower the propagation delay of the CSKA significantly. Hence, in this paper, we present a modified CSKA structure that reduces this delay.

A. General Description of the Proposed Structure

The structure is based on combining the concatenation and the incrementation schemes [13] with the Conv-CSKA structure, and hence, is denoted by CI-CSKA. It provides us with the ability to use simpler carry skip logics. The logic replaces 2:1 multiplexers by AOI/OAI compound gates (Fig. 2). The gates, which consist of fewer transistors, have lower delay, area, and smaller power consumption compared with those of the 2:1 multiplexer [37]. Note that, in this structure, as the carry propagates through the skip logics, it becomes complemented. Therefore, at the output of the skip logic of even stages, the complement of the carry is generated. The structure has a considerable lower propagation delay with a slightly smaller area compared with those of the conventional one. Note that while the power consumptions of the AOI (or OAI) gate are smaller than that of the multiplexer, the power consumption of the proposed CI-CSKA is a little more than that of the conventional one. This is due to the increase in the number of the gates, which imposes a higher wiring capacitance (in the noncritical paths).

Now, we describe the internal structure of the proposed CI-CSKA shown in Fig. 2 in more detail. The adder contains two N bits inputs, A and B, and Q stages. Each stage consists of an RCA block with the size of M_j ($j = 1, \ldots, Q$). In this structure, the carry input of all the RCA blocks, except for the first block which is C_i, is zero (concatenation of the RCA blocks). Therefore, all the blocks execute their jobs simultaneously. In this structure, when the first block computes the summation of its corresponding input bits (i.e., $S_{M_1}, \ldots, S_1$), and C_1, the other blocks simultaneously compute the intermediate results [i.e., $\{Z_{K_j+M_j}, \ldots, Z_{K_j+2}, Z_{K_j+1}\}$ for $K_j = \sum_{r=1}^{j-1} M_r (j = 2, \ldots, Q)$], and also C_j signals. In the proposed structure, the first stage has only one block, which is RCA. The stages 2 to Q consist of two blocks of RCA and incrementation. The incrementation block uses the intermediate results generated by the RCA block and the carry output of the previous stage to calculate the final summation of the stage. The internal structure of the incrementation block, which contains a chain of half-adders (HAs), is shown in Fig. 3. In addition, note that, to reduce the delay considerably, for computing the carry output of the stage, the carry output of the incrementation block is not used. As shown in Fig. 2, the skip logic determines the carry output of the jth stage ($C_{O,j}$) based on the intermediate results of the jth stage and the carry output of the previous stage ($C_{O,j-1}$) as well as the carry output of the corresponding RCA block (C_j). When determining $C_{O,j}$, these cases may be encountered. When C_j is equal to one, $C_{O,j}$ will be one. On the other hand, when C_j is equal to zero, if the product of the intermediate results is one (zero), the value of $C_{O,j}$ will be the same as $C_{O,j-1}$ (zero).

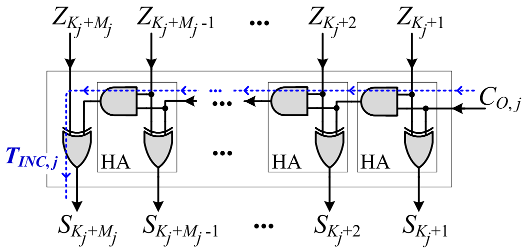


Fig. 3. Internal structure of the jth incrementation block, $K_j = \sum_{r=1}^{j-1} M_r$ ($j = 2, \ldots, Q$).

The reason for using both AOI and OAI compound gates as the skip logics is the inverting functions of these gates in standard cell libraries. This way the need for an inverter gate, which increases the power consumption and delay, is eliminated. As shown in Fig. 2, if an AOI is used as the skip logic, the next skip logic should use OAI gate. In addition, another point to mention is that the use of the proposed skipping structure in the Conv-CSKA structure increases the delay of the critical path considerably. This originates from the fact that, in the Conv-CSKA, the skip logic (AOI or OAI compound gates) is not able to bypass the zero carry input until the zero carry input propagates from the corresponding RCA block. To solve this problem, in the proposed structure, we have used an RCA block with a carry input of zero (using the concatenation approach). This way, since the RCA block of the stage does not need to wait for the carry output of the previous stage, the output carries of the blocks are calculated in parallel.

B. Area and Delay of the Proposed Structure

As mentioned before, the use of the static AOI and OAI gates (six transistors) compared with the static 2:1 multiplexer (12 transistors), leads to decreases in the area usage and delay of the skip logic [37], [38]. In addition, except for the first RCA block, the carry input for all other blocks is zero, and hence, for these blocks, the first adder cell in the RCA chain is a HA. This means that $(Q-1)$ FAs in the conventional structure are replaced with the same number of HAs in the suggested structure decreasing the area usage (Fig. 2). In addition, note that the proposed structure utilizes incrementation blocks that do not exist in the conventional one. These blocks, however, may be implemented with about the same logic gates (XOR and AND gates) as those used for generating the select signal of the multiplexer in the conventional structure. Therefore, the area usage of the proposed CI-CSKA structure is decreased compared with that of the conventional one.

The critical path of the proposed CI-CSKA structure, which contains three parts, is shown in Fig. 2. These parts include the chain of the FAs of the first stage, the path of the skip logics, and the incrementation block in the last stage. The delay of this path (T_D) may be expressed as

$$T_D = [M_1 T_{\text{CARRY}}] + [(Q-2)T_{\text{SKIP}}] \\ + [(M_Q - 1)T_{\text{AND}} + T_{\text{XOR}}] \quad (10)$$

where the three brackets correspond to the three parts mentioned above, respectively. Here, T_{AND} and T_{XOR} are the delays of the two inputs static AND and XOR gates, respectively. Note that, $[(M_j - 1)T_{\text{AND}} + T_{\text{XOR}}]$ shows the critical path delay of the jth incrementation block ($T_{\text{INC},j}$), which is shown in Fig. 3.

To calculate the delay of the skip logic, the average of the delays of the AOI and OAI gates, which are typically close to one another [35], is used. Thus, (10) may be modified to

$$T_D = [M_1 T_{\text{CARRY}}] + \left[(Q-2)\left(\frac{T_{\text{AOI}} + T_{\text{OAI}}}{2}\right)\right] \\ + [(M_Q - 1)T_{\text{AND}} + T_{\text{XOR}}] \quad (11)$$

where T_{AOI} and T_{OAI} are the delays of the static AOI and OAI gates, respectively.

The comparison of (1) and (11) indicates that the delay of the proposed structure is smaller than that of the conventional one. The First reason is that the delay of the skip logic is considerably smaller than that of the conventional structure while the number of the stages is about the same in both structures. Second, since T_{AND} and T_{XOR} are smaller than T_{CARRY} and T_{SUM}, the third additive term in (11) becomes smaller than the third term in (1) [37]. It should be noted that the delay reduction of the skip logic has the largest impact on the delay decrease of the whole structure.

C. Stage Sizes Consideration

Similar to the Conv-CSKA structure, the proposed CI-CSKA structure may be implemented with either FSS or VSS. Here, the stage size is the same as the RCA and incrementation blocks size. In the case of the FSS (FSS-CI-CSKA), there are $Q = N/M$ stages with the size of M. The optimum value of M, which may be obtained using (11), is given by

$$M_{\text{opt}} = \sqrt{\frac{N(T_{\text{AOI}} + T_{\text{OAI}})}{2(T_{\text{CARRY}} + T_{\text{AND}})}}. \quad (12)$$

In the case of the VSS (VSS-CI-CSKA), the sizes of the stages, which are M_1 to M_Q, are obtained using a method similar to the one discussed in Section III-B. For this structure, the new value for T_{SKIP} should be used, and hence, α becomes $(T_{\text{AOI}} + T_{\text{OAI}})$ / $(2 \times T_{\text{CARRY}})$. In particular, the following steps should be taken.

1) The size of the RCA block of the first stage is one.
2) From the second stage to the nucleus stage, the size of jth stage is determined based on the delay of the product of the sum of its RCA block and the delay of the carry output of the $(j-1)$th stage. Hence, based on the description given in Section III-B, the size of the RCA block of the jth stage should be as large as possible, while the delay of the product of the its output sum should be smaller than the delay of the carry output of the $(j-1)$th stage. Therefore, in this case, the sizes of the stages are either not changed or increased.
3) The increase in the size is continued until the summation of all the sizes up to this stage becomes larger than $N/2$. The last stage, which has the largest size, is considered as the nucleus (pth) stage. There are cases that we should consider the stage right before this stage as the nucleus stage (Step 5).
4) Starting from the stage $(p+1)$ to the last stage, the sizes of the stage i is determined based on the delay of the incrementation block of the ith and $(i-1)$th stages ($T_{\text{INC},i}$ and $T_{\text{INC},i-1}$, respectively), and the delay of the skip logic. In particular
$$T_{\text{INC},i} \le T_{\text{INC},i-1} - T_{\text{SKIP},i-1}; \quad \text{for} \quad i \ge p+1. \quad (13)$$
In this case, the size of the last stage is one, and its RCA block contains a HA.
5) Finally, note that, it is possible that the sum of all the stage sizes does not become equal to N. In the case, where the sum is smaller than N by d bits, we should add another stage with the size of d. The stage is placed close to the stage with the same size. In the case, where the sum is larger than N by d bits, the size of the stages should be revised (Step 3). For more details on how to revise the stage sizes, one may refer to [19].

Now, the procedure for determining the stage sizes is demonstrated for the 32-bit adder. It includes both the conventional and the proposed CI-CSKA structures. The number of stages and the corresponding size for each stage, which are given in Fig. 4, have been determined based on a 45-nm static CMOS technology [38]. The dashed and dotted lines in the plot indicate the rates of size increase and decrease. While the increase and decrease rates in the conventional structure are balanced, the decrease rate is more than the

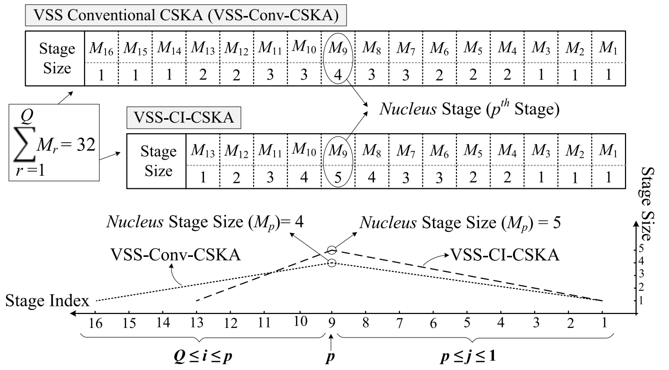

VSS Conventional CSKA (VSS-Conv-CSKA)
Stage Size
M_{16} M_{15} M_{14} M_{13} M_{12} M_{11} M_{10} M_9 M_8 M_7 M_6 M_5 M_4 M_3 M_2 M_1
1 1 1 2 2 3 3 4 3 3 2 2 2 1 1 1
$\sum_{r=1}^{Q} M_r = 32$
VSS-CI-CSKA
Nucleus Stage (p^{th} Stage)
Stage Size
M_{13} M_{12} M_{11} M_{10} M_9 M_8 M_7 M_6 M_5 M_4 M_3 M_2 M_1
1 2 3 4 5 4 3 3 2 2 1 1 1
Nucleus Stage Size (M_p)= 4
Nucleus Stage Size (M_p) = 5
VSS-Conv-CSKA
VSS-CI-CSKA
Stage Index
Stage Size
5 4 3 2 1
16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
$Q \leq i \leq p$
p
$p \leq j \leq 1$

Fig. 4. Sizes of the stages in the case of VSS for the proposed and conventional 32-bit CSKA structures in 45-nm static CMOS technology.

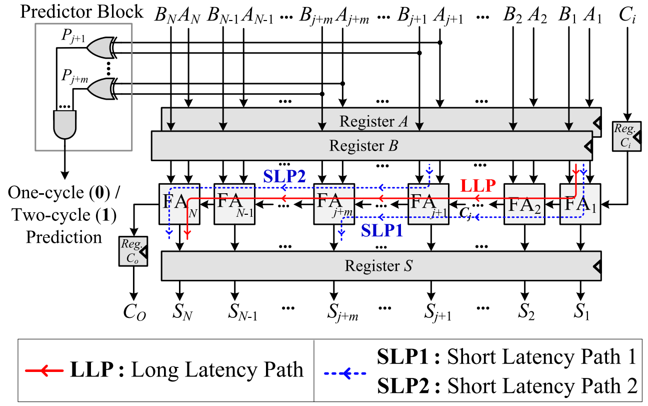

Predictor Block
$B_N A_N$ $B_{N-1} A_{N-1}$ ··· $B_{j+m} A_{j+m}$ ··· B_{j+1} A_{j+1} ··· B_2 A_2 B_1 A_1 C_i
P_{j+1}
P_{j+m}
Register A
Register B
Reg C_i
One-cycle (0) / Two-cycle (1) Prediction
SLP2
LLP
SLP1
FA_N FA_{N-1} FA_{j+m} FA_{j+1} FA_2 FA_1
C_j
Reg C_o
Register S
C_O S_N S_{N-1} ··· S_{j+m} ··· S_{j+1} ··· S_2 S_1
LLP : Long Latency Path
SLP1 : Short Latency Path 1
SLP2 : Short Latency Path 2

Fig. 5. Generic structure of variable latency adders based on RCA.

increase one in the case of the proposed structure. It originates from the fact that, in the Conv-CSKA structure, both of the stages size increase and decrease are determined based on the RCA block delay [according to (4) and (5)], while in the proposed CI-CSKA structure, the increase is determined based on the RCA block delay and the decrease is determined based on the incrementation block delay [according to (13)]. The imbalanced rates may yield a larger nucleus stage and smaller number of stages leading to a smaller propagation delay.

V. Proposed Hybrid Variable Latency CSKA

In this section, first, the structure of a generic variable latency adder, which may be used with the voltage scaling relying on adaptive clock stretching, is described. Then, a hybrid variable latency CSKA structure based on the CI-CSKA structure described in Section IV is proposed.

A. Variable Latency Adders Relying on Adaptive Clock Stretching

The basic idea behind variable latency adders is that the critical paths of the adders are activated rarely [33]. Hence, the supply voltage may be scaled down without decreasing the clock frequency. If the critical paths are not activated, one clock period is enough for completing the operation. In the cases, where the critical paths are activated, the structure allows two clock periods for finishing the operation. Hence, in this structure, the slack between the longest off-critical paths and the longest critical paths determines the maximum amount of the supply voltage scaling. Therefore, in the variable latency adders, for determining the critical paths activation, a predictor block, which works based on the inputs pattern, is required [28].

The concepts of the variable latency adders, adaptive clock stretching, and also supply voltage scaling in an N-bit RCA adder may be explained using Fig. 5. The predictor block consists of some XOR and AND gates that determines the product of the propagate signals of considered bit positions. Since the block has some area and power overheads, only few middle bits are used to predict the activation of the critical paths at price of prediction accuracy decrease [31], [33]. In Fig. 5, the input bits $(j+1)$th–$(j+m)$th have been exploited to predict the propagation of the carry output of the jth stage (FA) to the carry output of $(j+m)$th stage. For this configuration, the carry propagation path from the first stage to the Nth stage is the longest critical path (which is denoted by Long Latency Path (LLP), while the carry propagation path from first stage to the $(j+m)$th stage and the carry propagation path from $(j+1)$th stage to the Nth stage (which are denoted by Short Latency Path (SLP1) and SLP2, respectively) are the longest off-critical paths. It should be noted the paths that the predictor shows are (are not) active for a given set of inputs are considered as critical (off-critical) paths. Having the bits in the middle decreases the maximum of the off-critical paths [33]. The range of voltage scaling is determined by the slack time, which is defined by the delay difference between LLP and max(SLP1, SLP2). Since the activation probability of the critical paths is low $(<1/2^m)$, the clock stretching has a negligible impact on the throughput (e.g., for a 32-bit adder, m = 6–10 may be considered [33]). There are cases that the predictor mispredicts the critical path activation. By increasing m, the number of misprediction decreases at the price of increasing the longest off-critical path, and hence, limiting the range of the voltage scaling. Therefore, the predictor block size should be selected based on these tradeoffs.

B. Proposed Hybrid Variable Latency CSKA Structure

The basic idea behind using VSS CSKA structures was based on almost balancing the delays of paths such that the delay of the critical path is minimized compared with that of the FSS structure [21]. This deprives us from having the opportunity of using the slack time for the supply voltage scaling. To provide the variable latency feature for the VSS CSKA structure, we replace some of the middle stages in our proposed structure with a PPA modified in this paper. It should be noted that since the Conv-CSKA structure has a lower speed than that of the proposed one, in this section, we do not consider the conventional structure. The proposed hybrid variable latency CSKA structure is shown in Fig. 6 where an M_p-bit modified PPA is used for the pth stage (nucleus stage). Since the nucleus stage, which has the largest size (and delay) among the stages, is present in both SLP1 and SLP2, replacing it by the PPA reduces the delay of the longest

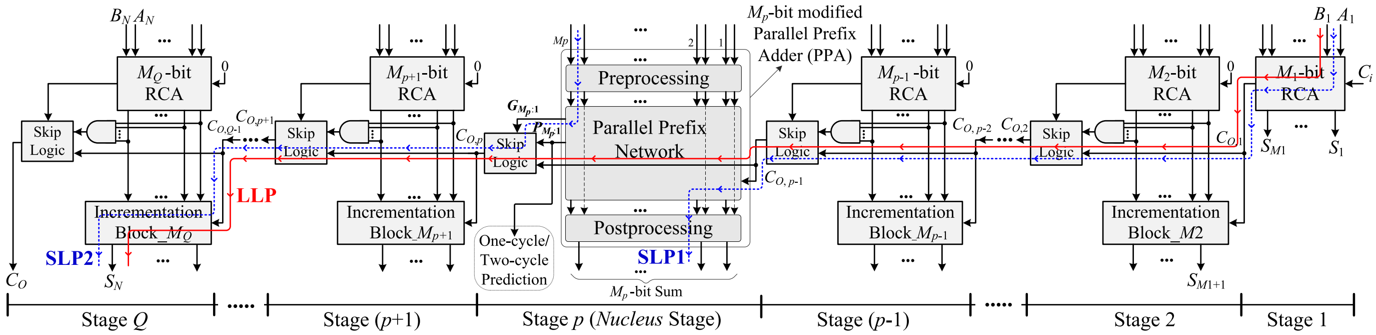


Fig. 6. Structure of the proposed hybrid variable latency CSKA.

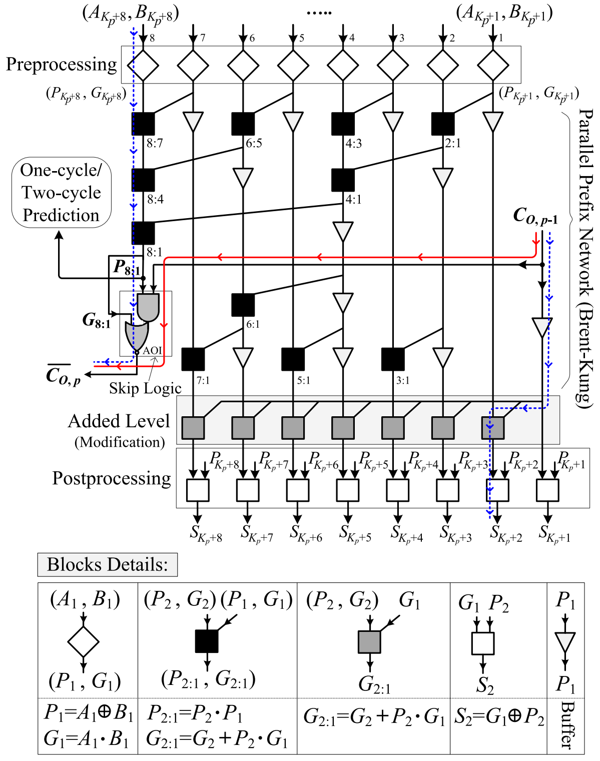


Fig. 7. Internal structure of the pth stage of the proposed hybrid variable latency CSKA. M_p is equal to 8 and $K_p = \sum_{r=1}^{p-1} M_r$.

off-critical paths. Thus, the use of the fast PPA helps increasing the available slack time in the variable latency structure. It should be mentioned that since the input bits of the PPA block are used in the predictor block, this block becomes parts of both SLP1 and SLP2.

In the proposed hybrid structure, the prefix network of the Brent–Kung adder [39] is used for constructing the nucleus stage (Fig. 7). One the advantages of the this adder compared with other prefix adders is that in this structure, using forward paths, the longest carry is calculated sooner compared with the intermediate carries, which are computed by backward paths. In addition, the fan-out of adder is less than other parallel adders, while the length of its wiring is smaller [14]. Finally, it has a simple and regular layout. The internal structure of the stage p, including the modified PPA and skip logic, is shown in Fig. 7. Note that, for this figure, the size of the PPA is assumed to be 8 (i.e., $M_p = 8$).

As shown in the figure, in the preprocessing level, the propagate signals (P_i) and generate signals (G_i) for the inputs are calculated. In the next level, using Brent–Kung parallel prefix network, the longest carry (i.e., $G_{8:1}$) of the prefix network along with $P_{8:1}$, which is the product of the all propagate signals of the inputs, are calculated sooner than other intermediate signals in this network. The signal $P_{8:1}$ is used in the skip logic to determine if the carry output of the previous stage (i.e., $C_{O,p-1}$) should be skipped or not. In addition, this signal is exploited as the predictor signal in the variable latency adder. It should be mentioned that all of these operations are performed in parallel with other stages. In the case, where $P_{8:1}$ is one, $C_{O,p-1}$ should skip this stage predicting that some critical paths are activated. On the other hand, when $P_{8:1}$ is zero, $C_{O,p}$ is equal to the $G_{8:1}$. In addition, no critical path will be activated in this case. After the parallel prefix network, the intermediate carries, which are functions of $C_{O,p-1}$ and intermediate signals, are computed (Fig. 7). Finally, in the postprocessing level, the output sums of this stage are calculated. It should be noted that this implementation is based on the similar ideas of the concatenation and incrementation concepts used in the CI-CSKA discussed in Section IV. It should be noted that the end part of the SPL1 path from $C_{O,p-1}$ to final summation results of the PPA block and the beginning part of the SPL2 paths from inputs of this block to $C_{O,p}$ belong to the PPA block (Fig. 7). In addition, similar to the proposed CI-CSKA structure, the first point of SPL1 is the first input bit of the first stage, and the last point of SPL2 is the last bit of the sum output of the incrementation block of the stage Q.

The steps for determining the sizes of the stages in the hybrid variable latency CSKA structure are similar to the ones discussed in Section IV. Since the PPA structure is more efficient when its size is equal to an integer power of two, we can select a larger size for the nucleus stage accordingly [14]. This implies that the third step discussed in that section is modified. The larger size (number of bits), compared with that of the nucleus stage in the original CI-CSKA structure, leads to the decrease in the number of stages as well smaller delays for SLP1 and SLP2. Thus, the slack time increases further.

VI. Results and Discussion

In this section, we assess the efficacies of the proposed structures by comparing their delays, powers, energies,

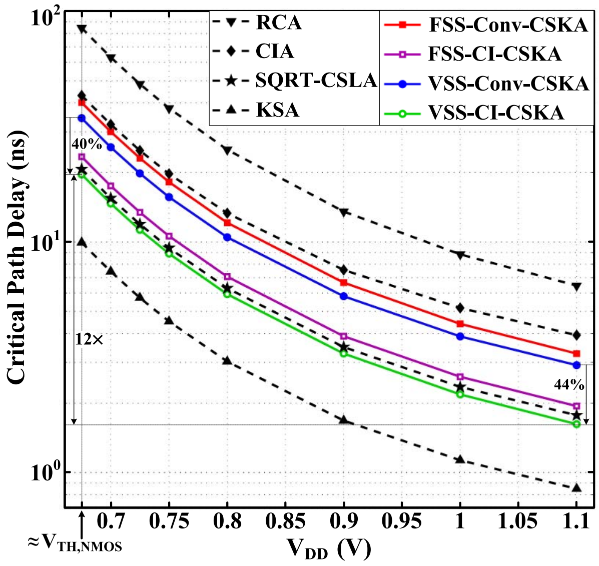


Fig. 8. Critical path delay of the adders versus the supply voltage.

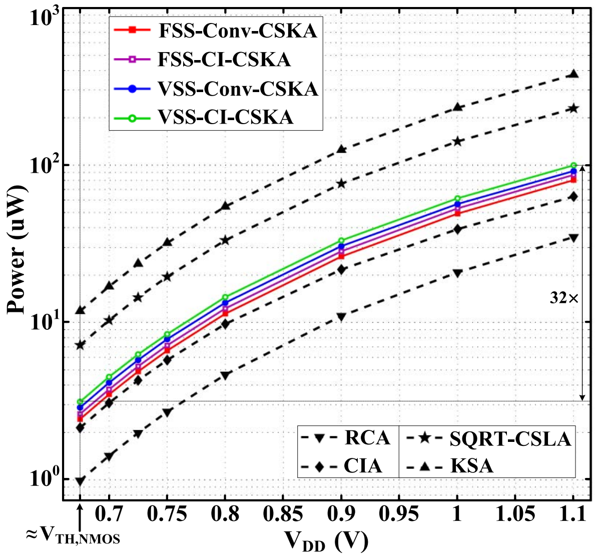


Fig. 9. Power consumption of the adders versus the supply voltage.

and areas with those of some other adders. All the adders considered here had the size of 32 bits and were designed and simulated using a 45-nm static CMOS technology [38]. The simulations were performed using HSPICE [40] in the room temperature of 25 °C. The nominal supply voltage of the technology was 1.1 V, and the threshold voltages of the nMOS and pMOS transistors were 0.677 and −0.622 V, respectively. It should be noted that, to extract the power consumption of the adders, 10 000 uniform random stimuli were injected to them. In addition, for each adder structure in each supply voltage level, the injection rate of the stimuli was chosen based on the maximum operating frequency of the structure. In the following Section VI-A and Section VI-B, we first concentrate on studying the effectiveness of the proposed CI-CSKA structure and then investigate the efficiency of the proposed hybrid variable latency structure based on the CI-CSKA.

A. CSKA Structures With Fixed and Variable Stage Sizes

In this section, both proposed and Conv-CSKA structures with FSS and VSS are considered. The optimum size of the stages for the FSS was 4 in the proposed (CI-CSKA) and Conv-CSKA adders. The sizes of the stages in the case of VSS were the same, as indicated in Fig. 4. The comparative study also included the RCA, CIA, square-root CSLA (SQRT-CSLA), and KSA. The results were obtained for a wide range of voltage levels from the nominal voltage (superthreshold) to nMOS threshold voltage ($V_{TH,nMOS}$) (near threshold).

The delays of the adders versus the supply voltage are plotted in Fig. 8. As the results show, the RCA (KSA) has the highest (lowest) delay due to its serial (parallel) structure under all the supply voltages. In addition, the smaller delay of SQRT-CSLA compared with that of CIA is due to the logic duplication. In addition, as was expected the CSKA structures have significantly smaller delays compared with that of the RCA. In addition, their delays are less than that of the CIA. As is observed from this figure, compared with the Conv-CSKA, our proposed structures reduce the delays further such that in the case of VSS, the delay becomes even lower than that of SQRT-CSLA. For the supply voltages considered here, the delay reductions of the CI-CSKA compared with those of the Conv-CSKA in the case of the FSS (VSS) were in the range of 40%–42% (40%–44%). In addition, using VSS scheme in the CI-CSKA (Conv-CSKA), provides us with the delay reductions of 15%–17% (11%–14%). Finally, the results indicate that reducing the supply voltage from 1.1 V to the nMOS threshold voltage causes an about 12 fold increase in the delay for all the adders.

The power consumptions of the adders versus the supply voltage are shown in Fig. 9. The results reveal that the smallest power consumption belongs to the RCA, while the KSA structure consumes the highest power owing to its parallel structure. The power consumption of the CIA is more than the RCA while it is smaller than that of the SQRT-CSLA. The reason for the high power of the SQRT-CSLA is its logic duplication. The power consumptions of the conventional and proposed CI-CSKA structures are slightly more than that of the CIA. The powers of these adders increase further using VSS scheme where the number of stages is larger. As mentioned before, the power of the CI-CSKA structure is little more than that of the conventional one. For example, the power of VSS-CI-CSKA is ∼5%–7% larger than that of the VSS-Conv-CSKA. It should be pointed out that while the delay of the VSS-CI-CSKA was smaller than delay of the SQRT-CSLA, its power is also considerably smaller than that of the SQRT-CSLA. Finally, the results reveal, on average, a 32× reduction in the power consumption of the adders when scaling the supply voltage from 1.1 V to the nMOS threshold voltage.

Fig. 10 shows the PDP of the adders for different supply voltages. The proposed CI-CSKA has the best PDP compared with those of the other structures in the supply voltage range considered in this paper. The highest PDP (with ∼2.5× more than that of the CI-CSKA structure)

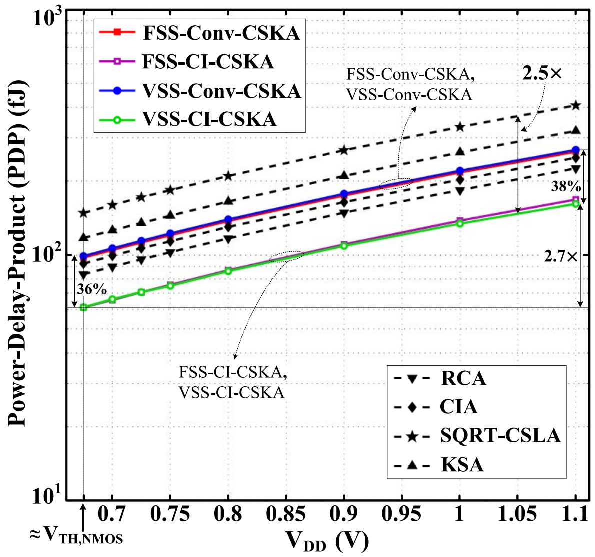


Fig. 10. PDP of the adders versus the supply voltage.

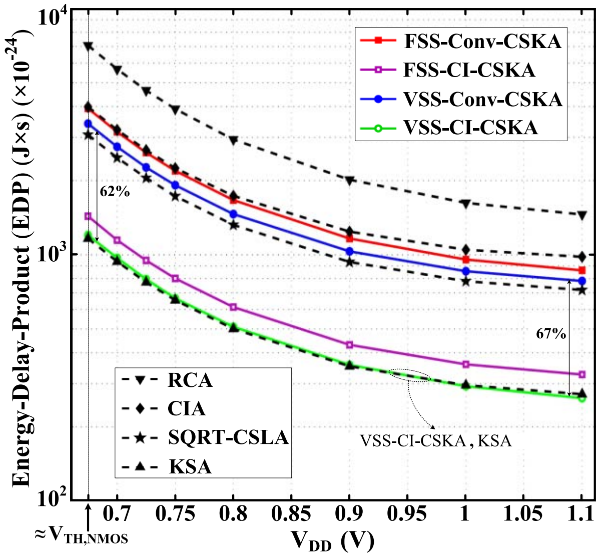


Fig. 11. EDP of the adders versus the supply voltage.

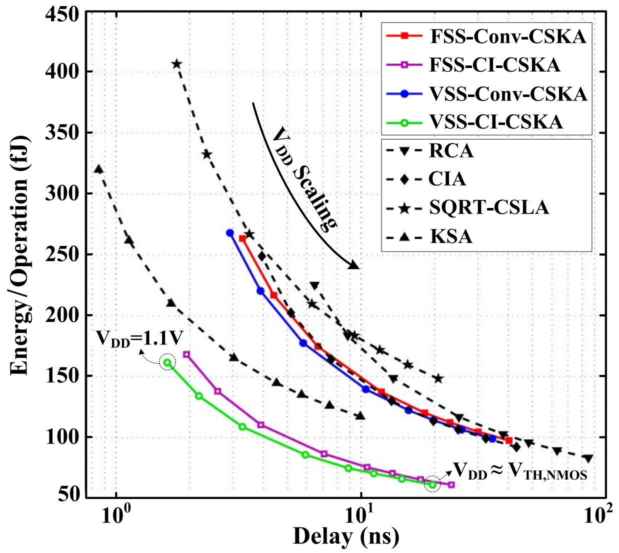


Fig. 12. Energy–delay Pareto-optimal curves for different adders.

TABLE I
AREA USAGES AND NUMBER OF TRANSISTORS OF THE ADDERS

Adder Structure	Area (μm^2)	# of Transistors
RCA	151.3	896
CIA	230.0	1402
SQRT-CSLA	357.4	2096
KSA	403.2	2010
FSS-Conv-CSKA	254.2	1456
FSS-CI-CSKA	246.3	1370
VSS-Conv-CSKA	253.1	1464
VSS-CI-CSKA	241.5	1332

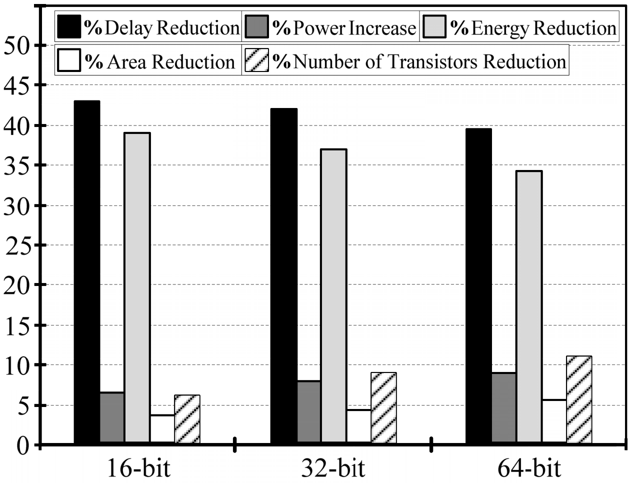
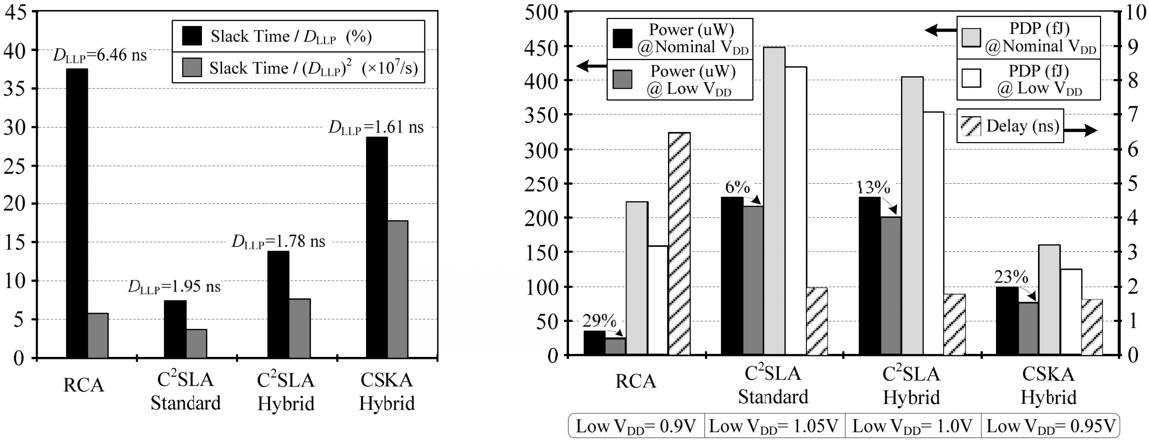


Fig. 13. Changes of delay, power, energy, area, and number of transistors for the proposed VSS-CI-CSKA structure compared with those of the VSS-Conv-CSKA structure in the case of 16-, 32-, and 64-bit length.

corresponds to SQRT-CSLA. After SQRT-CSLA, KSA has the highest PDP. The results show that the PDP of the proposed CI-CSKA structure is 35%–38% less than that of the Conv-CSKA structure. In addition, in both the conventional and the proposed structures, the PDP of FSS and VSS are about the same.

The values of the energy–delay product (EDP) of the adders versus the supply voltage are plotted in Fig. 11. As the results reveal, the RCA has the largest EDP due to its lowest speed. The EDP of the proposed VSS-CI-CSKA is almost the same as that of the KSA structure. The lower value of the EDP for the proposed CI-CSKA originates from the smaller power consumption as well as higher speed of the structure. Furthermore, the VSS-CI-CSKA has smaller area and power consumption compared with those of the KSA. Finally, to demonstrate the tradeoffs between the delay and the energy for each adder structure, the energy–delay Pareto-optimal curves are plotted in Fig. 12, which suggests the proposed VSS-CI-CSKA structure as the better adder.

Table I reports the area usages and number of transistors for each adder structure. The RCA has the smallest area, while the KSA has the highest area. The next largest adder



(a) (b)

Fig. 14. (a) Ratio of the slack time to D_{LLP}, and the ratio of the slack time to $(D_{\text{LLP}})^2$ for the four studied variable latency adders. (b) Power consumption and PDP at the nominal and low V_{DD} for the adders. (Nominal V_{DD} is 1.1 V for all the structures.)

is SQRT-CSLA. All four CSKA structures and the CIA have about the same area. In addition, as stated before, the proposed CI-CSKA structure slightly decreases the area compared with that of the conventional one. In addition, the number of transistors of the proposed CI-CSKA structure is smaller than that of the Conv-CSKA structure in both FSS and VSS styles. It should be noted that the lowest PDP (energy) and low area of the proposed CI-CSKA structure were the motivation behind extending the structure for variable latency applications.

Finally, to investigate the effect of bit length on the efficiency of the proposed CI-CSKA structure, we compare the changes [(Value$_{\text{Conventional}}$–Value$_{\text{Proposed}}$)/Value$_{\text{Conventional}}$] of the delay, power, energy, and area of the CI-CSKA and Conv-CSKA structures for 16-, 32-, and 64-bit. For the sake of space, we present the average results of different supply voltage levels. In addition, for the same reason, we limit the comparison with VSS structures because the VSS-CI-CSKA is the more efficient structure among the considered CSKA structures. Furthermore, as mentioned before, the proposed hybrid variable latency CSKA is constructed based on the VSS-CI-CSKA. The results are presented in Fig. 13. The figure reveals that the delay reduction and energy saving slightly decreases and the power increase enlarges a bit with increasing the length. In addition, the increase in the bit length improves the area and number of transistors of the proposed VSS-CI-CSKA compared with those of the VSS-Conv-CSKA. In Section VI-B, we present the results for the variable latency adders.

B. Variable Latency Adders

In this part, the performance of the proposed hybrid variable latency CSKA structure is compared with those of some other variable latency adders, including RCA [27], C^2SLA [29], and hybrid C^2SLA [31], [33]. In the proposed 32-bit hybrid structure, an 8-bit modified PPA block was used in the nucleus stage (Fig. 7). The sizes of the stages from LSB to MSB were {1, 1, 1, 2, 2, 3, 3, 8, 3, 3, 2, 2, 1} where the prediction was performed using the input bits of 14–21. In the case of the RCA, eight intermediate bits of 13–20 were exploited for the prediction block. The C^2SLA is an extension of the SQRT-CSLA where the variable latency feature is achieved by increasing the number of stages as well as having different sizes for their RCA blocks. In the C^2SLA, cascading was done by dividing the 32 bits into groups of {2, 2, 3, 4, 5, 2, 2, 3, 4, 5} where the partial sum was computed in parallel for $C_i = 0$ as well as $C_i = 1$ using the RCA. Next, the multiplexers selected the appropriate sum based on the actual carry. In this structure, seven intermediate bits of 17–23 were used in the prediction block. In the hybrid C^2SLA, nine intermediate results were calculated using KSA where the details may be found in [33].

As a measure of the ability of a structure in using the variable latency feature for reducing the power consumption, one may use the ratio of the slack time to the delay of the adder (which is equal to the delay of the LLP denoted by D_{LLP}). The ratios for the four adder structures are shown in Fig. 14(a). The figure also contains the ratio of the slack time to $(D_{\text{LLP}})^2$ to include the speed of the adder in the figure of merit for the efficacy of the structure in reducing the power using the variable latency scheme. Note that, the details for the LLP, SLP1, and SPL2 in the C^2SLA and hybrid C^2SLA may be found in [31]. As the results show, the RCA can obtain the highest improvement using the adaptive clock stretching technique. This adder, however, has the worst delay among the four adder structures. The next highest improvement belongs to the proposed hybrid CSKA whose delay is in the order of the other two adders. The observation indicates that the proposed hybrid CSKA may be considered as a fast adder structure for low-power applications. To further clarify this, the results for the power and PDP at both the nominal and the reduced supply voltages for each adder structure are plotted in Fig. 14(b). The amounts of power and energy savings are functions of the supply voltage deduction, which is determined by the slack time. Since the slack times are different for the structures, the amounts of the voltage reduction are different too. The power and PDP at the nominal voltage are for the corresponding baseline structure of each adder (no variable latency structure) while at the reduced voltage, the variable latency structure is considered.

The results show that the highest power (energy) reduction of ∼29% belongs to the RCA structure, which has due the highest slack time. In this case, the supply voltage reduction was 0.2 V. In the case of the standard C^{2}SLA, since the slack time was small, the voltage reduction was ∼0.05 V, which led to a power reduction of 6%. For the hybrid C^{2}SLA, the slack time was higher than that of the standard C^{2}SLA and hence the voltage reduction of ∼0.1 V became possible. This provided a higher power reduction (∼13%). Finally, the proposed hybrid CSKA had a larger slack compared with that of the hybrid C^{2}SLA, and hence, the voltage reduction of ∼0.15 V was possible. This provided the structure with a power reduction of ∼23% (larger than those of the C^{2}SLA structures). The very low delay of the hybrid variable latency CSKA along with its lower power consumption result in the minimum PDP for this structure. In addition, the higher PDP of the C^{2}SLA structures is due to their high-power consumptions.

VII. Conclusion

In this paper, a static CMOS CSKA structure called CI-CSKA was proposed, which exhibits a higher speed and lower energy consumption compared with those of the conventional one. The speed enhancement was achieved by modifying the structure through the concatenation and incrementation techniques. In addition, AOI and OAI compound gates were exploited for the carry skip logics. The efficiency of the proposed structure for both FSS and VSS was studied by comparing its power and delay with those of the Conv-CSKA, RCA, CIA, SQRT-CSLA, and KSA structures. The results revealed considerably lower PDP for the VSS implementation of the CI-CSKA structure over a wide range of voltage from super-threshold to near threshold. The results also suggested the CI-CSKA structure as a very good adder for the applications where both the speed and energy consumption are critical. In addition, a hybrid variable latency extension of the structure was proposed. It exploited a modified parallel adder structure at the middle stage for increasing the slack time, which provided us with the opportunity for lowering the energy consumption by reducing the supply voltage. The efficacy of this structure was compared versus those of the variable latency RCA, C^{2}SLA, and hybrid C^{2}SLA structures. Again, the suggested structure showed the lowest delay and PDP making itself as a better candidate for high-speed low-energy applications.

References

[1] I. Koren, *Computer Arithmetic Algorithms*, 2nd ed. Natick, MA, USA: A K Peters, Ltd., 2002.

[2] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy–delay optimization of 64-bit carry-lookahead adders with a 240 ps 90 nm CMOS design example," *IEEE J. Solid-State Circuits*, vol. 44, no. 2, pp. 569–583, Feb. 2009.

[3] S. K. Mathew, M. A. Anders, B. Bloechel, T. Nguyen, R. K. Krishnamurthy, and S. Borkar, "A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 44–51, Jan. 2005.

[4] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 6, pp. 754–758, Jun. 2005.

[5] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2, pp. 371–375, Feb. 2012.

[6] M. Vratonjic, B. R. Zeydel, and V. G. Oklobdzija, "Low- and ultra low-power arithmetic units: Design and comparison," in *Proc. IEEE Int. Conf. Comput. Design, VLSI Comput. Process. (ICCD)*, Oct. 2005, pp. 249–252.

[7] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 10, pp. 689–702, Oct. 1996.

[8] Y. He and C.-H. Chang, "A power-delay efficient hybrid carry-lookahead/carry-select based redundant binary to two's complement converter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 1, pp. 336–346, Feb. 2008.

[9] C.-H. Chang, J. Gu, and M. Zhang, "A review of 0.18 μm full adder performances for tree structured arithmetic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 6, pp. 686–695, Jun. 2005.

[10] D. Markovic, C. C. Wang, L. P. Alarcon, T.-T. Liu, and J. M. Rabaey, "Ultralow-power design in near-threshold region," *Proc. IEEE*, vol. 98, no. 2, pp. 237–252, Feb. 2010.

[11] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," *Proc. IEEE*, vol. 98, no. 2, pp. 253–266, Feb. 2010.

[12] S. Jain *et al.*, "A 280 mV-to-1.2 V wide-operating-range IA-32 processor in 32 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, Feb. 2012, pp. 66–68.

[13] R. Zimmermann, "Binary adder architectures for cell-based VLSI and their synthesis," Ph.D. dissertation, Dept. Inf. Technol. Elect. Eng., Swiss Federal Inst. Technol. (ETH), Zürich, Switzerland, 1998.

[14] D. Harris, "A taxonomy of parallel prefix networks," in *Proc. IEEE Conf. Rec. 37th Asilomar Conf. Signals, Syst., Comput.*, vol. 2. Nov. 2003, pp. 2213–2217.

[15] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Trans. Comput.*, vol. C-22, no. 8, pp. 786–793, Aug. 1973.

[16] V. G. Oklobdzija, B. R. Zeydel, H. Dao, S. Mathew, and R. Krishnamurthy, "Energy-delay estimation technique for high-performance microprocessor VLSI adders," in *Proc. 16th IEEE Symp. Comput. Arithmetic*, Jun. 2003, pp. 272–279.

[17] M. Lehman and N. Burla, "Skip techniques for high-speed carry-propagation in binary arithmetic units," *IRE Trans. Electron. Comput.*, vol. EC-10, no. 4, pp. 691–698, Dec. 1961.

[18] K. Chirca *et al.*, "A static low-power, high-performance 32-bit carry skip adder," in *Proc. Euromicro Symp. Digit. Syst. Design (DSD)*, Aug./Sep. 2004, pp. 615–619.

[19] M. Alioto and G. Palumbo, "A simple strategy for optimized design of one-level carry-skip adders," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 1, pp. 141–148, Jan. 2003.

[20] S. Majerski, "On determination of optimal distributions of carry skips in adders," *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 1, pp. 45–58, Feb. 1967.

[21] A. Guyot, B. Hochet, and J.-M. Muller, "A way to build efficient carry-skip adders," *IEEE Trans. Comput.*, vol. C-36, no. 10, pp. 1144–1152, Oct. 1987.

[22] S. Turrini, "Optimal group distribution in carry-skip adders," in *Proc. 9th IEEE Symp. Comput. Arithmetic*, Sep. 1989, pp. 96–103.

[23] P. K. Chan, M. D. F. Schlag, C. D. Thomborson, and V. G. Oklobdzija, "Delay optimization of carry-skip adders and block carry-lookahead adders using multidimensional dynamic programming," *IEEE Trans. Comput.*, vol. 41, no. 8, pp. 920–930, Aug. 1992.

[24] V. Kantabutra, "Designing optimum one-level carry-skip adders," *IEEE Trans. Comput.*, vol. 42, no. 6, pp. 759–764, Jun. 1993.

[25] V. Kantabutra, "Accelerated two-level carry-skip adders—A type of very fast adders," *IEEE Trans. Comput.*, vol. 42, no. 11, pp. 1389–1393, Nov. 1993.

[26] S. Jia *et al.*, "Static CMOS implementation of logarithmic skip adder," in *Proc. IEEE Conf. Electron Devices Solid-State Circuits*, Dec. 2003, pp. 509–512.

[27] H. Suzuki, W. Jeong, and K. Roy, "Low power adder with adaptive supply voltage," in *Proc. 21st Int. Conf. Comput. Design*, Oct. 2003, pp. 103–106.

[28] H. Suzuki, W. Jeong, and K. Roy, "Low-power carry-select adder using adaptive supply voltage based on input vector patterns," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 2004, pp. 313–318.

[29] Y. Chen, H. Li, K. Roy, and C.-K. Koh, "Cascaded carry-select adder (C^2SA): A new structure for low-power CSA design," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 2005, pp. 115–118.

[30] Y. Chen, H. Li, J. Li, and C.-K. Koh, "Variable-latency adder (VL-adder): New arithmetic circuit design practice to overcome NBTI," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 2007, pp. 195–200.

[31] S. Ghosh and K. Roy, "Exploring high-speed low-power hybrid arithmetic units at scaled supply and adaptive clock-stretching," in *Proc. Asia South Pacific Design Autom. Conf. (ASPDAC)*, Mar. 2008, pp. 635–640.

[32] Y. Chen *et al.*, "Variable-latency adder (VL-adder) designs for low power and NBTI tolerance," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 11, pp. 1621–1624, Nov. 2010.

[33] S. Ghosh, D. Mohapatra, G. Karakonstantis, and K. Roy, "Voltage scalable high-speed robust hybrid arithmetic units using adaptive clocking," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 9, pp. 1301–1309, Sep. 2010.

[34] Y. Liu, Y. Sun, Y. Zhu, and H. Yang, "Design methodology of variable latency adders with multistage function speculation," in *Proc. IEEE 11th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2010, pp. 824–830.

[35] Y.-S. Su, D.-C. Wang, S.-C. Chang, and M. Marek-Sadowska, "Performance optimization using variable-latency design style," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 10, pp. 1874–1883, Oct. 2011.

[36] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in *Proc. Design, Autom., Test Eur. Conf. Exhibit. (DATE)*, Mar. 2012, pp. 1257–1262.

[37] J. M. Rabaey, A. Chandrakasa, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.

[38] *NanGate 45 nm Open Cell Library*. [Online]. Available: http://www.nangate.com/, accessed Dec. 2010.

[39] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," *IEEE Trans. Comput.*, vol. C-31, no. 3, pp. 260–264, Mar. 1982.

[40] *Synopsys HSPICE*. [Online]. Available: http://www.synopsys.com, accessed Sep. 2011.

Milad Bahadori received the M.Sc. degree in electrical and electronic engineering from the Sharif University of Technology, Tehran, Iran, in 2011. He is currently pursuing the Ph.D. degree with the University of Tehran, Tehran, Iran.

He was a Research Assistant with the Sharif Integrated Circuits Design Center, Sharif University of Technology, from 2009 to 2012, where he was involved in digital systems design. He joined the Low-Power High-Performance Nanosystems Laboratory, University of Tehran, in 2012, as a Research Assistant. His current research interests include low-power high-performance VLSI design, reliability in nanoscale design, near-threshold computing, high-performance low-power arithmetic circuits design, and cryptographic systems design.

Mehdi Kamal received the B.Sc. degree from the Iran University of Science and Technology, Tehran, Iran, in 2005, the M.Sc. degree from the Sharif University of Technology, Tehran, in 2007, and the Ph.D. degree from the University of Tehran, Tehran, in 2013, all in computer engineering.

He is currently a Research Associate with the Low-Power High-Performance Nanosystem Laboratory, School of Electrical and Computer Engineering, University of Tehran. His current research interests include reliability in nanoscale design, application-specific instruction set processor design, hardware/software co-design, and low-power design.

Ali Afzali-Kusha (SM'06) received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 1988, the M.Sc. degree from the University of Pittsburgh, Pittsburgh, PA, USA, in 1991, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 1994, all in electrical engineering.

He was a Post-Doctoral Fellow with the University of Michigan from 1994 to 1995. He has been with the University of Tehran, since 1995, where he is currently a Professor of the School of Electrical and Computer Engineering and the Director of the Low-Power High-Performance Nanosystems Laboratory. He was a Research Fellow with the University of Toronto, Toronto, ON, Canada, and the University of Waterloo, Waterloo, ON, Canada, in 1998 and 1999, respectively. His current research interests include low-power high-performance design methodologies from the physical design level to the system level for nanoelectronics era.

Massoud Pedram (F'01) received the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 1991.

He is currently the Stephen and Etta Varra Professor with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA. He holds 10 U.S. patents and has authored four books, 12 book chapters, and over 140 archival and 350 conference papers. His current research interests include low-power electronics, energy-efficient processing, and cloud computing to photovoltaic cell power generation, energy storage, and power conversion, and RT level optimization of VLSI circuits to synthesis and physical design of quantum circuits.

Prof. Pedram and his students have received six conference and two IEEE TRANSACTIONS Best Paper Awards for the research. He was a recipient of the 1996 Presidential Early Career Award for Scientists and Engineers and an ACM Distinguished Scientist, and currently serves as the Editor-in-Chief of the *ACM Transactions on Design Automation of Electronic Systems*. He has served on the Technical Program Committee of a number of premiere conferences in his field. He was the Founding Technical Program Co-Chair of the 1996 International Symposium on Low-Power Electronics and Design and the Technical Program Chair of the 2002 International Symposium on Physical Design.