ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Novel High-Speed Carry Skip Adder with AOI and OAI Logic Using Verilog HDL

Adoni Shiva Pradeep
M.Tech,
Dr.K.V.Subba Reddy Institute of Technology.

S.Feroz Shah Ahmed, M.Tech
Assistant Professor,
Dr.K.V.Subba Reddy Institute of Technology.

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.

Index Terms:
Carry skip adder (CSKA), high performance, hybrid variable latency adders.

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 sub threshold 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 super threshold, near-threshold, or sub threshold regions. Working in the super threshold region provides us with lower delay and higher switching and leakage powers compared with the near/sub threshold regions. In the sub threshold 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 sub threshold current causes a large delay for the circuits operating in the sub threshold 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 sub threshold one, because it results in lower delay compared with the sub threshold region and significantly lowers switching and leakage powers compared with the super threshold 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 sub threshold 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 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 super threshold 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 theefficiency of the proposed CSKA structure (from thenominal supply voltage to the near-threshold voltage).
4) Proposing a hybrid variable latency CSKA structurebased on the extension of the suggested CSKA, byreplacing some of the middle stages in its structure witha PPA, which is modified in this paper.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 thevariable latency adder structures are discussed.

A. Modifying CSKAs for Improving Speed:

The conventional structure of the CSKA consists of stagescontaining chain of full adders (FAs) (RCA block) and2:1 multiplexer (carry skip logic). The RCA blocks areconnected to each other through 2:1 multiplexers, which canbe placed into one or more level structures [19]. The CSKAconfiguration (i.e., the number of the FAs per stage) has a greatimpact on the speed of this type of adder [23]. Many methodshave been suggested for finding the optimum number of theFAs [18]–[26]. The techniques presented in [19]–[24] makeuse of VSSs to minimize the delay of adders based on a singlelevelcarry skip logic. In [25], some methods to increase thespeed of the multilevel CSKAs are proposed. The techniques,however, cause area and power increase considerably andless regular layout.

The design of a static CMOS CSKAwhere the stages of the CSKA have a variable sizes wassuggested in [18]. In addition, to lower the propagation delayof the adder, in each stage, the carry look-ahead logics wereutilized. Again, it had a complex layout as well as large powerconsumption and area usage. In addition, the design approach,which was presented only for the 32-bit adder, was not generalto be applied for structures with different bits lengths.Alioto and Palumbo [19] propose a simple strategy for thedesign of a single-level CSKA. The method is based on theVSS technique where the near-optimal numbers of the FAs aredetermined based on the skip time (delay of the multiplexer),and the ripple time (the time required by a carry to ripplethrough a FA). The goal of this method is to decrease thecritical path delay by considering a noninteger ratio of theskip time to the ripple time on contrary to most of the previousworks, which considered an integer ratio [17], [20]. In all ofthe works reviewed so far, the focus was on the speed, whilethe power consumption and area usage of the CSKAs were 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.

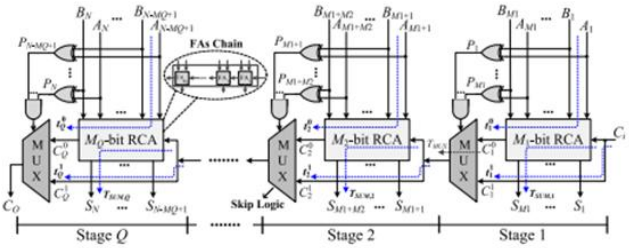

Fig. 1.Conventional structure of the CSKA [19].

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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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

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

where Pi is the propagation signal related to Ai and Bi .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 Mj FAs (j = 1,..., Q) and a skip logic. In each stage, theinputs of the multiplexer (skip logic) are the carry input ofthe stage and the carry output of its RCA block (FA chain).

In addition, the product of the propagation signals (P) of thestage is used as the selector signal of the multiplexer.The CSKA may be implemented using FSS and VSSwhere the highest speed may be obtained for the VSSstructure [19], [22]. Here, the stage size is the same as theRCA block size. In Sections III-A and III-B, thesetwo different implementations of the CSKA adder aredescribed 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 thejth multiplexer are the carry output of the FAs chain in thejth stage denoted by C0j , the carry output of the previous stage(carry input of the jth stage) denoted by C1j (Fig. 1).The critical path of the CSKA contains three parts:

1) Thepath of the FA chain of the first stage whose delay is equalto M × TCARRY;
2) 2) the path of the intermediate carry skipmultiplexer whose delay is equal to the (Q – 1) × TMUX; and
3) 3) the path of the FA chain in the last stage whose its delayis equal to the(M −1) × TCARRY + TSUM. Note that TCARRY, TSUM, and TMUX 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

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

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_{MUX}/T_{CARRY}. Therefore, the optimum propagation delay (TD,opt) is obtained from

$$T_{D,\text{opt}} = 2\sqrt{2NT_{\text{CARRY}}T_{\text{MUX}}} + (T_{\text{SUM}} - T_{\text{CARRY}} - T_{\text{MUX}}) = T_{\text{SUM}} + (2\sqrt{2N\alpha} - 1 - \alpha) \times T_{\text{CARRY}}. \quad (2)$$

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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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^1_j (t^1_j) by

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

where t^0_{j-1} (t^1_{j-1}) shows the calculating delay of C^0_{j-1}(C^1_{j-1}) signal in the (j − 1)th stage.In a FSS CSKA, except in the first stage, t^0_j is smaller than t^1_j . Hence, based on (3), the delay of t^0_{j-1} may be increased from t^0_1 to t^1_{j-1} without increasing the delay of C^1_j signal. This means that one could increase the size of the (j − 1)th stage (i.e., Mj−1) without increasing the propagationdelay of the CSKA. Therefore, increasing the size of Mj 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 lastRCA 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^1_j + TSUM,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 ≥ p, for the (i +1)th stage, the output delay is t^1_i + TMUX + TSUM,i+1, where TSUM,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 TSUM,i+1 preventing the increase in the worst case delay (TD) 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^0_j. In this case, the maximum of t^0_j is equal to Mj × TCARRY. To satisfy (4), we increase the size of thefirst 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 TSUM,i is equal to (Mi − 1) × TCARRY + TSUM. 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 M1, MQ , and Q as well as the delay of the optimal CSKA may be calculated [19]. In the case, where α is a non-integer value, one may realize only a near optimal structure, as detailed in [19] and [21]. In this case, most of the time, by setting M1 to 1 and using (6) and (7), the near-optimal structure is determined. It should be noted that, in practice, α is non-integer 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]

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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

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

$$T_{\text{PD}_{\text{opt}}} = T_{\text{CARRY}} + \left(2\sqrt{\frac{N}{\alpha}} - 1\right)T_{\text{SKIP}} + T_{\text{SUM}}. \quad (9)$$

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

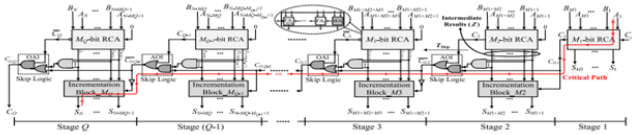

Fig. 2. Proposed CI-CSKA structure

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 Mj (j = 1,..., Q). In this structure, the carry input of all the RCA blocks, except for the first block which is Ci , 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., SM1 ,..., S1), and C1, the other blocks simultaneously compute the intermediate results [i.e., {ZK j+Mj,..., ZK j+2, ZK j+1} for K j = j−1 r=1 Mr(j = 2,..., Q)], and also Cj 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

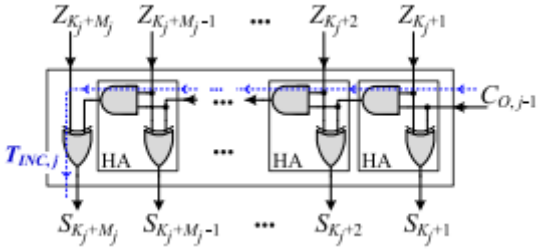

Fig. 3. Internal structure of the jth incrementation block, $K_j = \sum_{r=1}^{j-1} Mr$ (j = 2,..., Q).

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 (CO,j) based on the intermediate results of the jth stage and the carry output of the previous stage (CO,j−1) as well as the carry output of the corresponding RCA block (Cj). When determining

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

$C_{O,j}$, these cases may be encountered. When Cj is equal to one, $C_{O,j}$ will be one. On the other hand, when Cj 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). 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 (TD) 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 threeparts mentioned above, respectively. Here, TAND and TXORare the delays of the two inputs static AND and XOR gates,respectively. Note that, [(Mj − 1)TAND + TXOR] shows thecritical path delay of the jthincrementation block (TINC,j),which is shown in Fig. 3.To calculate the delay of the skip logic, the average of thedelays of the AOI and OAI gates, which are typically close toone 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 TAOI and TOAI 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 TAND and TXOR are smaller than TCARRY and TSUM, 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.

B. 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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

(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 M1 to MQ , are obtained using a method similar to the one discussed in Section III-B. For this structure, the new value for TSKIP should be used, and hence, α becomes (TAOI+TOAI) / (2×TCARRY). 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 consideredas the nucleus (pth) stage. There are cases that we should consider the stage right before this stage as thenucleu s 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 (TINC,i and TINC,i−1, respectively), and the delay of the skip logic. In particular

$$T_{\text{INC},i} \le T_{\text{INC},i-1} - T_{\text{SKIP},i-1}; \quad \text{for } i \ge p+1. \quad (13)$$

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 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.

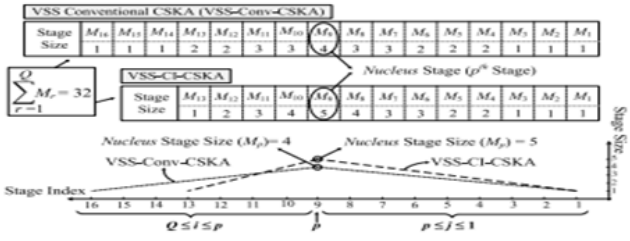

Fig. 4. Sizes of the stages in the case of VSS for the proposed andconventional 32-bit CSKA

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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 (<1/2m), 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.

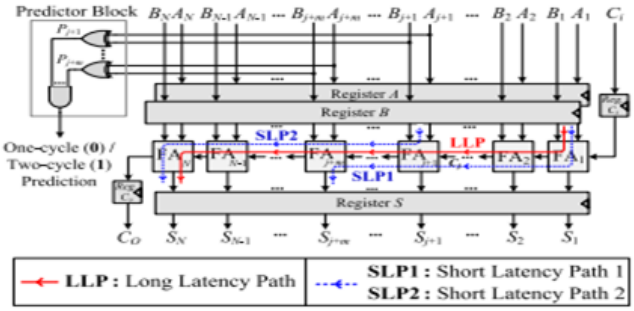

Fig. 5. Generic structure of variable latency adders based on RCA.

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 Mp-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 off-critical paths. Thus, the use of the fast PPA

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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.

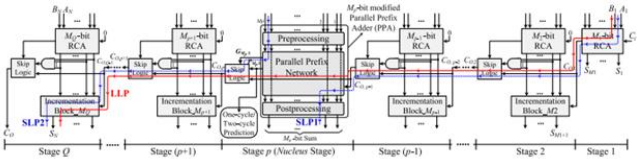

Fig. 6.Structure of the proposed hybrid variable latency CSKA.

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 thestage 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., Mp = 8).

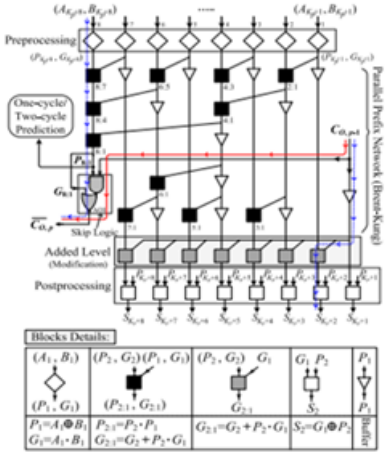

Fig. 7. Internal structure of the pth stage of the proposed hybrid variablelatency CSKA. Mp is equal to 8 and Kp=$\sum^{j-1}_{r=1}$ Mr.

As shown in the figure, in the preprocessing level, the propagate signals (Pi) and generate signals (Gi) for the inputs are calculated. In the next level, using Brent–Kung parallel prefix network, the longest carry (i.e., G8:1) of the prefix network along with P8:1, which is the product of the all propagate signals of the inputs, are calculated sooner than other intermediate signals in this network. The signal P8:1 is used in the skip logic to determine if the carry output of the previous stage (i.e., CO,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 P8:1 is one, CO,p−1 should skip this stage predicting that some critical paths are activated. On the other hand, when P8:1 is zero, CO,p is equal to the G8:1. In addition, no critical path will be activated in this case. After the parallel prefix network, the intermediate carries, which are functions of CO,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 CO,p−1 to final summation results of the PPA block and the beginning part of the SPL2 paths from inputs of this block to CO,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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

structure, leads to the decrease in the number of stages as well smaller delays for SLP1 and SLP2. Thus, the slack time increases further.

V.SIMULATION RESULTS:

All the synthesis and simulation results are performed using Verilog HDL. The synthesis and simulation are performed on Xilinx ISE 14.4. The simulation results are shown below figures. The corresponding simulation results of the variable stage size carry skip adders are shown below.

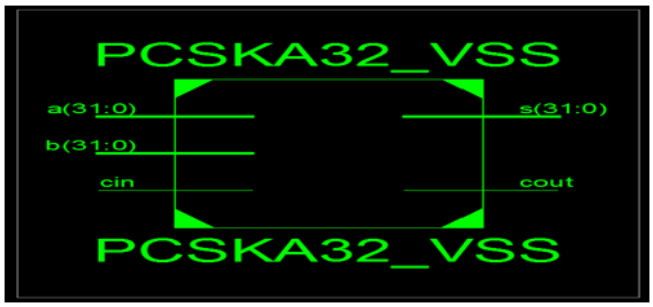

Figure 7.13: RTL schematic of Top-level Variable Stage Size Carry Skip Adder

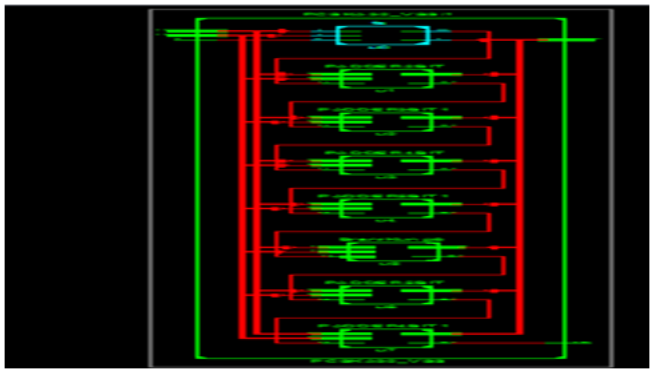

Figure 7.14: RTL schematic of Internal block Variable Stage Size Carry Skip Adder

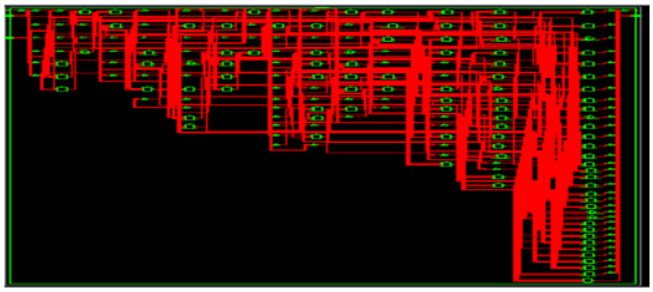

Figure 7.16: Technology schematic of Internal block Variable Stage Size Carry Skip Adder

PCSKA32_VSS Project Status			
Project File:	HCSKA.xise	Parser Errors:	No Errors
Module Name:	PCSKA32_VSS	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	•Errors:	No Errors
Product Version:	ISE 14.4	•Warnings:	20 Warnings (0 new)
Design Goal:	Balanced	•Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	
Environment:	System Settings	•Final Timing Score:	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	51	4656	1%
Number of 4 input LUTs	91	9312	0%
Number of bonded IOBs	98	232	42%

Detailed Reports					[-]
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Sat Aug 13 13:33:32 2016	0	20 Warnings (0 new)	0

Figure 7-1: Synthesis report of Variable Stage Size Carry Skip Adder

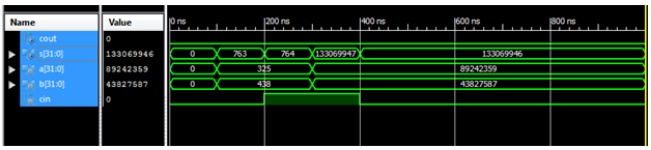

Figure 8-2: Simulated output for Variable Stage Size Carry Skip Adder

CONCLUSION:

In this paper, 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 bycomparing 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. The efficacy of this structure was compared versus those of the variable latency RCA, C2SLA, and hybrid C2SLA structures. Again, the suggested structure showed the

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

lowest delay as a better candidate for high-speed 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 carrylookahead/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-32processor 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 andtheir synthesis," Ph.D. dissertation, Dept. Inf. Technol. Elect.Eng., SwissFederal 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 efficientsolution 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, andR. Krishnamurthy, "Energy-delay estimation technique for

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

highperformancemicroprocessor 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 carrypropagationin 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 carryskip adder," in Proc. EuromicroSymp.Digit. Syst. Design (DSD),Aug./Sep. 2004, pp. 615–619.

[19] M. Alioto and G. Palumbo, "A simple strategy for optimized designof 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 inadders," 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 carryskipadders," 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-lookaheadadders 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," IEEETrans. Comput., vol. 42, no. 6, pp. 759–764, Jun. 1993.

[25] V. Kantabutra, "Accelerated two-level carry-skip adders—A type of veryfast 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 adaptivesupply 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 (C2SA): 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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.