DESIGN OF HIGH SPEED CARRY SKIP ADDER USING KOGGE STONE ADDER

[1] J.Balarama Krishna Murthi, [2] P.Mohan, [3] A.Navyendranath, [4]B.Anusha, [5] M.Saritha Devi

[1,2,3,4] U G Scholars, [5]Assistant Professor

[1]Electronics and Communication Engineering
[1]Godavari Institute of Engineering & Technology, Rajahmundry, Andhra Pradesh

Abstract: In this paper, we present a carry skip adder (CSKA) structure that has a higher speed 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 an OR-AND-Invert (OAI) compound gate for the skip logic and Kogge-Stone adder is used in the place of fixed Ripple Carry Adder (RCA). The structure may be realized with fixed stage size style, wherein the latter further improves the speed and energy parameters of the existing adder. Both Existing and Proposed architecture design of CSKAs has been simulated by using Xilinx 14.7. Analysis has been done in terms of speed as well as hardware complexity.

***Index Terms* – CSKA, OAI, Kogge-Stone Adder.**

I. INTRODUCTION

Adders are a key building block in arithmetic and logic units (ALUs) 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 speed and power of these units, which have been reported. Obviously, it is highly desirable to achieve higher speeds

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. 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 nano scale 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. Recently, the near-threshold region has been considered as a region that provides a more desirable trade off 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, 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. In these circuits, to reduce the energy consumption, the system may change the voltage (and frequency) of the circuit based on the workload requirement. 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. 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. 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. There are different types of the parallel prefix algorithms that lead to different PPA structures with different performances.

The major objective of this work is to evaluate and explore the existing CSA with the modified approach with a focused goal on adder performance in addition to low power and high speed. Computational performance is noted by utilizing the Xilinx. Predominantly high- speed design in addition to the reconfigurable organization is the main intention of this work. Furthermore, the design is made in such a way it must reliable as well as should have less in complexity. Since a large number of transistors are put into operation in the small area. Subsequently, to ensure the performance, the system is simulated as well as tested in VHDL. The Proposed adder has architecture of 20-bit adder using Kung-Stone adder. Implementation of the adder develops the VHDL code and then simulation using Xilinx 14.7.

II. LITERATURE SURVEY

A detailed presentation of the existing design work that is required for the proposed high CSKA implementation. It covers the necessity of reduce of delay, since power and speed efficient implementation is a very challenging task. At the same time, to achieve the high computational speed, existing work covers the minimization of power consumption by diminishing the switched capacitance that confers one by one as follows:

The Authors **M. Lehman** and **N. Burla** proposed Carry skip adder addition is performed in multiple stages. Each stage consists of an RCA block, a multiplexer and a carry prediction unit. RCA is used to find the sum of each stage. It takes the input bits of two numbers A and B, and generates the sum. The designed model uses a 4-bit RCA block. When all bits of corresponding stage are in propagation condition and the carry prediction unit will generate one as output.

The Authors **Alioto** and **Palumbo** 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 non-integer ratio of the skip time to the ripple time on contrary to most of the previous works, which considered an integer ratio.

The Authors **M. Lehman** and **N. Burla** proposed CSKA developed architecture of CSKA was developed have very less delay when compared to conventional CSKA. The model is CICSA (Concatenation Incrementation CSKA) which combines the concatenation and incrementation schemes to conventional CSKA and carry is skipped by using AOI and OAI logic. This structure focus on increasing speed of operation and decreasing the area. The use of AND OR invert logic as carry skip will also help to decrease area as it has a smaller number of transistors when compared to multiplexer.

The Authors **Premson** and **Sakthivel** proposed two different Carry skip adders are compared and slightly modified version of both architectures is proposed that can increase the speed. It incorporates a carry feed forward block to the existing structures so that the delay can be reduced. The proposed architecture can be used for high speed application by the cost of area. The adders compared in this paper are CSKA (conventional carry skip adder) and CI-CSKA (Concatenation Incrementation Carry skip adder), and the proposed models are CFF- CSKA (Carry Feed Forward CSKA) and CFF-CI-CSKA.

The Authors **Jaishree Sundar** and **G. Keerthana** proposed CSKA attempts to examine the features such as memory, time delay, performance of adder circuits like 64bit ripple carry, carry look ahead and carry skip adder which guarantees superior performance compared to existing circuits. Now a days designing adder circuits is not a big issue. Designing an adder with less delay reduced chip area and consumes less power this is main objective for a designer.

The Authors **R.Ezhilarasi** and **T.Perarasi** proposed CSKA, the proposed structure, parallel prefix adder network is used to improve the speed and energy parameters. In addition, the proposed structure uses AND-OR-Invert (AOI) and ORAND-Invert (OAI) gates for the skip logic. These compound gates consist of fewer transistors, have a lower delay, area, and smaller power consumption compared with those 2:1 multiplexer. In the Proposed system, the Kogge-Stone adder is used. Kogge-stone adder is a type of parallel prefix adder that has a lower fan-out at each stage.

III. CONVENTIONAL CSKA

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 2:1multiplexers, which can be placed into one or more level structures. The CSKA configuration (i.e., the number of the FAs per stage) has a great impact on the speed of this type of adder. Many methods have been suggested for finding the optimum number of the FAs. The techniques presented. Make use of VSSs to minimize the delay of adders based on a single level carry skip logic. 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. 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 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 non-integer ratio of the skip time to the ripple time on contrary to most of the previous works, which considered an integer ratio. In all of the works reviewed so far, the focus was on the speed, while the power consumption and area usage of the CSKAs.

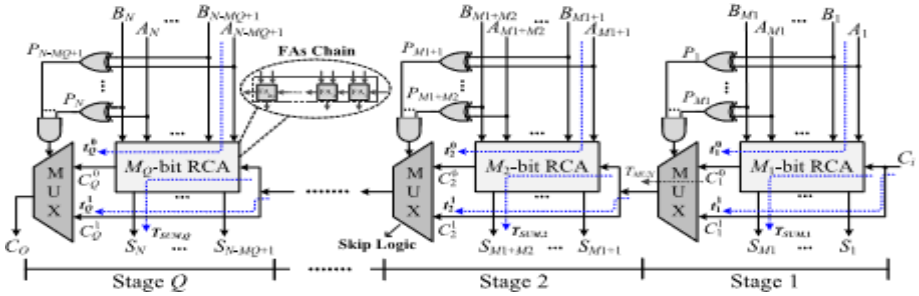

Fig. 1 Conventional structure of the CSKA.

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. 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.

The efficiency of this method for reducing the power consumption of the RCA structure has been demonstrated. The CSLA structure in was enhanced to use adaptive clock stretching operation where the enhanced structure was called cascade CSLA (C2SLA). Compared with the common CSLA structure, C2SLA 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 the proposed hybrid structure, the KSA has been used in the middle part of the C2SLA where this combination leads to the positive slack time increase. However, the C2SLA 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.

IV. EXISTING CSKA

The structure of an N-bit Conv-CSKA, which is based on blocks of the RCA (RCA blocks), is shown in Fig. 2. 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 Pi = Ai ⊕ Bi =1 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. 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, 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. 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.

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 C0 j, the carry output of the previous stage (carry input of the jth stage) denoted by C1 j (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 × TCARRY;
2. the path of the intermediate carry skip multiplexer whose delay is equal to the (Q−1)×TMUX;
3. the path of the FA chain in the last stage whose its delay is 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

TD=[M×TCARRY]+NM−1×TMUX+[(M −1)×TCARRY +TSUM] (1)

Based on (1), the optimum value of M (Mopt) that leads to optimum propagation delay may be calculated as (0.5Nα)1/2 where α is equal to TMUX/TCARRY. Therefore, the optimum propagation delay (TD,opt) is obtained from

TD,opt=22NTCARRYTMUX+(TSUM−TCARRY−TMUX)=TSUM+(2√2Nα−1−α)×TCAR RY.

(2) Thus, the optimum delay of the FSS CSKA is almost proportional to the square root of the product of N and α.

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 areminimized 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 propagationdelay of the C1 j (t1 j) by

t1 j =maxt0 j−1,t1 j−1+TMUX (2)

where t0 j−1 (t1 j−1) shows the calculating delay of C0 j−1(C1 j−1)signal in the (j −1)th stage. In a FSS CSKA, except in the first stage, t0 j is smaller than t1 j. Hence, based on (2), the delay of t0 j−1 may be increasedfromt0 1 to t1 j−1 without increasing the delay of C1 j signal. This means that one could increase the size of the (j − 1)th stage (i.e., Mj−1) without increasing the propagation delay of the CSKA. Therefore, increasing the size of Mj for the jth stage should be bounded by

t0 j ≤t1 j =t0 1 +(j −1)TMUX. (3)

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 t1 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 t1 i +TMUX+TSUM,i+1, whereTSUM,i+1 isthe 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

TSUM,i+1 ≤ TSUM,i −TMUX; for i ≥ p. (4)

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 (3) and (4) 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 C0 j. In this case, the maximum of t0 j is equal to Mj ×TCARRY. To satisfy (4), we increase the size of the first p stages up to the nucleus using

$$Mj \le M1 + (j-1)\alpha; \text{ for } 1 \le j \le p. \quad (5)$$

In addition, the maximum of TSUM,i is equal to (Mi −1)×T CARRY +TSUM. To satisfy (5), the size of the last (Q − p) stages from the nucleus to the last stage should decrease based on

$$Mi \ge MQ + (Q-i)\alpha; \text{ for } p \le i \le Q. \quad (6)$$

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. In the case, where α is a non-integer value, one may realize only a near optimal structure, as detailed. 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, where the estimation of the near-optimal propagation delay of the CSKA is given by

$$TD,opt = 2\alpha 2 - 1TCARRY + 2N\alpha - 1TMUX + TSUM. \quad (7)$$

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 TSKIP/TCARRY. Finally, note that in real implementations, TSKIP < TCARRY, and hence

$$TPDopt = TCARRY + 2N\alpha - 1TSKIP + TSUM. \quad (8)$$

Note that, as (8) reveals that a large portion of the critical path delay is due to the carry skip logics.

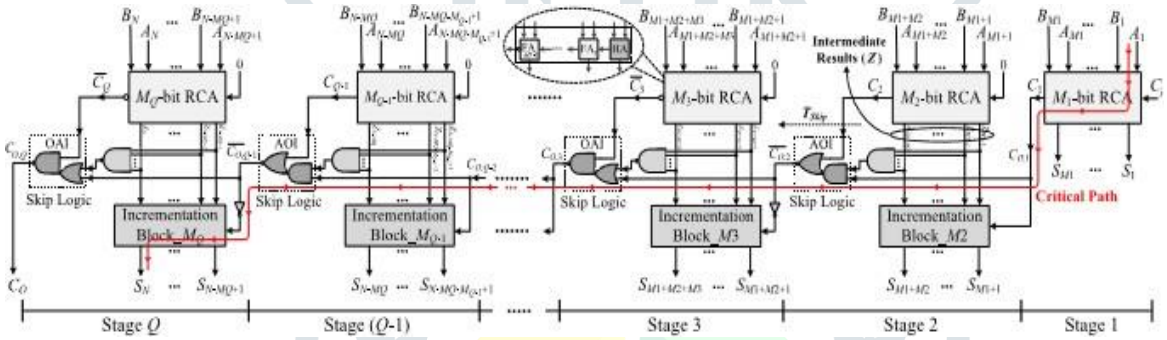

Fig. 2 Existing CI-CSKA structure

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.

General Description of the Existing Structure The structure is based on combining the concatenation and the incrementation schemes 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 2:1multiplexers 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. 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.,

{ZKj+Mj,...,ZKj+2, ZKj+1} for Kj = j−1 r=1 Mr(j = 2,...,Q)], and alsoCj 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.

$$Kj = j-1\ r=1\ Mr(\ j = 2,...,Q).$$

Intermediate results generated by the RCA block and the carry output of the previous stage to calculate the fi nal 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 CO,j, these cases may be encountered. When Cj is equal to one, CO,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 CO,j will be the same as CO,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.

V. PROPOSED CSKA

The Brent–Kung adder is a parallel prefix adder (PPA) form of carry-lookahead adder (CLA). Proposed by Richard Peirce Brent and Hsiang Te Kung in 1982 it introduced higher regularity to the adder structure and has less wiring congestion leading to better performance and less necessary chip area to implement compared to the Kogge–Stone adder (KSA). It is also much quicker than ripple-carry adders (RCA).

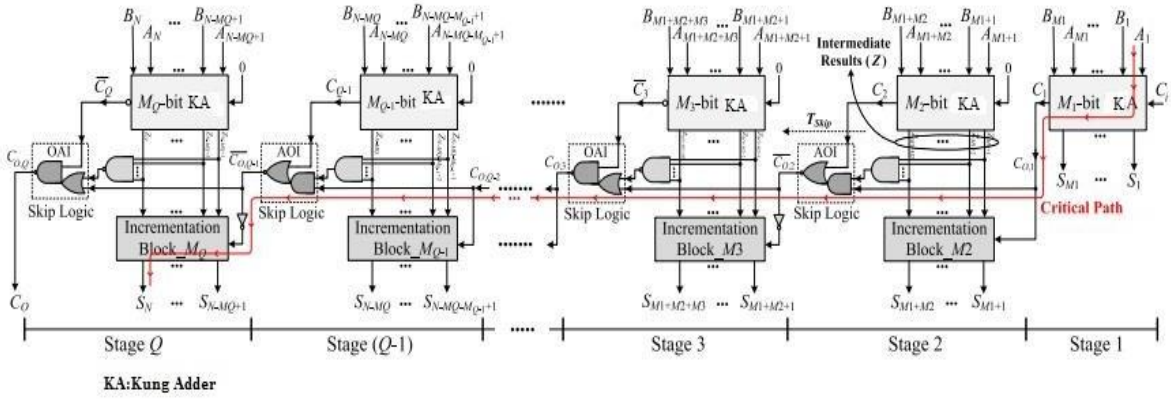

Fig. 3 Proposed Carry Skip Adder Using Kogge-Stone Adder

Ripple-carry adders were the initial multi-bit adders which were developed in the early days and got their name from the ripple effect which the carry made while being propagated from right to left. The time taken for addition was directly proportional to the length of the bit being added. This is reverse in Brent–Kung adders where the carry is calculated in parallel thus reducing the addition time drastically. Further work has been done on Brent–Kung adders and other parallel adders to reduce the power consumption and chip area as well as to increase the speed thus making them suitable for low-power designs. A Brent–Kung adder is a parallel adder made in a regular layout with an aim of minimizing the chip area and ease manufacturing. The addition of n-bit number can be performed in time O(nLog2n) with a chip size of area O(Log2n) thus making it a good-choice adder with constraints on area and maximizing the performance. Its symmetry and regular build structure reduces costs of production effectively and enable it to be used in pipeline architectures. In parallel adders the critical path is decided by computation of the carry from least significant bit (LSB) adder to the most significant bit (MSB) adder, therefore efforts are in reducing the critical path for the carry to reach the MSB.

VI. RESULTS AND DISCUSSION

The design was synthesized on Xilinx ISE and the functional verification of Existing CSKA and proposed CSKA was done on Xilinx ISIM. The targeted device is of Spartan-3e of Spartan family. The grade speed of the design is set to -5. The following section contains the results obtained by synthesizing the design in Xilinx ISE.

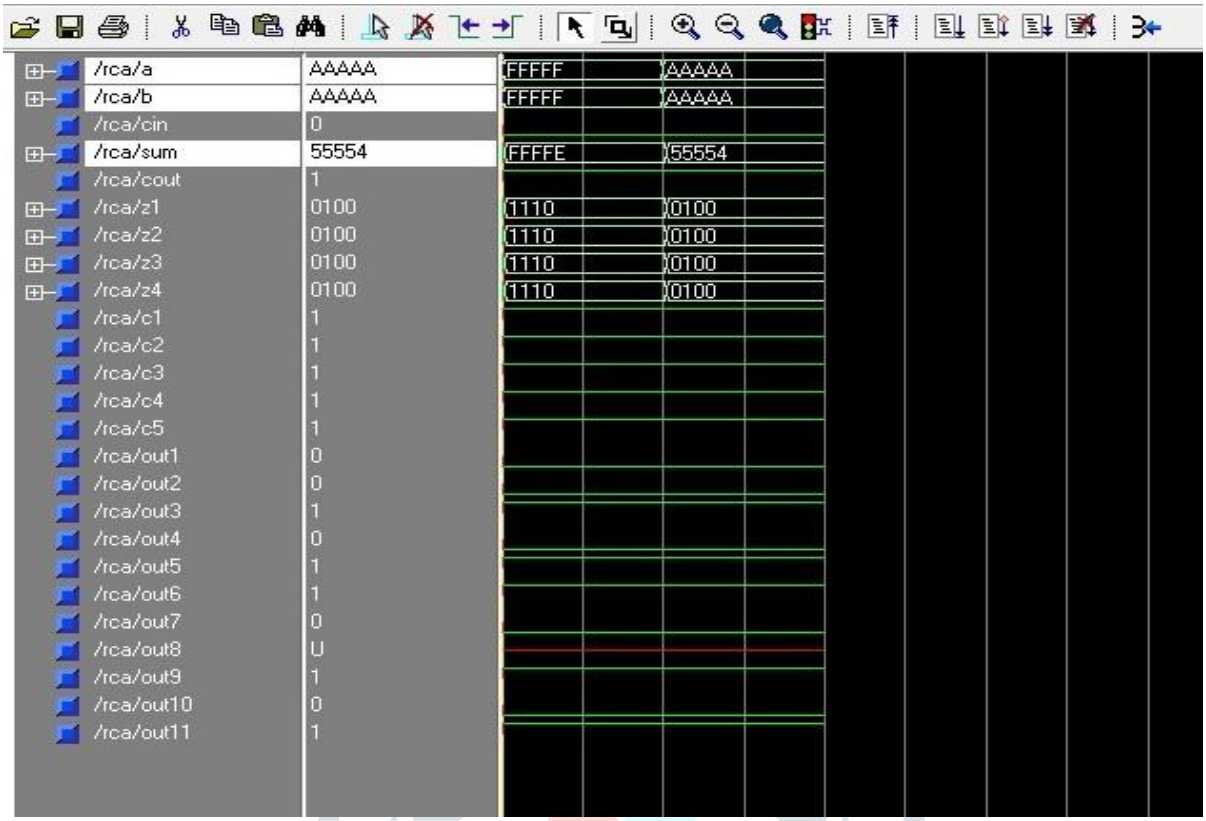

Fig. 4 Simulation output of Proposed 20-Bit CSKA

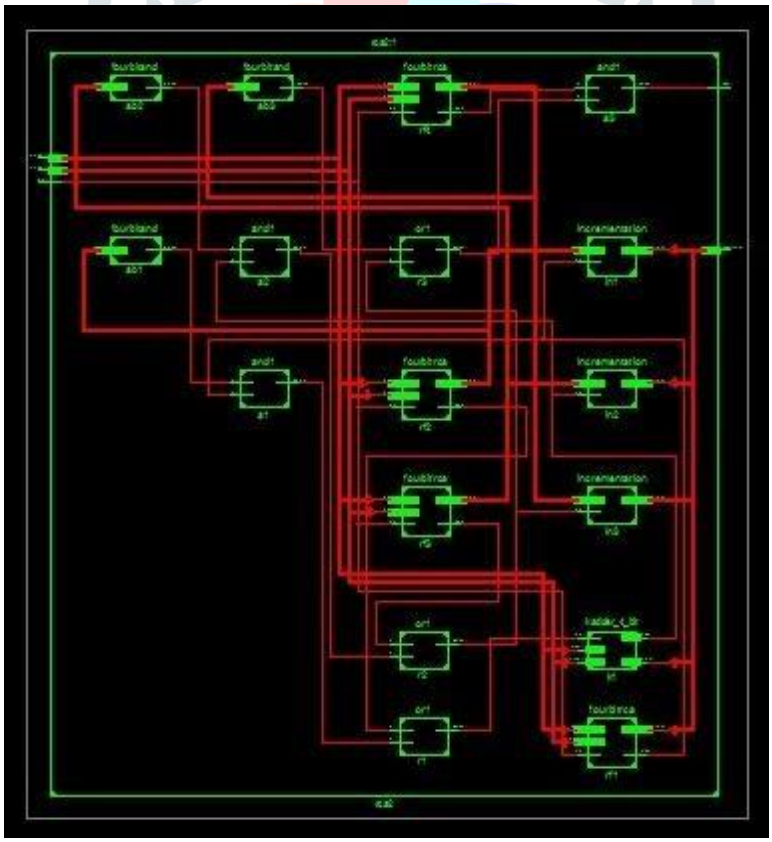

Fig. 5 Proposed 20-Bit CSKA RTL Schematic

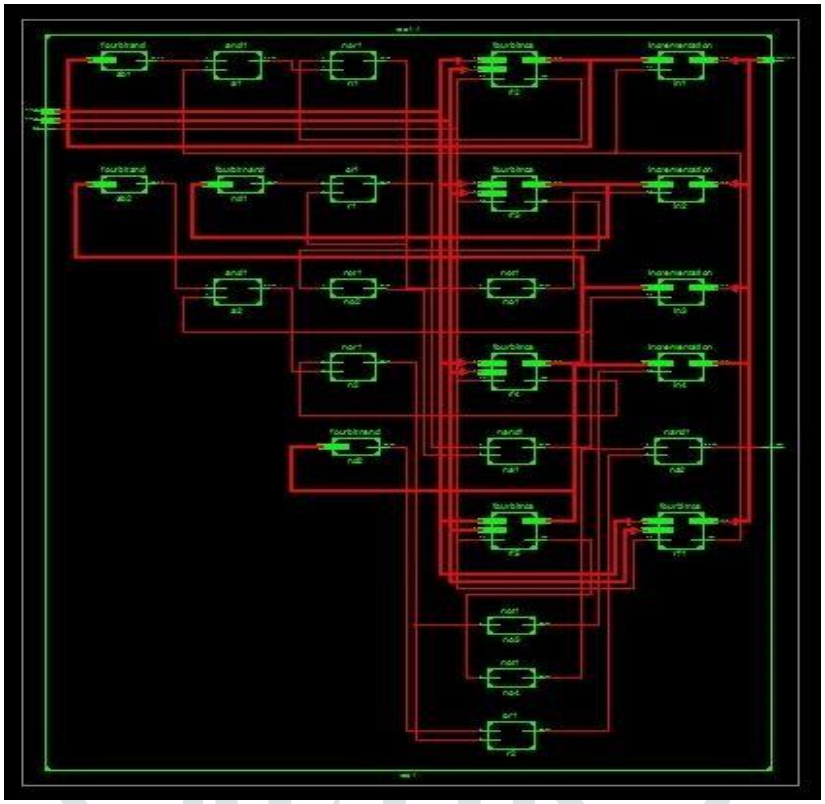

Fig. 6 Existing 20-Bit CSKA RTL Schematic

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, C2SLA, and hybrid C2SLA 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] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy–delay optimization of 64-bit carrylookahead adders with a 240 ps 90 nm CMOS design example," IEEE J. Solid-State Circuits, vol. 44, no. 2, pp. 569–583, Feb. 2009.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

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

[17] 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.

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

[19] 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.

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