ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.6

VLSI Design of a High-Speed and Energy-Efficient Carry Skip Adder

M.Manasa[1], E.Swapna[2]
M.Tech Student[1], Assistant Professor[2]
Department of ECE (DSCE) [1], Department of ECE[2]
KITS, Singapur, Telangana, India

Abstract:
In this paper implemented a carry skip adder (CSKA) structure that has a higher speed and 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. Finally, a hybrid variable latency extension of the proposed structure, which lowers the power consumption without considerably impacting the speed, is presented.

Keywords: CSKA, hybrid structure, AOI, OAI.

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 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.

II. LITERATURE SURVEY

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.

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 singlelevel 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 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.

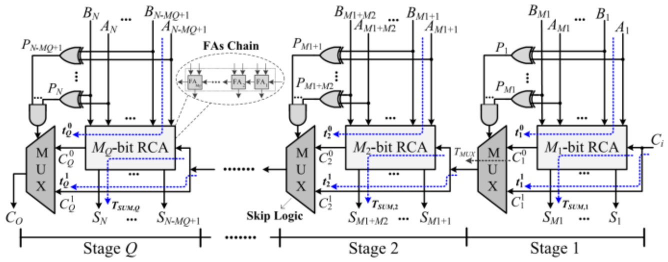
Improving Efficiency of Adders at Low Supply Voltages:

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

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

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, 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.

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 *j*th multiplexer are the carry output of the FAs chain in the *j*th stage denoted by C^0_j, the carry output of the previous stage(carry input of the *j*th stage) denoted by C^1_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 $M \times T_{\mathrm{CARRY}}$; 2) the path of the intermediate carry skip multiplexer whose delay is equal to the $(Q - 1) \times T_{\mathrm{MUX}}$; and 3) the path of the FA chain in the last stage whose its delay is equal to the $(M - 1) \times T_{\mathrm{CARRY}} + T_{\mathrm{SUM}}$. 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

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

Based on (1), the optimum value of M (M opt) 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

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

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

IV. PROPOSED CSKA STRUCTURE

Based on the discussion 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.

4.1 General Description of the Proposed 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 2:1 multiplexers by AOI/OAI compound gates. 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 below Fig. 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 =$\Sigma^{j-1}_{r=1}$ 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 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 above Fig. 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. 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 (Cj). 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). 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. 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.

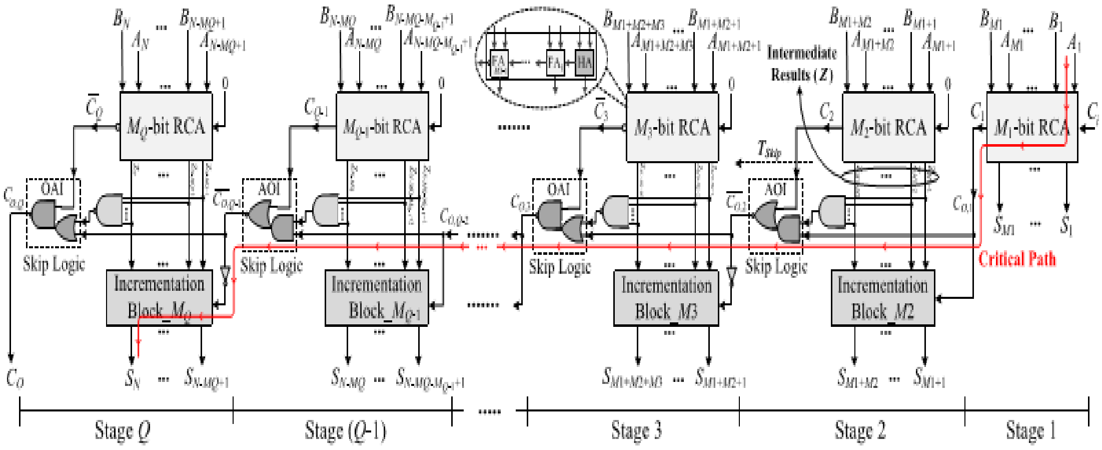


Figure. 4.1(a): Proposed CI-CSKA Structure.

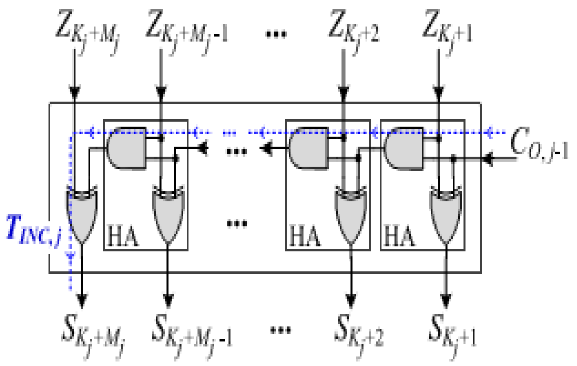


Figure 4.1(b): Internal Structure of the Jth Incrementation Block.

4 .2 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. 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. 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. 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

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

Where the three brackets correspond to the three parts mentioned above, respectively. Here, TAND and TXOR are the delays of the two inputs static AND and XOR gates, respectively. Note that, [(Mj −1) TAND+TXOR] shows the critical path delay of the jth incrementation block (TINC,j). 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, may be modified.

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

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). It should be noted that the delay reduction of the skip logic has the largest impact on the delay decrease of the whole structure.

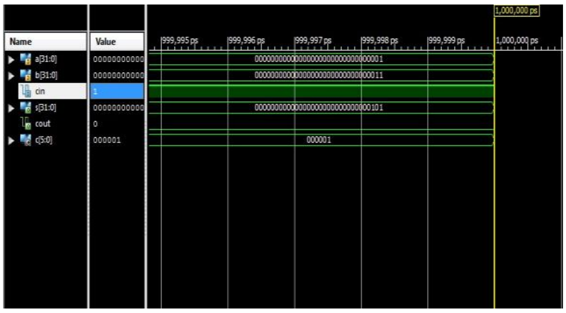
4.3Proposed Hybrid Variable Latency CSKA:

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 is proposed. 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. 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 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. 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 is used for constructing the nucleus stage. 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. Finally, it has a simple and regular layout. The internal structure of the stage p, including the modified PPA and skip logic.

V.RESULTS

5.1 Simulation Results:

In this paper architecture is designed by Verilog HDL, this is simulated by using Xillinx 13.2, verify the functionality. We have verified the functionality for 32 bit. Below fig shows the addition of two 32 bit numbers.



Name	Value
a[31:0]	000000000000
b[31:0]	000000000000
cin	1
s[31:0]	000000000000
cout	0
c[5:0]	000001

5.2 Synthesis Results:

This design is synthesized and its results were analyzed as follows.

5.2.1 RTL Schematic

The synthesis results of both the conventional and proposed CSKA are shown in below figures.

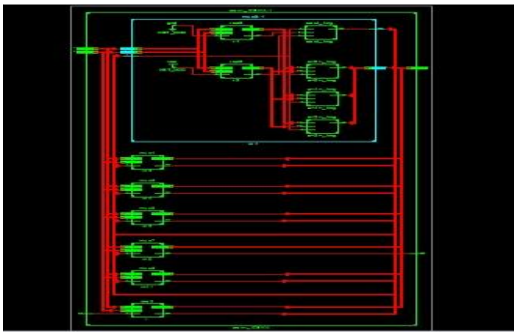


Figure 5.2(a): RTL Schematic of Conventional CSKA.

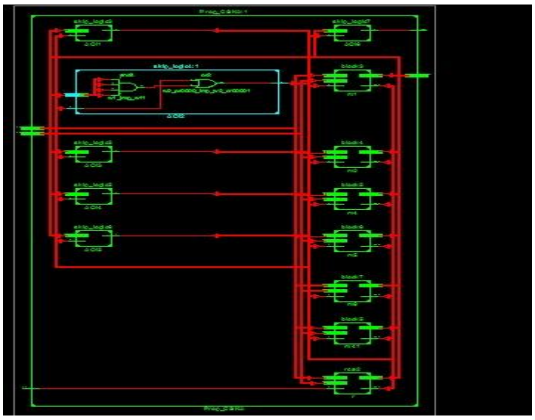


Figure.5.2(b): RTL Schematic of Proposed CSKA.

5.2.2 Technology Schematic:

The schematic diagrams of both conventional and proposed CSKA are shown in below figures.

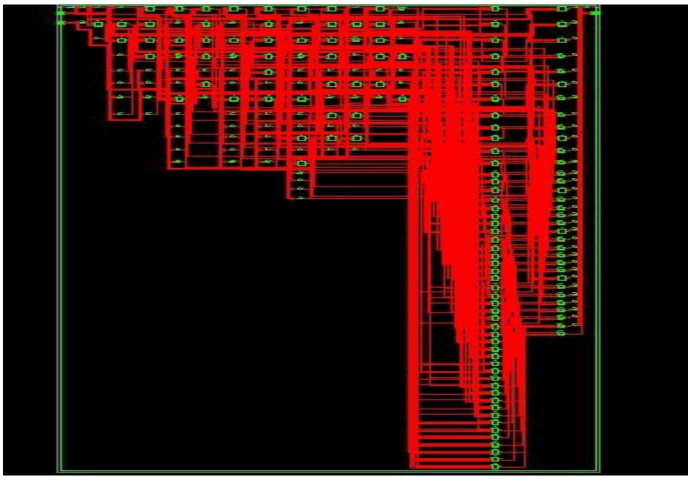


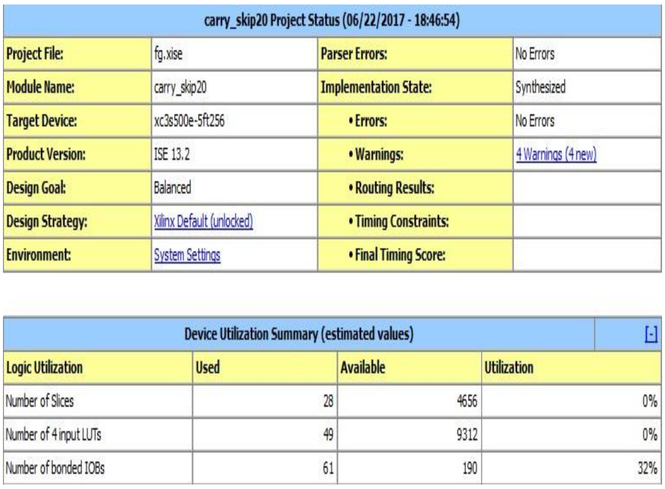
Figure.5.2(c): Technology Schematic of conventional CSKA.

Figure. 5.2(d): Technology Schematic of Proposed CSKA.

5.3 Design Summary:

5.3.1 Area

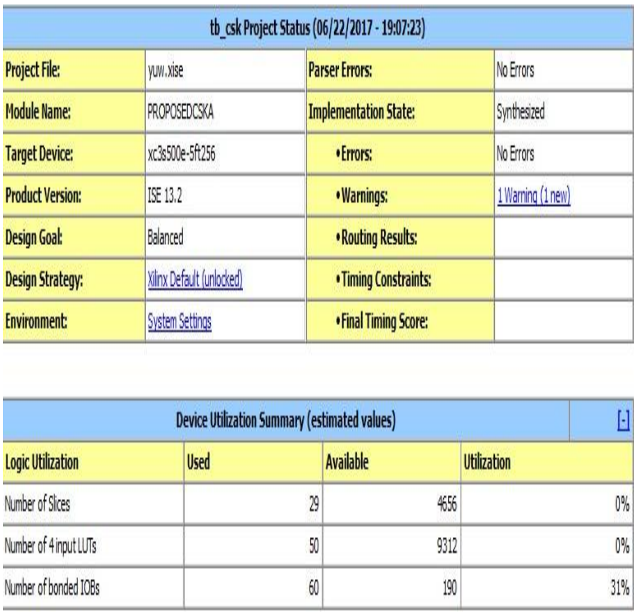
The design summary of conventional CSKA and Proposed CSKA are shown in below fig. From the following design summary it has been concluded that area usage by proposed CSKA has been decreased compared to the conventional CSKA Hence, as area decreased the components usage also decreased and so that power consumption decreases and speed increases



carry_skip20 Project Status (06/22/2017 - 18:46:54)			
Project File:	fg.xise	Parser Errors:	No Errors
Module Name:	carry_skip20	Implementation State:	Synthesized
Target Device:	xc3s500e-5ft256	• Errors:	No Errors
Product Version:	ISE 13.2	• Warnings:	4 Warnings (4 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	28	4656	0%
Number of 4 input LUTs	49	9312	0%
Number of bonded IOBs	61	190	32%

Figure.5.3(a): Design Summary of conventional CSKA.



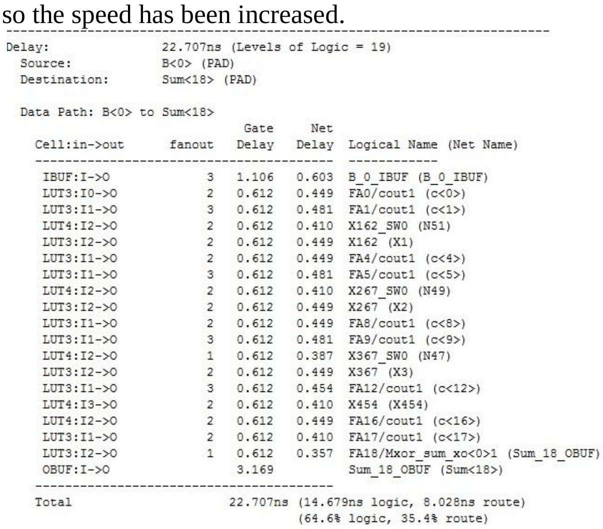
tb_csk Project Status (06/22/2017 - 19:07:23)			
Project File:	yuw.xise	Parser Errors:	No Errors
Module Name:	PROPOSEDCSKA	Implementation State:	Synthesized
Target Device:	xc3s500e-5ft256	•Errors:	No Errors
Product Version:	ISE 13.2	•Warnings:	1 Warning (1 new)
Design Goal:	Balanced	•Routing Results:	
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	29	4656	0%
Number of 4 input LUTs	50	9312	0%
Number of bonded IOBs	60	190	31%

Figure. 5.3(b): Design Summary of proposed CSKA.

7.3.2 Delay

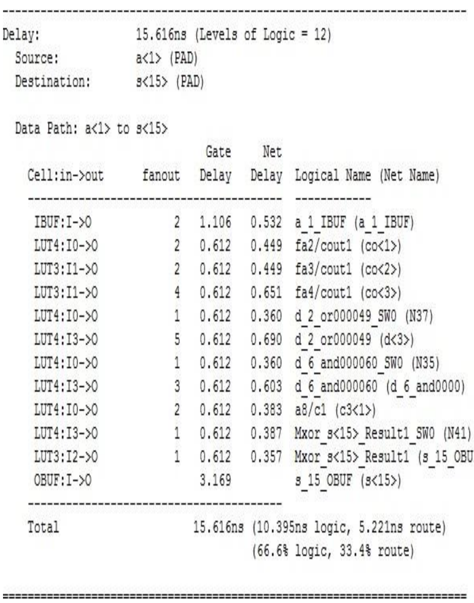
The delay of conventional and proposed CSKA has been shown in below fig. The delay of the proposed structure has been decreased compared to the conventional CSKA structure so the speed has been increased.



```
--------------------------------------------------------------------------------
Delay:               22.707ns (Levels of Logic = 19)
  Source:            B<0> (PAD)
  Destination:       Sum<18> (PAD)

  Data Path: B<0> to Sum<18>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             3   1.106   0.603  B_0_IBUF (B_0_IBUF)
     LUT3:I0->O            2   0.612   0.449  FA0/cout1 (c<0>)
     LUT3:I1->O            3   0.612   0.481  FA1/cout1 (c<1>)
     LUT4:I2->O            2   0.612   0.410  X162_SW0 (N51)
     LUT3:I2->O            2   0.612   0.449  X162 (X1)
     LUT3:I1->O            2   0.612   0.449  FA4/cout1 (c<4>)
     LUT3:I1->O            3   0.612   0.481  FA5/cout1 (c<5>)
     LUT4:I2->O            2   0.612   0.410  X267_SW0 (N49)
     LUT3:I2->O            2   0.612   0.449  X267 (X2)
     LUT3:I1->O            2   0.612   0.449  FA8/cout1 (c<8>)
     LUT3:I1->O            3   0.612   0.481  FA9/cout1 (c<9>)
     LUT4:I2->O            1   0.612   0.387  X367_SW0 (N47)
     LUT3:I2->O            2   0.612   0.449  X367 (X3)
     LUT3:I1->O            3   0.612   0.454  FA12/cout1 (c<12>)
     LUT4:I3->O            2   0.612   0.410  X454 (X454)
     LUT4:I2->O            2   0.612   0.449  FA16/cout1 (c<16>)
     LUT3:I1->O            2   0.612   0.410  FA17/cout1 (c<17>)
     LUT3:I2->O            1   0.612   0.357  FA18/Mxor_sum_xo<0>1 (Sum_18_OBUF)
     OBUF:I->O                 3.169          Sum_18_OBUF (Sum<18>)
    ----------------------------------------
    Total                     22.707ns (14.679ns logic, 8.028ns route)
                                       (64.6% logic, 35.4% route)
```

Figure. 5.3(c) Delay of conventional CSKA



```
--------------------------------------------------------------------------------
Delay:               15.616ns (Levels of Logic = 12)
  Source:            a<1> (PAD)
  Destination:       s<15> (PAD)

  Data Path: a<1> to s<15>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   1.106   0.532  a_1_IBUF (a_1_IBUF)
     LUT4:I0->O            2   0.612   0.449  fa2/cout1 (co<1>)
     LUT3:I1->O            2   0.612   0.449  fa3/cout1 (co<2>)
     LUT3:I1->O            4   0.612   0.651  fa4/cout1 (co<3>)
     LUT4:I0->O            1   0.612   0.360  d_2_or000049_SW0 (N37)
     LUT4:I3->O            5   0.612   0.690  d_2_or000049 (d<3>)
     LUT4:I0->O            1   0.612   0.360  d_6_and000060_SW0 (N35)
     LUT4:I3->O            3   0.612   0.603  d_6_and000060 (d_6_and0000)
     LUT4:I0->O            2   0.612   0.383  a8/c1 (c3<1>)
     LUT4:I3->O            1   0.612   0.387  Mxor_s<15>_Result1_SW0 (N41)
     LUT3:I2->O            1   0.612   0.357  Mxor_s<15>_Result1 (s_15_OBUF)
     OBUF:I->O                 3.169          s_15_OBUF (s<15>)
    ----------------------------------------
    Total                     15.616ns (10.395ns logic, 5.221ns route)
                                       (66.6% logic, 33.4% route)

================================================================================
```

Figure.5.3(d) Delay of Proposed CSKA

VI. CONCLUSION

In this project, 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 was studied by comparing its power and delay with the Conv-CSKA structure. 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.

VII. 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 look ahead/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 carrypropagation 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 carryskip 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 (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. 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.