VOL. 13, NO. 5, MARCH 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

INTEND OF POWER-DELAY OPTIMIZED KOGGE-STONE BASED CARRY SELECT ADDER

Prajoona Valsalan[1] and P. Manimegalai[2]
[1]Karpagam University, Coimbatore, Tamil Nadu, India
[2]Department of Electronics and Communication Engineering, Karpagam University, Coimbatore, India
E-Mail: prajoonavalsalan.phd@gmail.com

ABSTRACT

In computation designs, the arithmetic operation plays a major role in all type of logical designs. It decides the performance of all internal components and its behavior. Hence, recent researchers focused on obtaining the high speed of operation with minimal power consumption. The conventional Carry Select Adder results in more area due to the inbuilt nature of two Ripple Carry Adders in the same structure. Similarly, the AND-OR-Invert based Carry Select Adder consists of complex circuits for processing the basic operation. Hence, a Kogge–Stone adder based carry select adder is designed, which is replacing the AOI logic from the previous AND-OR-Invert based Carry Select Adder to achieve optimization in terms of area, delay, and power. The proposed design is synthesized and simulated under cadence 45nm CMOS technology. The comparison shows how the proposed KS-CSLA is better than the existing regular dual Ripple Carry Adder (RCA), modified adder with BEC concept, Dual RCA and modified Square root adder with BEC unit. Finally, the speed of proposed model is increased twice of the speed of dual RCA.

Keywords: adders, high-speed carry select adder, ripple carry adders, kogge-stone adder.

1. INTRODUCTION

In submicron technology, the scaling is one of the important tasks to reduce the power and area factors. In complex digital designs, the role of the adder is huge among the internal units. It decides the overall performance and characteristics. Some of the traditional adders are there such as Ripple Carry Adder (RCA), Carry Look a Head adder and so on. In this research, the Carry Select adder is considered because it alleviates the problem of carry propagation. In traditional methods, the Ripple carry Adder (RCA) is used to design a CSLA. But it results in a poor performance. Hence, our previous research focussed on designing a carry select adder with the help of AND-OR Inverter Logic. Since, there is a need to improve the performance of the Carry Select Adder.

The problem which is identified in the adder is a delay that may create due to the propagation of carry from primary to secondary units. Hence, to avoid this carry propagation delay there is a need to use the minimum amount of adder blocks in carry propagation model. Recently, the authors Bhatnagar *et al*., (2016) replaced the Binary to Excess converter with the modified D-Latch Enabled and named it as the BEC1 Carry-Select Adder. It is also one of the lagging concepts because the latch is also a storage device that stores the carry if the total circuits are in idle state. To avoid such effects, the adder process is to be selected under the same adder concept. Under this problem identification, the Kogge–Stone adder is selected for carry propagation units. The main aim of selecting Kogge-Stone is that it is one of the parallel computation methods that minimize the propagation delay of a carry bit.

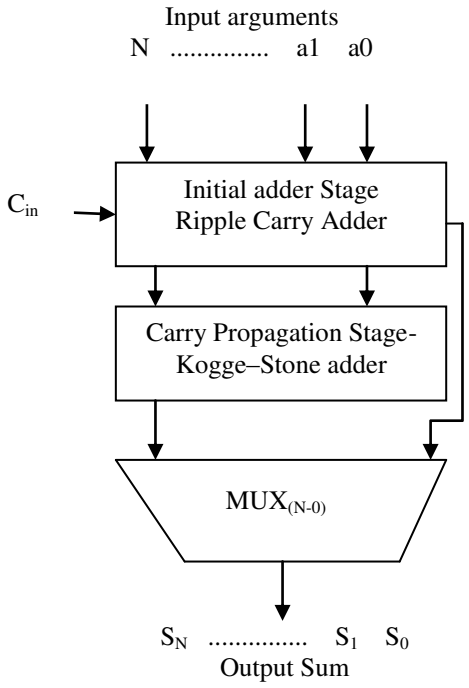

Figure-1. Representation of proposed basic building block.

The main contribution of this method is to implement the carry select adder with Kogge-Stone (KS) adder by replacing the existing adder. The representation of the carry select adder is shown in figure 1. The initial stage is same Ripple Carry Adder is used, for carry propagation model effective parallel prefix model is used with a Multiplexer. The input arguments are processed through the normal block. The KS adder utilized the carry

propagation units to transfer the carry. As per the Ghosh *et al*., (2008), the KS adder is a low fan out and utilizes the adders in various performances. It is well suited in for variable sized adder blocks.

In this paper, various adders are compared with the 8 bit and 16-bit combinations. It is measured to find the performance of the effectively optimized adder. The proposed KS-based CSLA is compared with the conventional RCA, Modified BEC, Regular SQRT adder and modified SQRT adder.

The rest of the paper is summarized as follows: In section 2, the detailed survey is made to get some knowledge on adders and find the exact problem definition of previous issues with its merits. The logics are reviewed in detail to manage the proposed design. The research methodology is focussed in section 3. The design and simulation made by cadence tool are displayed under section 4 with neat experimental results. The paper is summarized in section 5 and finally, some future enhancements are indicated in section 6.

2. REVIEW OF LITERATURE

Gaur *et al*., (2016) implemented various 16-bit adder, such as Carry skip, Ripple carry, Carry select, Carry save, Carry-look-ahead, kogge stone, Weinberger, Han-Carlson and Ling adder architectures. These processes are made to under 28nm Technology. The implementation made in Zynq-7000 FPGA board with the help of Xilinx design tool. Arunraj and Varatharajan (2016) replaced the carry propagation unit with the AND-OR-Inverter logic with the help of full adder designs.

Bahadori *et al*., (2017) modified a Carry Generation and Selection (CGS) scheme with the merged structure of carry generation and selection units. It is employed in the square-root CSLA (SQRT-CSLA) structure. The overall tests are carried out under HSPICE simulations in 45 nm bulk CMOS technology that combines varies bit level adders to verify the impact. Kaur *et al*., (2016) replaced a dual RCA with single RCA and single Binary to Excess-1 Converter (BEC). Many applications are prepared with the help of CLSA, Likewise, Kesava *et al*., (2016) designed a multiplier unit by replacing the CSLA with binary excess-1 counter mainly used to reduce the power consumption. Kumar and Krishnamurthy modified the procedure of CSLA, (i.e.,) the calculation of final stage is planned after carry select (CS) operation, in this process the redundant logic operations are eliminated in the conventional CSLA.

Akhter *et al.*, (2015) made a 16-bit Square Root Carry-Select Adder (SQRT CSA). They combined the 16-bit CSA architecture with five different stages namely, XOR gate, AND gate, 2:1 Mux, Half Adder (HA) and Full Adder (FA). These processes are implemented using CMOS transmission gate (CMOS TG). To verify the functionality of each stage it is verified with ripple carry adder and binary-to-excess converter under different bit sizes. The tests are done under Mentor Graphics Design Architect and simulations are carried out with TSMC 0.35μm CMOS technology with a supply voltage of 3.3 V.

Sukanya *et al*., (2016) implemented a FIR filter with the help of Carry Select Adder. Similarly, Reddy (2017) processed some multi-precision adders under Xilinx ISE14.7, also tested in SPARTAN3E, XC3S1600E with speed of -5. Lavanya *et al*., (2014) compared the Gate count of Regular and Modified 16-bit carry select adders with a Common Boolean Logic carry select adder. Their main motive is to select the Gate count evaluation of regular, modified and proposed designs are given in terms of INVERTER, NAND and NOR Gates.

Wey *et al*., (2012) processed an area efficient carry select adder in terms of sharing common Boolean logic term. The summation operation is carried out with sharing a partial circuit with a XOR gate and one inverter gate. The transistor count is reduced to 1947 to 960. Furthermore, the power consumption also reduced by 1.26mw to 0.37mw. Finally, the power delay product is reduced to 1.28mw*ns from 2.14mw*ns.

Preußer and Krause (2016) reviewed some adder's concepts. The SRAM based programmed FPGA consist of large logic depth that may induce by the great fan-in. The process is addressed by carry chains to establish direct links with configurable logic blocks. Finally, the process is much effective than the linear adder. The main motive of the process is to identify the appropriate wide adder implementation. The limitations occur due to fan-in of addition is the achievable clock rate. Some traditional methods and authors have analyzed the performance of the classic fast addition schemes and proposed similar adder architectures to perform fast carry chains. The survey states some more concepts are to be implemented and evaluated in FPGA devices. The evaluation is processed and synthesized for recent devices based on manufacturers. The analyses of the adder were considered from the open-source IP core with the library of Pile of Cores (PoC).

Xing, S., & Yu, W. W. (1998) stated that the linear nature of the carry chains creates hardly accessible for hierarchical connection schemes. Based on these studies the block-parallel adders may participate with the boosted RCA. If the architecture creates a logarithmic nature the delay must eventually manage with several combinations which are mentioned by Vitoroulis and Al-Khalili (2007). It results in some selection of parallel prefix networks with simple RCA with operands more than 250 bits.

Shubin *et al*., (2010) presented a method to analyze and compare the ripple carry full adders with speed factor based on the "Equal Delay-Capacity". The comparison is made with N-bit devices to verify the functionality. Similar to this process the comparison is merged from the finished N-bit adders for complete simulation cycle. It is mostly used based on the software products. Their method allows quickly and exactly to compare the speed of various one-bit adders cell for a unique application. Chudasama and Sasamal (2016)

proposed a 4-bit Vedic multiplier (using Urdhva Tiryagbhyam sutra) in QCA Technology. For many complex multiplier applications, the partial product unit is realized using the carry-save technique. The design is formulated under QCA Designer tool by maximizing the technology scaling level.

Vijaysri and Raja (2015) presented several models for efficient implementation of generic carry-save compressor trees on FPGAs. It is considered with critical path, individual bit width and practically no area overhead when comparing it with the Carry Propagation Adder (CPA) trees. To verify the functionality the classic carry-save compressor tree is presented with a novel linear array structure for effective carry-chain resources. The Hardware Description Language (HDL) code is generated based on the CPAs, and it may compatible with several FPGA families. The study discussed wide range of bit widths and a large number of operands. Javali *et al.,* (2014) processed high-speed carry save adder using carry look-a-head adder. Prashanth *et al.,* (2016) presented a Spurious Power Suppression Technique (SPST) technique with the process of various adders.

Rajkumar *et al.*, (2010) suggested an approach to improve addition process. Hence, to verify the developed process, it is compared with the 16, 32 and 64-bit adder combinations. In many applications, the addition is the only process to speed up the parallel process. The structure of carry select adder is improved by CLA it occupies more chip area. The process uses several pairs of RCA's to convey the partial sum and carry by considering Cin=0 and Cin=1. Due to this reason, the complexity of final adder gets complicated, to minimize the complexity the RCA (CLA) is replaced with C_{in}=1 with BEC logic. The area is reduced and pointed as an advantage but the delay is increased and limitation exists.

To significantly reduce the area and power of the CSLA the authors Ramkumar and Kittur (2012) uses a simple and efficient gate-level modification. Based on this modification 8-, 16-, 32-, and 64-b square-root CSLA (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. The proposed design has reduced area and power as compared with the regular SQRT CSLA with only a slight increase in the delay. This work evaluates the performance of the proposed designs in terms of delay, area, power, and their products by hand with logical effort and through custom design and layout in 0.18-µm CMOS process technology. The results analysis shows that the proposed CSLA structure is better than the regular SQRT CSLA.

3. RESEARCH METHODOLOGY

The problem identified in the adder concept is based on the speed and power. A researcher has an important role to extend or merge the traditional concepts and produce the optimized adder. In this research, the previous concepts are analyzed and noticed parallel adder with high accuracy. The Kogge–Stone adder is considered here because it is one of the parallel prefix form adder and it has very lower fan-out at each stage. Initially, Kogge–Stone adder is developed by Kogge and Stone (1973) [24]. Mainly, it is selected because of generating and propagating the signal that is created in the first stage itself. It is used to compute the intermediate carries in a parallel process to improve the computation speed. The final sum is a XOR of the carry and propagates signals. It is specifically selected with an attributes that are associated with a Kogge--Stone in low logic depth, high node count and minimal fan out. These parameters decide high node count and provide minimal fan out to maximize the performance.

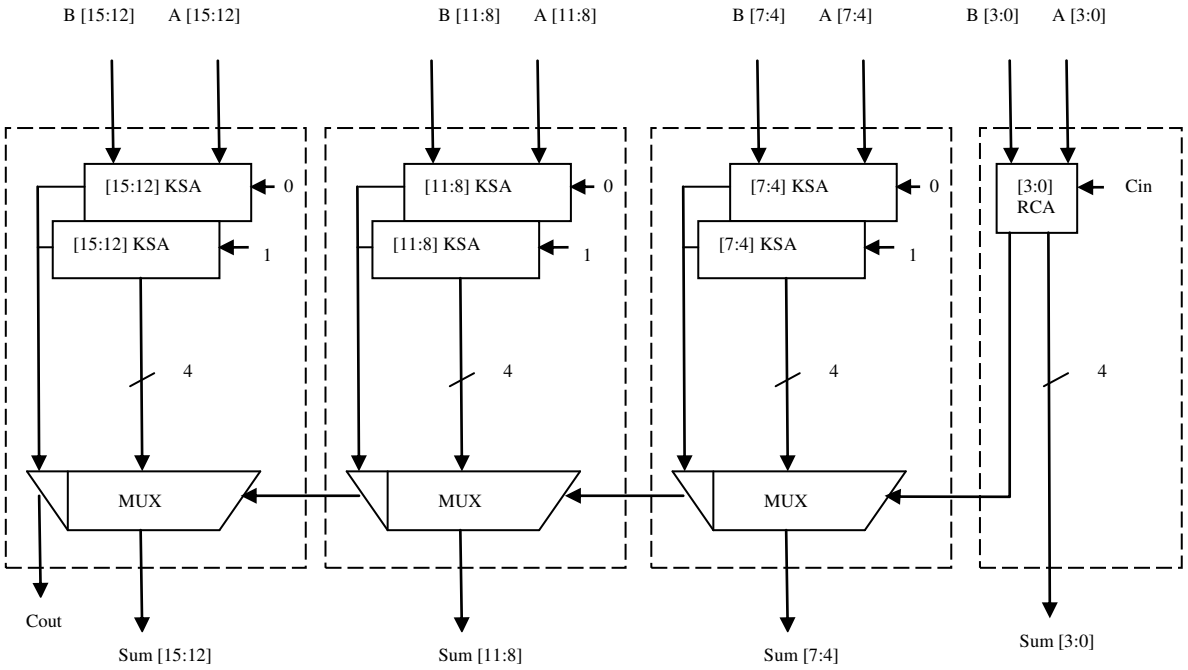

Figure-2. Architecture of proposed carry select adder based on Kogge–Stone adder.

As shown in Figure-2, the traditional RCA units and AOI units are replaced with the normal Kogge-Stone adder. Here the adder units are processed with parallel processing to minimize the delay. The initial stage is normally kept as a ripple carry adder with the internal blocks as full adders as basic components to initiate the computation. The internal blocks are represented as shown in the RTL schematic, which is represented in Figure-3. Similarly, the output representation of sum and carry in the first stage is given by,

$$Sum[0] = A[0] \oplus B[0] \oplus Cin \quad (1)$$

$$Sum[1] = A[1] \oplus B[1] \oplus Cc2 \quad (2)$$

$$Sum[2] = A[2] \oplus B[2] \oplus Cy0 \quad (3)$$

$$Sum[3] = A[3] \oplus B[3] \oplus Cyx \quad (4)$$

$$Cy1 = A[3].B[3] \oplus B[3].Cyx \oplus Cyx.A[3] \quad (5)$$

The simple 4 bit kogge stone adder is shown in figure 4, it consists of three stages namely Pre-processing, Carry generation network and Post processing. In initial pre-processing stage, the Generate (G) and propagate (P) signals of all the input pairs are generated based on the input signals A and B. The logical equations of the propagate and generate signals are represented by,

$$Pi = Ai \text{ xor } Bi \quad (6)$$

$$Gi = Ai \text{ and } Bi \quad (7)$$

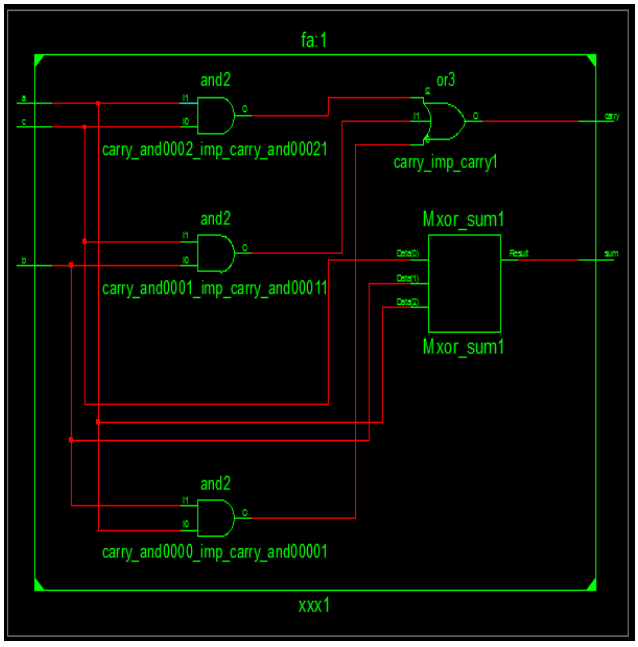

Figure-3. RTL schematic of full adder.

After completing the pre-processing stage the Carry Generation Stage is considered. It is the second stage of KSA unit; it separates the process in parallel format. The logical representation of these units is given below:

$$CPi{:}j = Pi{:}k + 1 \text{ and } Pk{:}j \quad (8)$$

$$CGi{:}j = Gi{:}k + 1 \text{ or } (Pi{:}k + 1 \text{ and } Gk{:}j) \quad (9)$$

Post Processing Stage gives the detail representation of KSA with its final summation and carry logics. It is given by,

Ci–1 = (Pi and Cin) or Gi (10)

Si= Pix or Ci–1 (11)

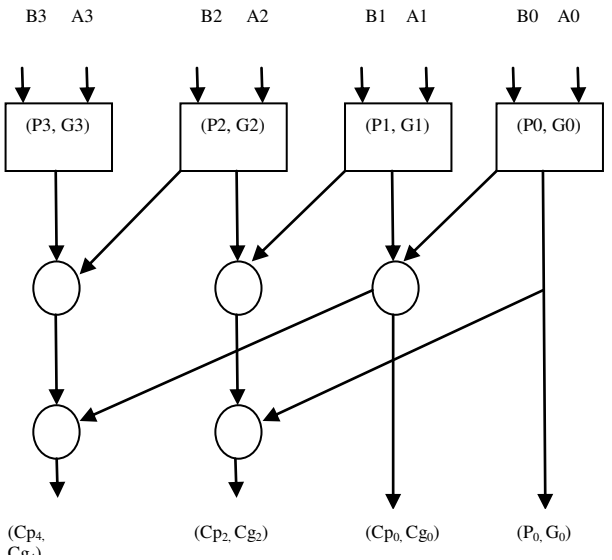

Figure-4. Bit Kogge-Stone adder.

To implement this process, Xilinx 14.5 is used, initially, the Verilog code is framed as per the proposed architecture; it is transferred into RTL schematic with proper syntax. After completing the synthesis the waveform verification is carried out with the given adder concept. It is carried through simulation tab in ISE Design Suite. The 4 bit kogge stone adder RTL schematic is represented in Figure-5.

The multiplexer unit is shown in figure 6 utilized for selecting the first stage or second stage results. If the input carry Cin is ideally zero then the first stage sum s1 [15:4] is selected from KSA unit and remaining s1 [3:0] from basic RCA unit from least Stage.

The propagation unit P is processed by the basic XOR function as p=a^b and its G term is g = a and b. the internal equations provide detail description in the form of carry and sum. Each stage is framed with respect to the carry unit. Its main objective is to speed up the carry from one stage to another through propagation unit.
For example:

cg[0]=(g[0]) (12)

cp[0]=(p[0]) (13)

Finally, the sum is calculated for simple four bit logic is given as Sum [3:0], it is given by,

sum[0]=p[0] xor cin (14)
sum[1]=p[1] xor c[0]
sum[2]=p[2] xor c[1]
carryout=c[3]
sum[3]=p[3] xor c[2]

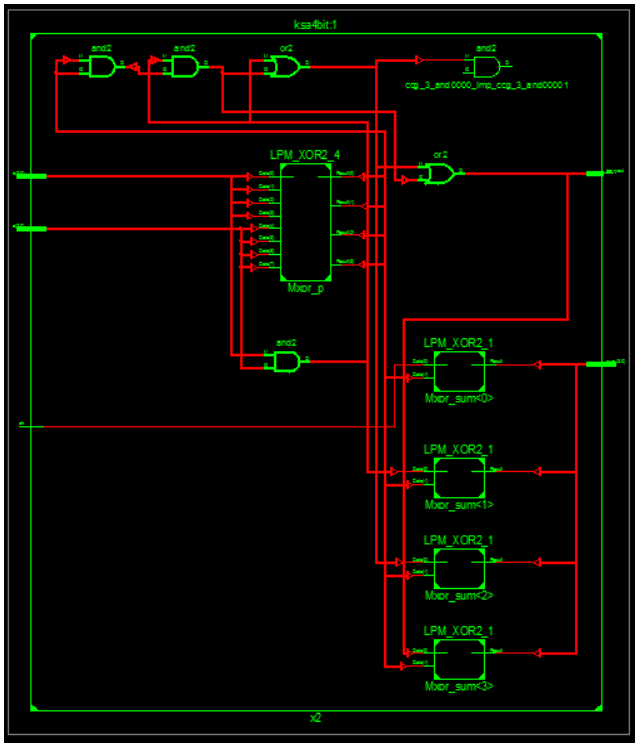

Figure-5. Four bit Kogee Stone adder.

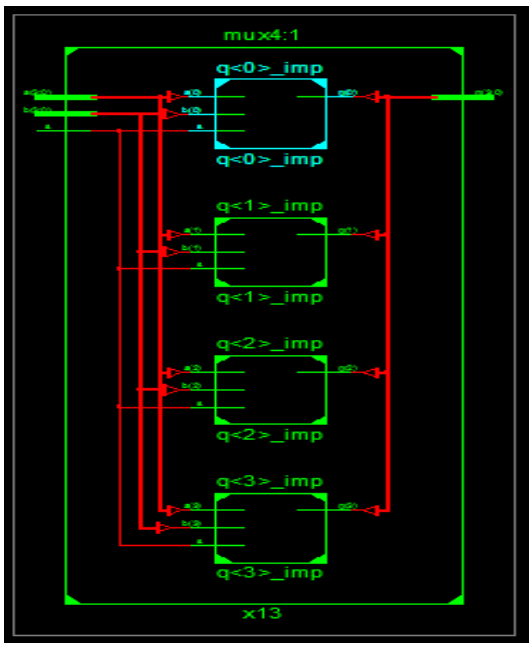

Figure-6. Multiplexer unit.

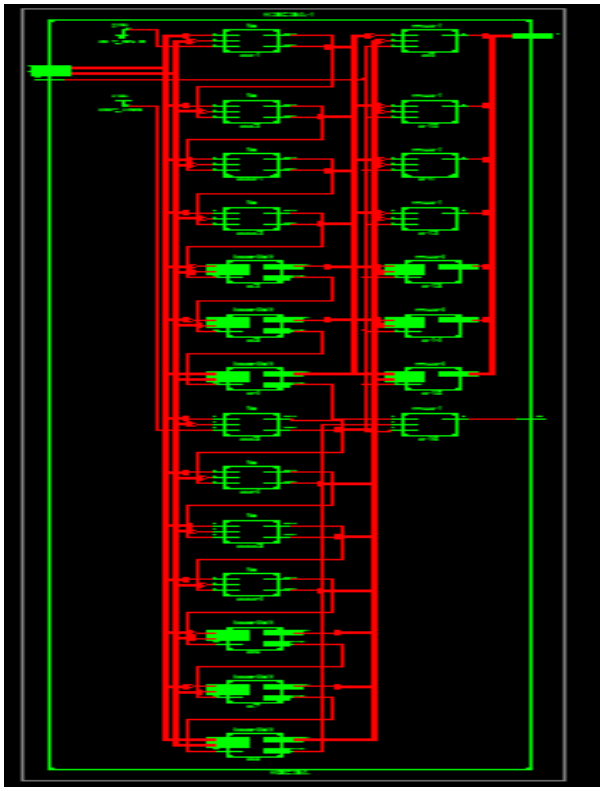

Figure-7. Schematic view of proposed KS-CLSA design.

The proposed carry select adder concept differs from traditional adder concepts. The parallel processing concepts are merging from the previous steps and it is grouped from the propagation and carry selection process. The RTL schematic of proposed method is shown in Figure-7. The major delay occurs in intermediate stages. The process is totally different from other stages. Initial stage RCA is used to process the initial carry Cin with real time-varying inputs. The process completely depends on internal logical blocks. Hence, it is important to reduce the internal carry flow path and make it's a least delay and power.

The gate count is calculated by number of gates used to construct the unit. For example, consider a single bit generation unit. Sum [1] =A [1] ⊕ B [1] ⊕ Cc2. Here the gate count is calculated by number of logics with the multiplication of gates. For each full adder there is 14 is required, as per the Table-1.

Table-1. Gate Count of basic gates.

	Inverter	**NAND**	**NOR**
AND	1	1	0
OR	1	0	1
XOR	5	2	1
2*1 MUX	4	2	1
Half Adder	6	3	1
Full Adder	8	4	2

Hence, the first stage consists of 4 full adders. Its gate count is 4(FA) = 4*14=56. Similarly, for Single 4 bit KSA unit need four AND gate. Totally, 76 optimized logics are displayed.

EXPERIMENTAL RESULTS

The simulated results are verified along with the basic truth table for each stage. Here A [15:0] and B [15:0] is considered as input sequence and the output Sum [15:0] and its Carry out (co) is displayed. To verify the functionality it is varied at the input stream. The experimental results prove that the proposed adder has 10.74ns delay which is 0.08ns minimized when comparing with the Modified SQRT (with M-AOI) adder. Similarly, the power is reduced by 35.95mW when comparing with conventional (Dual RCA). The circuit follows similar to conventional but in practically it follows Modified SQRT adder logic. The Power Delay Product (PDP) is shown in table II. the least PDP factor is 640.07mw-ns. It helps to reduce the computation process.

Table-2. Performances analysis comparisons.

Adder types	**Delay** (ns)	**Power** (mW)	**Power delay product** (mw-ns)
Conventional (Dual RCA)	14.62	95.60	1366.36
Modified conventional (with BEC)	16.52	85.32	1445.47
Modified Conventional (with AOI)	15.56	82.65	1286.034
Modified Conventional (with M-AOI)	13.89	80.95	1124.40
Regular SQRT (Dual RCA)	10.98	182	2300.56
Modified SQRT (with BEC)	12.25	175	2464.2
Modified SQRT(with AOI)	11.91	169	2012.79
Modified SQRT(with M-AOI)	10.82	151	1633.82
Proposed Kogge–Stone CSLA (KS-CSLA)	10.74	59.65	640.07

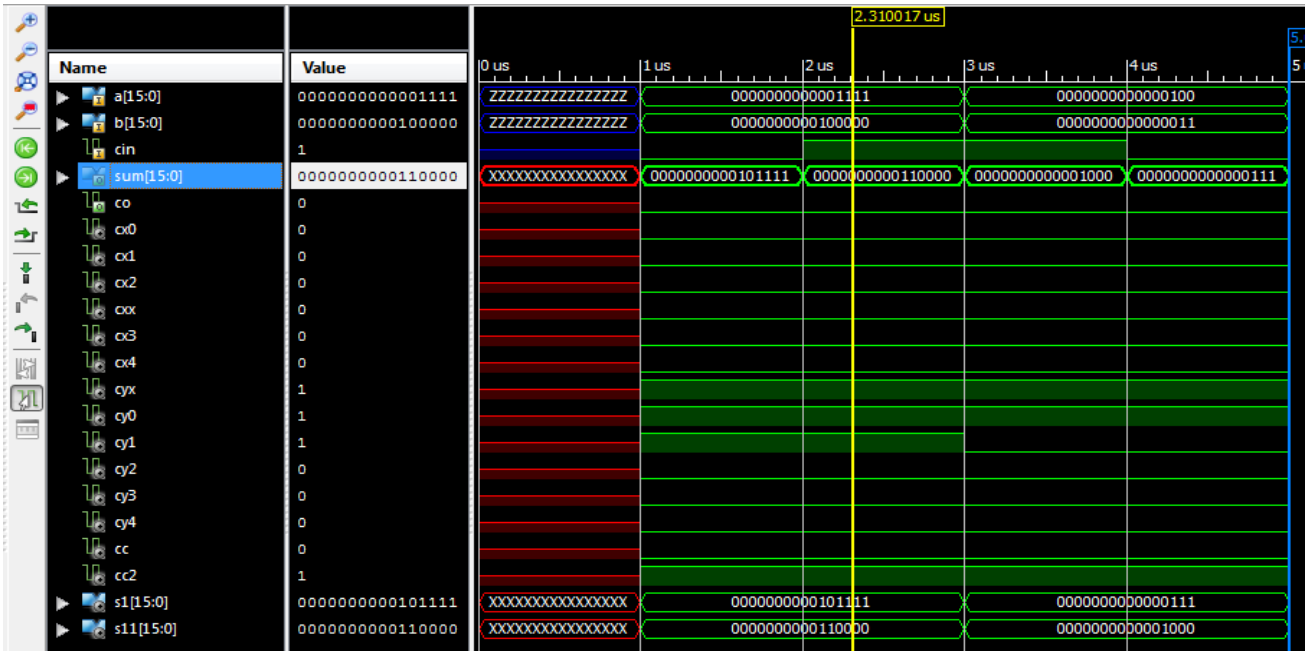

Figure-8. Simulation results for proposed KS-CSLA adder.

The general concept is drawn from this experimental result and it is suggested with a new model with KSA adder. The main objective of this research is solved by implementing the adder block in suitable low power carry select adder unit. This research investigates and examines the fast KS adder schemes. A low power, area efficient and high-speed adder with A [15:0] and B [15:0] inputs have been processed in this paper. The designs of various adders have been implemented previously with Xilinx 14.5 software and Xilinx ISIM simulator for simulation.

5. CONCLUSIONS

A study has been reviewed based on carry select adder with different techniques. From the review, it is noticed that the internal carry propagation process is carried out by AOI logics and RCA schemes. Hence, a traditional method has some limitation like delay and power. The proposed method works under the concept of kogge stone adder with single ripple Carry adder and multiplexer logics. The experimental results prove that the proposed method has improved the speed of 0.08ns delay. The power consumption rate is reduced to a great extent by the parallel prefix technique is used for proposed unit. Further, comparison of different adders and multipliers are done on various performance measures. Finally, the adder topologies illustrate that the proposed Kogge Stone-Carry Select Adder (KS-CSA) is better among all other adders.

There is a need to extend this research in future to different abstraction level, with a detailed power calculation the process is to be carried out. Need to compare all the parameters like throughput of each stage and concentrate particularly on power optimization.

REFERENCES

Bhatnagar S., Gupta H. & Jain S. 2016. Modified D-Latch Enabled BEC1 Carry-Select Adder with Low Power-Delay Product and Area Efficiency. In Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing (pp. 441-449). Springer India.

Ghosh S., Ndai P., & Roy K. 2008, March. A novel low overhead fault tolerant Kogge-Stone adder using adaptive clocking. In Design, Automation and Test in Europe, 2008. DATE'08 (pp. 366-371). IEEE.

Gaur N., Tyagi D. & Mehra A. 2016. September. Performance comparison of adder architectures on 28nm FPGA. In Advances in Computing, Communication, & Automation (ICACCA)(Fall), International Conference on (pp. 1-5). IEEE.

Arunraj G. V. & Varatharajan R. 2016. Implementation of an efficient Carry Select Adder. International Journal of Applied Engineering Research. 11(13): 7989-7991.

Bahadori M., Kamal M., Afzali-Kusha A. & Pedram M. 2017. An energy and area efficient yet high-speed square-root carry select adder structure. Computers & Electrical Engineering. 58: 101-112.

Kaur G., Goyal C. & Singh K. 2016. Design of 64-bit hybrid carry select adder using CMOS 32nm Technology.
Kesava R. B. S., Rao B. L., Sindhuri K. B. & Kumar N. U. 2016, June. Low power and area efficient Wallace tree multiplier using carry select adder with binary to excess-1

converter. In Advances in Signal Processing (CASP), Conference on (pp. 248-253). IEEE.

Kumar D. & Krishnamurthy P. 2016. Design of Area Efficient and Power Optimized Carry Select Adder.

Akhter S., Chaturvedi S. & Pardhasardi K. 2015, February. CMOS implementation of efficient 16-Bit square root carry-select adder. In Signal Processing and Integrated Networks (SPIN), 2015 2nd International Conference on (pp. 891-896). IEEE.

Sukanya S. L. & Rao N. M. L. R. 2016. Design of FIR Filter using Efficient Carry Select Adder.

Reddy A. R. 2016, January. Multi precision arithmetic adders. In Computer Communication and Informatics (ICCCI), 2016 International Conference on (pp. 1-6). IEEE.

Lavanya M. 2014. Gate count comparison of Different 16-bit Carry Select Adders. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 3(7): 10846-10853.

Wey I. C., Ho C. C., Lin Y. S. & Peng C. C. 2012, March. An area-efficient carry select adder design by sharing the common Boolean logic term. In Proceedings on the International Multiconference of Engineering and computer scientist, IMECS.

Preußer T. B. & Krause M. 2016, November. Survey on and re-evaluation of wide adder architectures on FPGAs. In ReConFigurable Computing and FPGAs (ReConFig), 2016 International Conference on (pp. 1-6). IEEE.

Xing S. & Yu W. W. 1998. FPGA adders: Performance evaluation and optimal design. IEEE Design & Test of Computers. 15(1): 24-29.

Vitoroulis K. & Al-Khalili A. J. 2007, August. Performance of parallel prefix adders implemented with FPGA technology. In Circuits and Systems, 2007. NEWCAS 2007. IEEE Northeast Workshop on (pp. 498-501). IEEE.

Shubin V. V. 2010, June. Analysis and comparison of ripple carry full adders by speed. In Micro/Nanotechnologies and Electron Devices (EDM), 2010 International Conference and Seminar on (pp. 132-135). IEEE.

Chudasama A. & Sasamal T. N. 2016, April. Implementation of 4× 4 vedic multiplier using carry save adder in quantum-dot cellular automata. In Communication and Signal Processing (ICCSP), 2016 International Conference on (pp. 1260-1264). IEEE.

Vijaysri G. M. & Raja A. 2015, March. Implementations of redundant carry save adders on FPGA. In Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on (pp. 1-9). IEEE.

Javali R. A., Nayak R. J., Mhetar A. M. & Lakkannavar M. C. 2014, November. Design of high speed carry save adder using carry lookahead adder. In Circuits, Communication, Control and Computing (I4C), 2014 International Conference on (pp. 33-36). IEEE.

Prashanth A., Waran R. P., Khandekar S. & Pawar S. 2016. Low Power High Speed based Various Adder Architectures using SPST. Indian Journal of Science and Technology. 9(29).

Rajkumar B., Kittur M. & Kanna P. 2010. ASIC implementation of modified faster carry select adder. Eur. j. sci. res. 42(1): 53-58.

Ramkumar B. & Kittur H. M. 2012. Low-power and area-efficient carry select adder. IEEE transactions on very large scale integration (VLSI) systems. 20(2): 371-375.

Kogge P. M. & Stone H. S. 1973. A parallel algorithm for the efficient solution of a general class of recurrence equations. IEEE transactions on computers. 100(8): 786-93.