ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

Design and Analysis of Carry Select Adder Using Kogge Stone Adder

K. Swetha[1], M. Athappan[2]

PG Scholar, Department of ECE, SVCE, Sriperumbudur, India[1]

Assistant Professor, Department of ECE, SVCE, Sriperumbudur, India [2]

ABSTRACT: Carry select adder (CSLA) is considered as the one of the fastest adder and used in many data processing applications to perform the fast arithmetic functions. The speed of operation of the CSLA is limited by the time required to propagate a carry through the adder. This paper discusses about the implementation of 16-bit Linear Carry Select Adder (LCSLA) with Kogge Stone Adder (KSA) in terms of Binary to Excess- 1 converter (BEC) and extra logic gates. A high speed adder is designed by merging the CSLA and KSA algorithms instead of using ripple carry adder (RCA) and due to that the delay is reduced, but it occupies more area. Synthesis and simulation result of CSLA with KSA is done by using Xilinx ISE 13.3 and cadence EDA tool and compared interms of ADP and PDP.

KEYWORDS- Kogge Stone Adder (KSA), Carry Select Adder(CSLA), Binary to Excess-1 converter (BEC),Ripple Carry Adder (RCA), Linear Carry Select Adder (LCSLA).

I. INTRODUCTION

Area and power reduction in data path logic systems are the main area of research in VLSI system design. Adder is the building block of the applications of VLSI, Digital Signal Processing, Image Processing. Even though Ripple Carry Adder structure is simple, Carry Propagation Delay (CPD) is more for wide bit adders. To reduce the Carry Propagation Delay, Carry Select Adder is introduced. Digital computers perform a variety of information – processing tasks. The basic arithmetic operation is the addition of two binary digits. This addition consists of four possible elementary operations such as 0+0=0, 0+1=1, 1+0=1, 1+1=10. The first three operations produce a sum of one digit, but when both augend and addend bits are equal to 1, the binary sum consists of two digits. The higher significant bit of this result is called a carry. When the augend and addend numbers contain more significant digits, the carry obtained from the addition of two bits is added to the next higher order pair of significant bits. The details of Ripple Carry Adder,Kogge Stone Adder, Carry Select Adder, and Binary to Excess-1 Converter are discussed in section II, and the implementation of proposed method is described in section III. The performance and simulation results were presented and discussed in section IV.

II. CARRY SELECT ADDER

A carry select adder is a particular way to implement an adder, which is a logic element that computes the (n+1) bit sum of two n- bit numbers. The carry select adder is simple but rather fast. The carry select adder generally consist of two ripple carry adders and multiplexer. The Carry Select adder performs the operation simultaneously for Cin=0 and Cin=1. After the two results are calculated, the correct sum as well as the correct carry, is then selected with the multiplexer once the correct carry is known. The block diagram of Carry Select Adder is shown in Figure2.1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

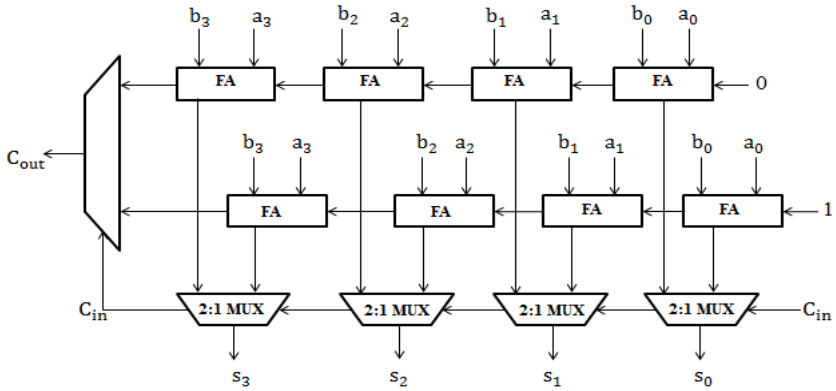

Figure 2.1: Block diagram of Carry Select Adder

The carry select adder can be classified in to two types:

1. Linear Carry Select Adder
2. Square – Root Carry Select Adder

The Linear Carry Select Adder consist of equal length adder stages of RCA with Cin=0 and Cin=1. The multiplexer is used to select the final sum and carry bit obtained from the RCA blocks. The linear carry select adder is an adder that is made to reduce the calculation time by having the carry be the limiting delay factor. It accomplishes the addition by adding small portions of bit (each of equal size) and wait for the carry to complete the calculation.

The Square-Root Carry Select Adder consists of unequal length adder stages. This is an extension of the linear carry select adder that improves the delay time greatly. If we use the square root carry select adder, the time can be improved, as the time waiting for the carry bit is used to calculate an extra input bit in each stage. The multiplexer is used to select the final sum and the carryout bit.

A. RIPPLE CARRY ADDER

A ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers .It can be constructed with full adders connected in cascade, with the output carry from each full adder connected to the input carry of the next full adder in the chain.

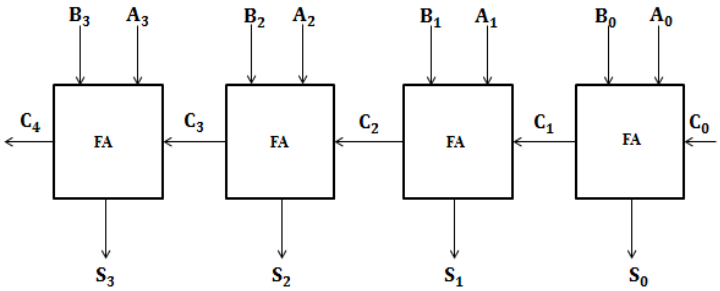

Figure 2.2: 4 bit ripple carry adder

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

Figure 2.2 shows that the interconnection of four full adder circuits to provide a four-bit binary ripple carry adder. The augend bits of A and the addend bits of B are designated by subscript the numbers from right to left with subscript 0 denoting the lest significant bit . The carries are connected in a chain through the full adders. The input carry to the adder is C0 and it ripples through the full adders to the output carry C4. The S outputs generate the required sum bits.

B. KOGGE STONE ADDER

Kogge stone adder is the derivative part of the carry look ahead ahead adder. It is also known as carry tree adder and broadly considered as one of the fastest addition method. Carry generation is much faster because of the parallel computation. It generates carry in the O (log n). The complete function of Kogge Stone Adder can be classified in to three stages:

Pre – Processing:
This step involves computation of generate and propagate signals corresponding too each pair of bits in A and B. These signals are given by the logic equations below:

$$P_i = A_i \text{ xor } B_i \text{ - - - - - - - - - - - - (2.1)}$$

$$G_i = A_i \text{and } B_i \text{ - - - - - - - - - - - - (2.2)}$$

Carry generation stage:
It is used to generate the carry for the next stages. It uses group propagate and generate as intermediate signals which are given by the logic equations:

$$G_{black} = (G_{prev} \cdot P) + G \text{ - - - - - (2.3)}$$
$$P_{black} = (P_{prev} \cdot P) \text{ - - - - - - - - (2.4)}$$
$$G_{grey} = (G_{prev} \cdot P) + G \text{ - - - - - -(2.5)}$$

Post processing:
It is used to compute the sum bits. The logic equation of the sum bit signals are given below:

$$S_i = P_i \text{ xor } C_{i-1} \text{ - - - - - - - - - - - - (2.6)}$$

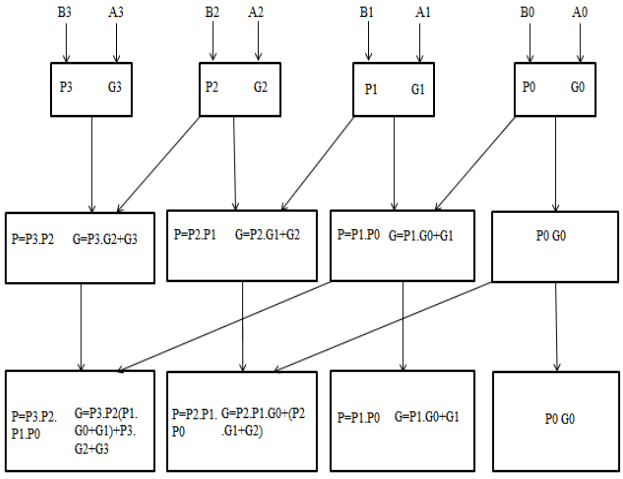

Figure 2.3: Block diagram of 4 bit kogge stone adder

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

The block diagram of Kogge Stone Adder is shown in Figure 2.3. The delay of structure is given by log (n). The KSA has the area (number of "*o*"operators) of *(n*log2n)-n+1* where *n* is the number of input bits.

C. BINARY TO EXCESS 1 CONVERTER

It is used to add 1 to the input bit. It can be logically converted based on the inputs at a gate.

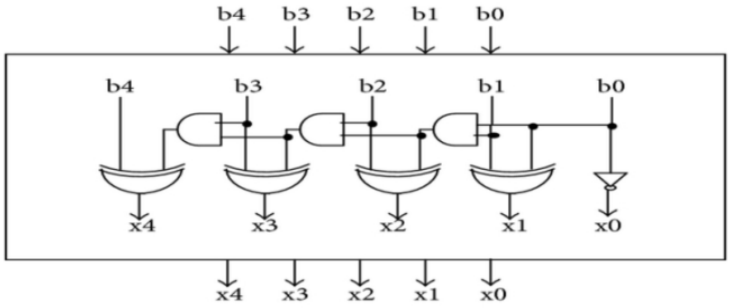

Figure 2.4: Binary to Excess 1 converter

III. PROPOSED METHOD

A high speed adder is designed by merging the Carry Select Adder and the Kogge Stone Adder algorithm instead of using Ripple Carry Adder and due to that the delay is reduced.

A. CSLA with KSA Implementation

Two different methods used for realizing the carry select adder using the kogge stone adder:

In **first method**, consider Cin=0 and Cin=1 and the simulation output was generated by usingExcess-1-adder as shown in Figure3.1. Final sum and carry was generated by using MUX.

Carry Equations:

$C_0=G_0$ - - - - - - - - - - - - - - - - - (3.1)
$C_1=P_1.G_0+G_1$ - - - - - - - - - - - - - - - - -(3.2)
$C_2=P_2.P_1.G_0+(P_2.G_1+G_2)$ - - - - - (3.3)
$C_3=P_3.P_2.C_1+(P_3.G_2+G_3)$ - - - - - - - - (3.4)

Sum Equations:

$S_0= P_0$- - - - - - - - - - - - - - - - - (3.5)
$S_1= P_1$ xor C_0- - - - - - - - - - - - - - - (3.6)
$S_2= P_2$ xor C_1- - - - - - - - - - - - - (3.7)
$S_3= P_3$ xor C_2 - - - - - - - - - - - - - - - - - (3.8)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

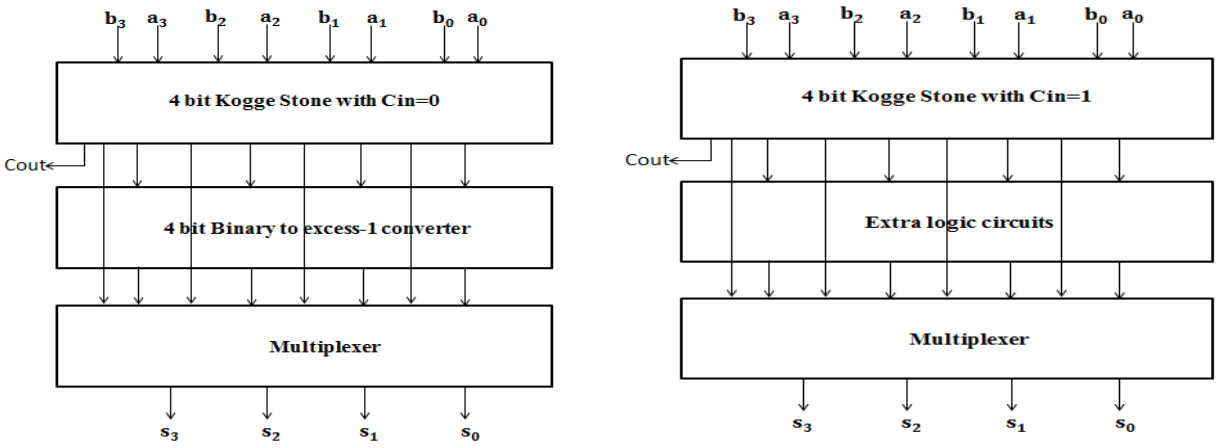

Figure 3.1: CSLA with KSA for Cin=0Figure 3.2: CSLA with KSA for Cin=1

Here area overhead is coming only in the final stage where the final sum and carry output are selected using a MUX [12]. In **second method,** consider Cin=1 and Cin=0 then the simulation output was generated by using the extra logic gates as shown in Figure 3.2. Finally sum and carry outputs is selected by using a MUX. Here also area overhead is coming only in the final stage where the final sum and carry output are selected using a MUX [12].

Carry equations:

$C_0=G_0+P_0$ - (3.9)

$C_1= (P_1.G_0+G_1) + P_1.P_0$ - - - - - - - - - - - - - --(3.10)

$C_2= P_2.P_1.(G_0+P_0) +(P_2.G_1+G_2)$ - - - - - - - (3.11)

$C_3= P_3. P_2.C_1+(P_3.G_2+G_3)+ P_3.P_2.P_1.P_0$ - --(3.12)

Sum equations:

$S_0= P_0$ - (3.13)

$S_1= P_1$ xor C_0 - - - - - - - - - - - - - - - - - - -(3.14)

$S_2= P_2$ xor C_1 - - - - - - - - - - - - - - - - - - - (3.15)

$S_3= P_3$ xor C_2 - - - - - - - - - - - - - - - - - - - (3.16)

B. 16-bit adder without & with Fast Carry Logic Implementation

The 16 bit adder is divided in to four 4 bit CSLA for Cin=0 and Cin=1and it is shown in Figure3.3& Figure 3.4. The output of the each 4 bit CSLA will provide the input to the each mux. Here MUX is used to select the final sum and the carry out bits and it's depends on the Cin which is carried out from the Carry Select Adder.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

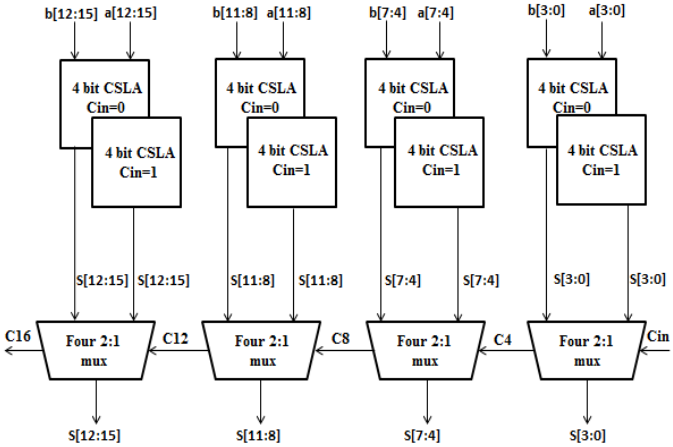

Figure 3.3: 16 bit adder without fast carry logic

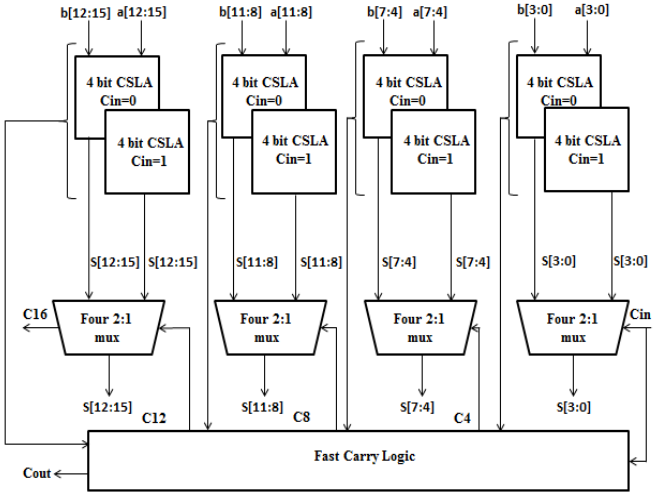

Figure 3.4: 16-bit adder with Fast Carry Logic

Since the carry passes through all the multiplexers the delay will be increased. To overcome this problem, the Fast Carry Logic can be produced from the carry output of each stage. Here Fast Carry Logic uses the Kogge Stone Tree to implement the intermediate carries [12].Fast Carry Logic improves the speed by reducing the amount of time required to determine the carry bits.

C. Implementation of 16-bit adder without & with Fast Carry Logic for KSA with BEC based CSLA for Cin=0

The 16 bit adder is divided in to four 4 bit KSA for Cin=0 adder shown in Figure 3.5& Figure 3.6. The output of the each 4 bit KSA with BEC based CSLA forCin=0 will provide the input to the each mux.

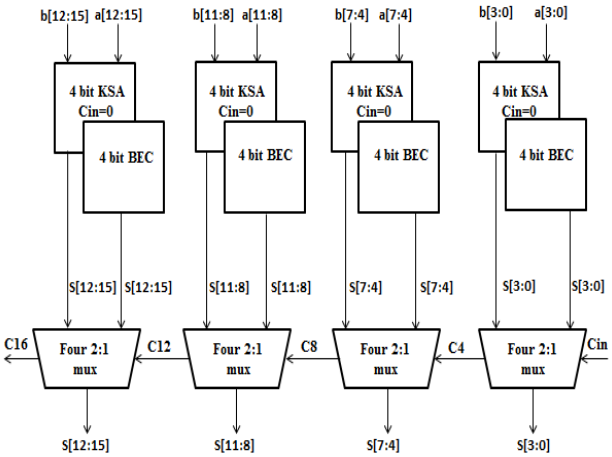

Figure 3.5: 16-bit adder without Fast Carry Logic for KSA with BEC based CSLA for Cin=0

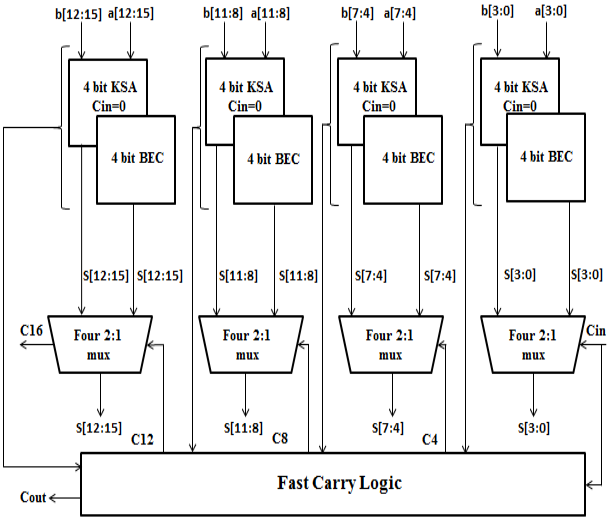

Figure 3.6: 16-bit adder with Fast Carry Logic for KSA with BEC based CSLA for Cin=0

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

D. *Implementation of 16-bit adder without & with Fast Carry Logic for KSA based CSLA for Cin=1 using Extra logic gate.*

The 16 bit adder is divided in to four 4 bit KSA for Cin=1 adder and it is shown in Figure 3.7 & Figure 3.8. The output of the each 4 bit KSA using extra logic gates for Cin=1 will provide the input to the each mux**.**

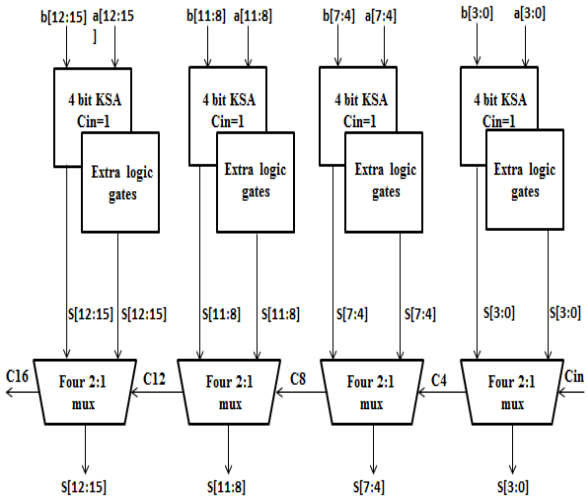

Figure 3.7: 16-bit adder without Fast Carry Logic for KSA based CSLA for Cin=1 using Extra logic gates

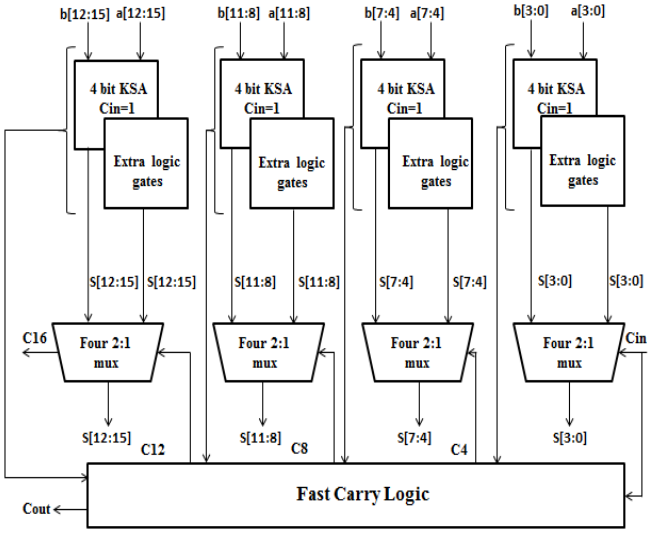

Figure 3.8: 16-bit adder with Fast Carry Logic for KSA based CSLA for Cin=1 using Extra logic gates

IV. SIMULATION RESULTS USING Xilinx TOOLS

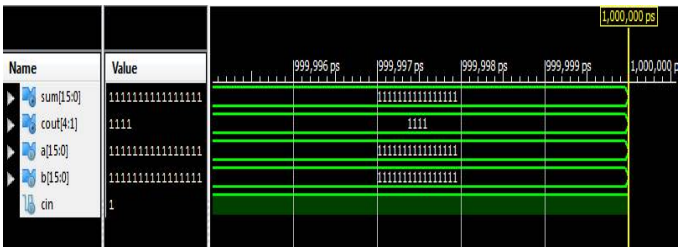

Figure 4.1: 16 bit adder without fast carry logic

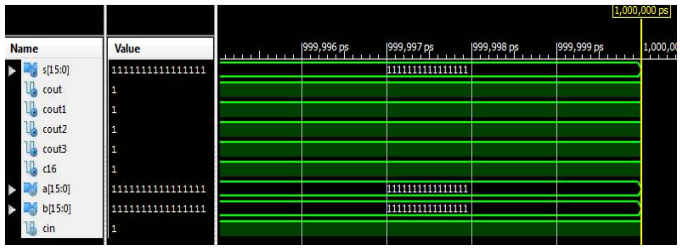

Figure 4.2: 16 bit adder with fast carry logic

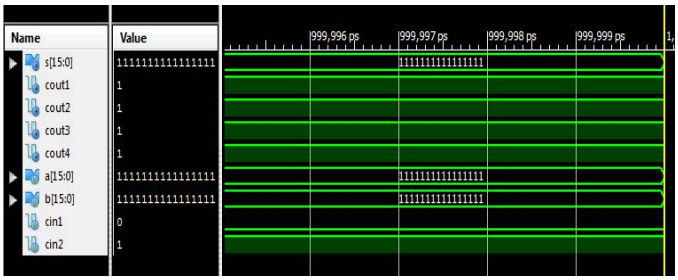

Figure 4.3: 16-bit adder without Fast Carry Logic for KSA with BEC based CSLA for Cin=0

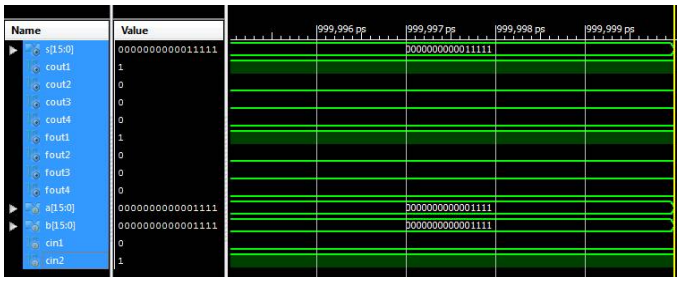

Figure 4.4: 16-bit adder with Fast Carry Logic for KSA with BEC based CSLA for Cin=0

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

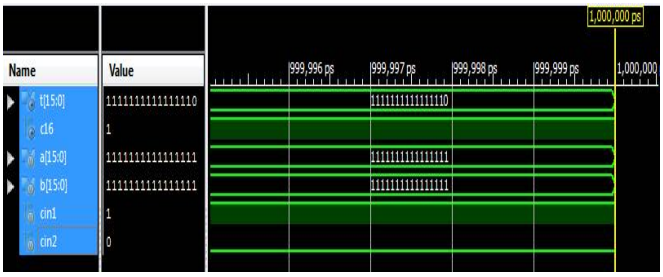

Figure 4.5: 16-bit adder without Fast Carry Logic for KSA with Cin=1 Using Extra logic gates

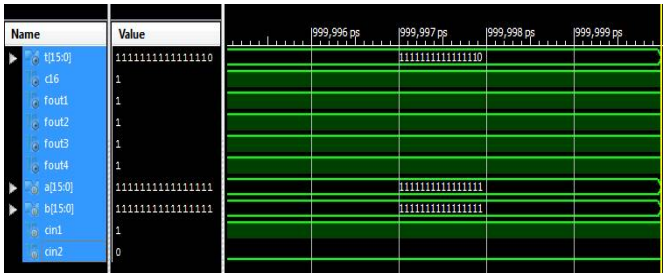

Figure 4.6: 16-bit adder with Fast Carry Logic for KSA with Cin=1 Using Extra logic gates

TABLE I

Method	Area (μm^2)	Delay (µs)	Power (nw)	ADP	PDP
CSLA without Fast Carry Logic	2794.18	2.72450	137104.59	0.007612	373.54
CSLA with Fast Carry Logic	3273.18	1.56040	158200.37	0.005107	246.85
16-bit adder without Fast Carry Logic for KSA with BEC based CSLA for Cin=0	1643.24	1.98670	89294.14	0.003264	177.40
16-bit adder with Fast Carry Logic for KSA with BEC based CSLA for Cin=0	1796.26	1.74910	98201.98	0.003141	171.76
16-bit adder without Fast Carry Logic for KSA based CSLA for Cin=1 using Extra logic gates	1463.62	2.16270	84380.91	0.003165	182.49
16-bit adder with Fast Carry Logic for KSA based CSLA for Cin=1 using Extra logic gates	1646.57	1.96250	91927.37	0.003231	180.40

By comparing the Area-Delay product (ADP) and Power delay product (PDP) for different methods. From the above table the 16-bit adder with Fast Carry Logic for KSA with BEC based CSLA for Cin=0 method gives less ADP and PDP compared to the other methods. The ADP obtained for this method is about 0.003141 and the PDP obtained for this method is about 171.76.

V. CONCLUSION

In this paper, 16-bit carry select adder with kogge stone adder is implemented and the designed is verified for the reduction in ADP and PDP. Two different methods of implementation are performed. In first method, KSA with BEC based CSLA for Cin=0 is generated. Mux is used to select the final sum and the carry bit .In second method, KSA based CSLA for cin=1 is generated by using Extra logic gates. Here also mux is used to select the final sum and the carry bit. Results shown that 16-bit adder with FCL for KSA with BEC based CSLA for Cin=0 provides better performance in terms of ADP and PDP and it has 2% efficiency in ADP and 7% efficiency in PDP than other methods.

VI. FUTURE SCOPE

Here design and analysis of Carry Select Adder using Kogge Stone Adder has been proposed. The synthesis and simulation result for different methods of the proposed system has been shown and compared interms of ADP and PDP. Among those methods 16-bit adder with FCL with BEC based CSLA for Cin=0 shows less amount of ADP and PDP. The performance of CSLA can be further improved by using Compressor circuits.

REFERENCES

1. Anand N, George Joseph, Johny S Raj, Jayakrishnan.P, (2013) 'Implementation of adder structure with Fast Carry Network for High Speed Processor' ,IEEE International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp. 188-190.
2. Alioto M (2004) 'A Gate Level Strategy to Design Carry Select Adder', ISCAS, pp. 465-468.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 3, April 2017

3. Alexander E.Shapiro, Francois Atallah, Kyugseok Kim, JihoonJeong, Jeff Fischer, Eby G. Friedman (2016) 'Adaptive power gating of 32-bit kogge stone adder', ELSEVIER, pp. 80 – 87.
4. Basant Kumar Mohanty and Sujit Kumar Patel (2014) , ' Area – Delay – Power Efficient Carry Select Adder', IEEE Transactions on circuits and systems – II, vol.61, No.6,pp. 418 – 422.
5. Bedrij O.J. (1962) , ' Carry Select Adder', IRE Transactions on Electronic computer vol.EC – II, No.3,pp. 340 – 344.
6. Feng Liu et.al (2009) 'A Compartitive Study of Parallel Prefix Adder in FPGA Implementation of EAC', 12th Euro micro Conference on Digital System Design and Architecture, Methods and Tools, IEEE conference, pp. 281-286.
7. I – Chyn Wey, Cheng – Chen Ho, Yi – Sheng Lin and Chien – Chang Peng(2012) , ' An Area Efficient Carry Select Adder Design by Sharing Common Boolean Logic term', Proc. on IMECS 2012,pp.1-4.
8. KuldeepRawat (2002), ' A Low Power and Reduced Area Carry Select Adder', IEEE conference , pp.467-470 .
9. Matthew M.Ziegler and MirceaR.Stan(2004) 'A Unified Design Space For Regular Parallel Prefix Adders' , proceedings of the Design , Automation and Test in Europe Conference and Exhibition.
10. Mano Morris M and Ciletti Michael D (2007), 'Digital Design', Pearson Education Inc., India.
11. Ramkumar B and Harish M Kittur (2011), 'low power and area efficient carry select adder', IEEEtransactions on VLSI Systems ,vol 2, pp. 371-375.
12. Sajesh Kumar Ua, Mohamed Salih K Kb (2012), 'Efficient Carry Select Adder Design for FPGA Implementation', ELSEVIER vol 30, pp.449-456.
13. Vinod Kumar Naik M and Mohammed Aneesh Y (2015), 'Design Of Carry Select Adder for Low – Power and High Speed VLSI Applications', IEEE conference.