Synthesis Of Asynchronous Carry Select Adder In FPGA

[1]Senthil Kumar J, [2]Gayathri S

[1]Assistant Professor, [2]PG Student

Department of Electronics and Communication Engineering,

Mepco Schlenk Engineering College (Autonomous),Sivakasi

***Abstract* — This paper describes an area efficient Asynchronous Carry Select Adder (ACSLA) implemented in a Field Programmable Gate Array (FPGA). CSLA is one of the fastest adders used in many data- processing systems to perform fast arithmetic operations and also has small areas and longer delay Ripple Carry Adder (RCA). The main scope of the proposed work is to reduce the area occupied by implemented CSLA in the FPGA. This paper presents an Asynchronous CSLA with Strong indication, early output and weak indication regimes realized using a delay insensitive dual rail code for data representation and processing. It also deals with a 4-phase return to zero protocol for handshaking. The simulation results of the 8, 16 and 32 bit ACSLAs synthesized in FPGA device are compared with its equivalent synchronous implementation. It is being observed that the ACSLA outperforms the synchronous CSLA in terms of area occupied and delay encountered in the logic circuit.**

Keywords— CSLA; ACSLA; FPGA; RCA .

I. INTRODUCTION

The most of the digital circuits are designed as synchronous circuits. But asynchronous circuits provide some advantages like low power consumption, no clock skew and also reduce electromagnetic interference. Carry Select adder is mostly used in high speed arithmetic operation ,Which is used to reduced area of RCA and improve high speed performance of carry look ahead adder(CLA) [1,2]. Robust asynchronous design employ delay insensitive code for data representation processing, 4-phase return to zero protocol for handshaking, there is no experimental work available in the asynchronous CSLA except theoretical content [3-10]. The method uses Muller C- element using 32bit asynchronous CSLA Which is consistent to different timing regimes such as strong indication, weak indication and early output. This is implemented in FPGA device and compared using mathematical calculation.

The rest of the paper is organized as follows. Preliminary plan of asynchronous circuit design have been discussed in section 2. Section 3 discusses the details of asynchronous CSLA architecture. The experimental result has been discussed in section 4. Section 5 concludes the paper with the summary of the carried out work.

II. ASYNCHRONOUS PRELIMINARY PLAN

An asynchronous logic block consists of asynchronous digital system is the combinational logic that is similar to synchronous digital system [10]. Delay insensitive data codes and 4 phase handshake protocol are used to build asynchronous logic block.

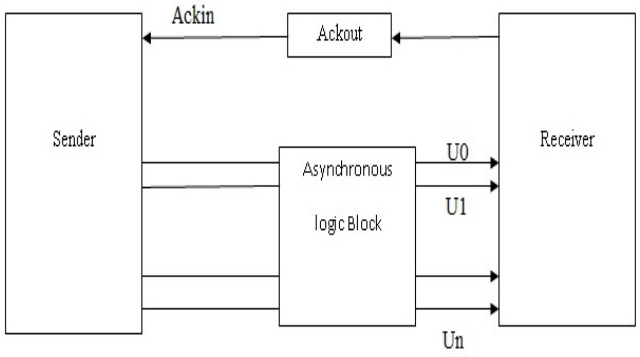

Fig. 1. Block diagram of sender and receiver.

The dual rail code is represent the delay insensitive data codes[10], in which a data wire U is divided into U0 and U1 as shown in Fig 1. U=0 is represented by U1=1 and U0=0, and U=1 is represented by U1=0 and U0=1. These two conditions are widely used in asynchronous circuit. This is referred to as valid data, and the condition of both U0 and U1 are 0. This is referred to as the spacer data. Another one condition is both of U0 and U1 are1 those state represent as illegal state. The 4 phase return to zero handshaking inputs are already predefined in valid data spacer-valid data spacer.

An asynchronous logic stage correlated with the sender and receiver technique. The valid data and spacer data operations of asynchronous circuit followed by predefined input sequence of valid data-spacer-valid data-spacer are defined in [10]. In Fig.2 the junction point represent isochronic forks. That the isochronic fork used to compromise the delay insensitivity and an isochronic fork refer signal transition occurrence of each net.

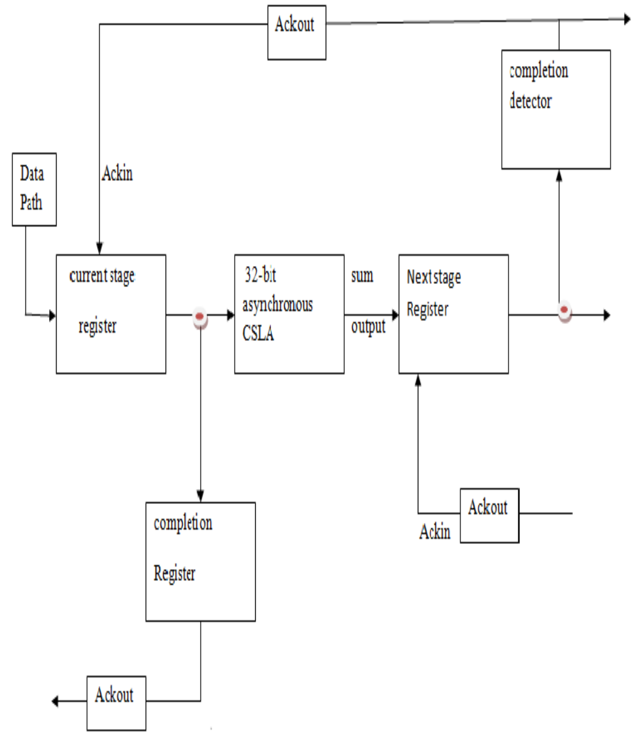

Fig. 2. Block diagram of asynchronous circuit stage

The dual rail data and the common acknowledge input feeds to the current stage register is initially present in the spacer state (i.e.,zero state) [11]. Then the Current stage register acknowledge value is binary 1 since the common acknowledge output these ackout is provided by the next stage register then it has binary value of 0. Now current stage register transmit code which is consistent to valid data state. This result of low to high transitions is correspond to dual rail data which is flow through asynchronus CSLA block. After the next stage register receives a code word sequently to data processing in the asynchronous CSLA block it drives ackout to 1, and ackin 0. The current stage registers waits for ackin to become 0 and then reset the data. After data are unbounded but positive and finite amount of time taken for the resetting of asynchronous CSLA block and spacer data flow through the next stage register drives ackout 0. After this arriving of data the asynchronous circuit is ready to start the next data transaction.

4 phase asynchronous signalling protocol for expressed with acknowledge (ack) and request (req) of data signal wire describe handshaking process. Sender wants to transfer data to receiver then request has been rised. After that receiver detects request that time rises the acknowledge then sender detects rising acknowledge at that time falling the request signal and then receiver detects falling signal then acknowledge has been falling and finally back to the initial state. It can be observe that four transitions are required to complete a data transaction based on this signalling protocol. Asynchronous logic blocks are classified into three methods such as strongly indicating, weakly indicating and early output. Indication in an asynchronous CSLA block means acknowledge of primary inputs through the primary outputs while performing intermediate outputs. These indication mechanism may be performed global or local in asynchronous circuit stage. If the indication is local - asynchronous block within the asynchronous circuit stage all the primary inputs are indicate itself and global - if the asynchronous circuit stage indicates all the primary inputs along with the asynchronous logic block present in it. Local week indication is most preferable than the global weak indication for asynchronous circuits because it working as delay insensitive data encoding and following a 4 phase handshake protocol combined to produce power cycle time area.

Strong indication means all the primary output of asynchronous logic block will be produced only after all the primary inputs to the asynchronous logic block [12,13] are supplied. For example, C-element is said to be a strong-indication element. An inverter is also a strong- indication gate. If the input of the inverter is 0, then the output is 1. This means when an inverter output is 1 surely know that its input is 0. Similarly, the inverter can produce 1 as output only after receiving 0 as input. Only the C-element and the inverter are strongly indicating gates. Weak indication means some primary output of asynchronous logic block will be produced after receiving just a subset of the primary inputs. An early output [14,15] asynchronous logic block is compared to the strong indication and weak indication asynchronous logic blocks in that process produce all the primary outputs after receiving a subset of the primary inputs. The early output type asynchronous logic block is producing all the spacer primary outputs after receiving a subset of spacer primary inputs. For example, the logic gates like AND, NAND, OR, NOR, XOR, XNOR and the other complex gates are early output type gates. Suppose if one input to an AND gate is 0, then its output becomes 0 regardless of the other inputs. Similarly, if one input of an OR gate is 1, then its output would be 1 regardless of the other inputs. So here, the AND gate and the OR gate can produce an output early before the rest of their inputs have arrived.

III. ASYNCHRONOUS CARRY SELECT ADDER ARCHITECTURE

In this work, asynchronous carry select adder consist of 32-bit dual operand addition. Ordinary carry select adder architecture are compared to asynchronous carry select adder. The strong indication carry select adder is constructed by using the strongly indicating full adder and strongly indicating MUX (i.e., The sum and carry output of the full adder will be produced only after all the inputs to the full adder are supplied. Hence it is said to be strongly indicating). The weak indication carry select adder is constructed by using the weakly full adder and strongly indicating MUX (i.e., The carry output can be produced early the sum output would be produced only after all the input are supplied. Hence it is said to be weakly indicating). The early output carry select adder constructed by using the early output full adder and the strong indicating MUX (i.e., When spacer data is applied both the sum and carry outputs may be produced early before all the inputs have arrived. Hence it is said to be early output).

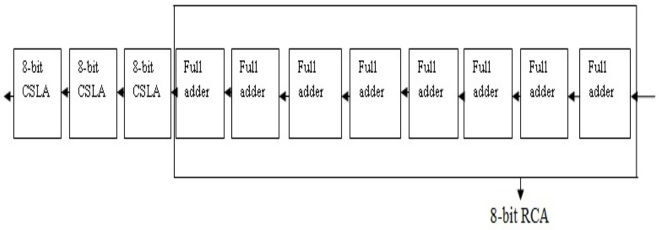

Fig 3. Asynchronous CSLA architecture consistent to the input partition 8-8-8-8.

To perform addition in the rest of the more significant byte positions, equal sized CSLA modules are used. In general, small size CSLA are used in the least significant positions and large size CSLA are used in the most significant positions in order to reduce the critical path delay.

The detail explanation of asynchronous CSLA architecture consist of eight asynchronous full adders are cascaded to produce a 8-bit asynchronous RCA [16-18]. Carry input of 8-RCA is zero. Carry inputs 0 and 1are parallely performed in 8-bit RCA. Then produce four set of output from the two 8-bit RCAs the correct set of sum outputs is selected based on the actual carry input supplied from the previous stage. Also the correct carry output is selected and forward actual carry input to the next stage.

Fig. 3 shows that the entire size of CSLA is 32 bit which is partitioned into 8-bits. Addition in the least significant byte position of the 32bit CSLA is performed using an 8-bit RCA.

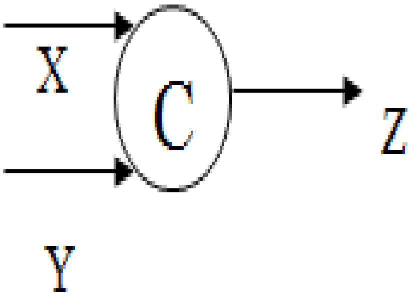

Fig. 4. Two input Muller C Element

Fig. 4 shows that the 2 input muller C- element is represented by the AND gate symbol with marking C on in it. Which is constructed by using the A0222 cell [19] by incorporating feedback and it was implemented the various asynchronous CSLAs because C-element has high quasi delay insensitive.muller C model model can be represented as

$$Z=XY+XZ+YZ \quad (1)$$

Where X and Y represent the dual rail augend and addend inputs and Z represent the dual rail output. Performance of Muller C element, output changes to zero, if all inputs are zero and output changes to one, if all inputs are one and often used in asynchronous circuits. Fig. 5 shows that the

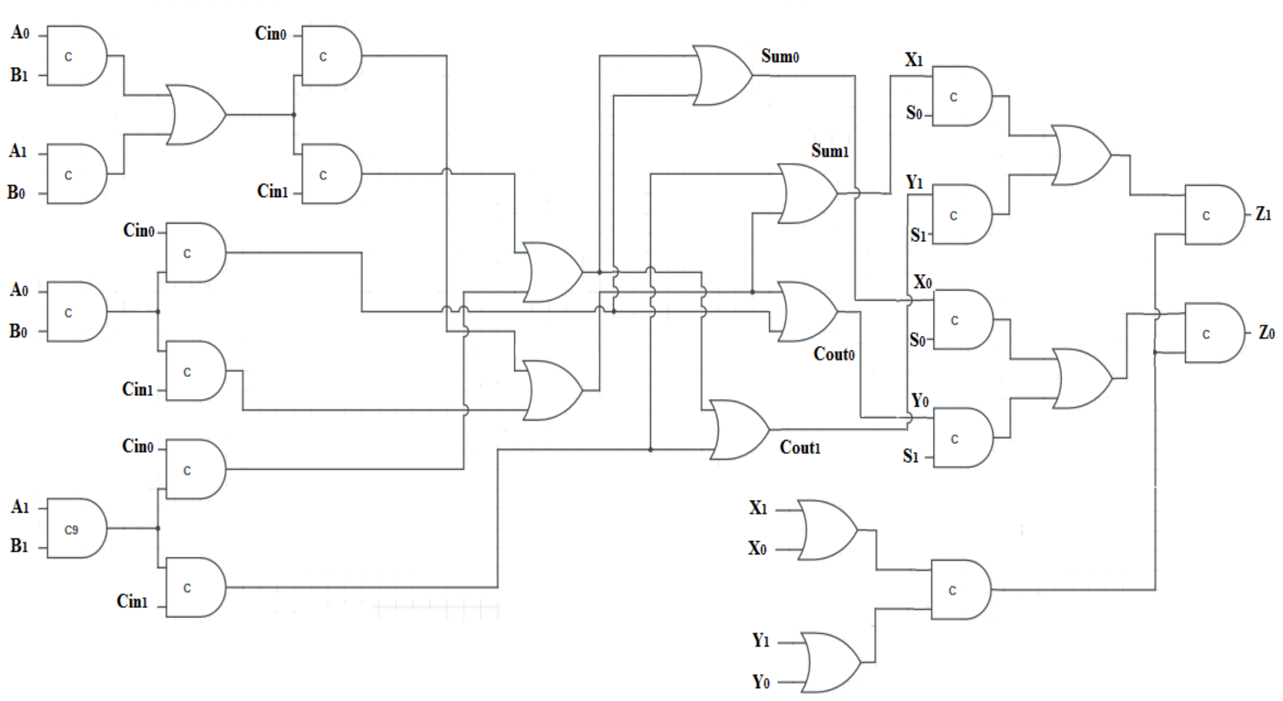

Fig.5. Two bit Strong indication asynchronous CSLA

2-bit strong indication of asynchronous CSLA. Where A0, A1 and B0, B1 represent the dual rail augend and addend inputs and Cin0, Cin1 represents the dual rail carry input. Sum0,sum1and Cout0, Cout1 represent the dual rail sum and carry outputs respectively.X0, X1and Y0, Y1 are the primary inputs S0, S1 is the dual rail selection input and Z0, Z1 is the dual rail primary output of MUX. The strong indicating MUX represent the sum and carry output of the full adder produced only after all the inputs to the full adders supplied. After that, the two outputs Sum0, Sum1 are given as inputs to the MUX and produce two output signals Z1, Z2 respectively.

IV.COMPARISION AND SYNTHESIS

The proposed design is compared with both 8, 16 and 32-bit asynchronous CSLA correspond to strong indication, weak indication and early output with classical using delay insensitive dual rail data and the 4- phase return to zero protocol for handshaking. The complete design along with all timing constraints, area utilization and optimization options are described using synthesis report. The design is synthesized using spartan-3E FPGA module.

A. Analysis of area utilization and delay encountered.

The area utilization summary of the 8-bit, 16-bit, 32-bit synchronous carry select adder with 8-bit, 16-bit, 32-bit asynchronous CSLA corresponding to strong indication, weak indication and Early output is briefly shown in Table 1, Table 2, Table 3 and Table 4. The area synthesis report shows that with the asynchronous CSLA has lesser area and also has less delay value as compared with synchronous CSLA.

Table 1. Area Utilization of carry select adder

Area Utilization	**Carry Select Adder**		
	32bit	**16bit**	**8bit**
Number of slices	38	19	10
Number of 4 input LUTs	72	37	19
Number of bonded IOBs	98	50	26
Delay (ns)	16.343	16.286	11.56

Table 1.shows that synchronous CSLA occupies large number of resource in hardware and also has large delay value.

Table 2. Area Utilization of Strong indication asynchronous CSLA.

Area Utilization	**Strong indication Asynchronous CSLA**		
	32bit	**16bit**	**8bit**
Number of slices	27	14	6
Number of 4 input LUTs	50	26	6
Number of bonded IOBs	99	51	19
Delay (ns)	9.784	9.566	6.521

Table 3. Area utilization of Weak indication asynchronous CSLA.

Area Utilization	**Weak indication Asynchronous CSLA**		
	32bit	**16bit**	**8bit**
Number of slices	18	9	4
Number of 4 input LUTs	28	15	6
Number of bonded IOBs	99	51	19
Delay (ns)	6.823	6.546	6.530

Table 4.Area utilization of early output asynchronous CSLA.

Area Utilization	**Early indication Asynchronous CSLA**		
	32bit	**16bit**	**8bit**
Number of slices	20	11	4
Number of 4 input LUTs	25	13	7
Number of bonded IOBs	99	51	19
Delay (ns)	6.81	6.542	6.521

The above Table 2, Table 3,Table 4 shown that early output asynchronous CSLA compared to the strongly indicating and weakly indicating inorder to know strongly indicating has large area utilization and delay than weakly indicating and early output CSLA.

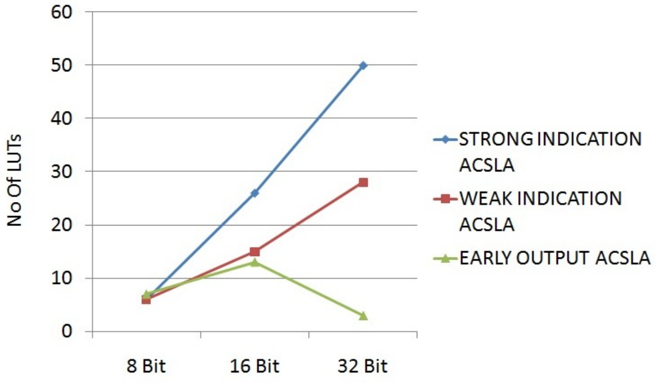

Fig. 6. LUTs Utilization

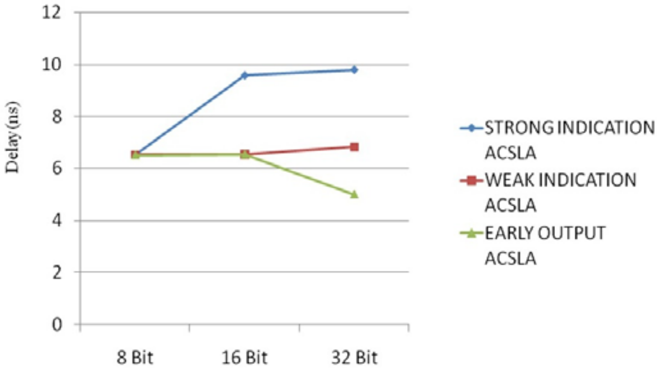

Fig. 7. Delay Occupancy.

The above Fig.8, Fig 9 shows that LUTs utilization and area occupied by strong indication, weak indication and Early output ACSLA. Hence, prove that strong indication has high delay, area than weak indication and early output ACSLA.

V.CONCLUSION

In this proposed method of 8, 16 and 32 bit ACSLA are implemented on a spartan-3E FPGA device. The asynchronous CSLA consistent to strong indication, Weak indication and early output types using delay insensitive dual rail data and 4-phase return to zero protocol for handshaking are syntesized. The results obtained interms of the area occupied and the delay encountered in the FPGA due to strong, weak and early indication of ACSLA are compared with the classical synchronous CSLA logic design. From the obatined results summarized in the Fig. 6 and Fig. 7 it is evident that the ACSLA logic design outperfomes the CSLA counterpart in terms of area occpied and the delay encountered in the design.

ACKNOWLEDGEMENT

The authors wish to thank the Management, Principal and Head of Electronics and Communication Engineering Department of Mepco Schlenk Engineering College, for their support in carrying out this research work

REFERENCES

[1] Lee Sup Kim and Youngjoon Kim, "64-bit Carry Select adder with reduced area," Electronics Letters,Vol.37,March 2001.

[2] Zbigniew Hajduk, "Simple method of asynchronous circuits implementation in commercial FPGAs," Integration , the VLSI journal,Vol.59,septemper 2017.

[3] D. Sokolov, J. Murphy, A. Bystrov, A. Yakovlev, “Design and analysis of dual-rail circuits for security applications,” IEEE Transactions on Computers, vol. 54, no. 4, pp. 449-460, April 2005.

[4] D.E. Muller, W.S. Bartky, “A theory of asynchronous circuits,” Proc. International Symposium on the Theory of Switching, Part I, pp. 204-243, Harvard University Press, 1959.

[5] Belle W.Y.Wei and Clark D.Thompson,"Area -Time Optimal Adder Design,"IEEE transactions on computers,Vol.31,pp.260-264,March 1982.

[6] Sparso,S.Furber,"Principles of Asynchronous Circuit Design:A Systems Serspective,"Kluwer Academic Publishers,Beston,USA 2001.

[7] T.Verhoeff,"Delay Insensitive Codes-An Overview," Distributed Computer,March 1998.

[8] B. Folco, V. Bregier, L. Fesquet, M. Renaudin, “Technology mapping for area optimized quasi delay insensitive circuits,” Proc. IFIP International Conference on VLSI-SoC, pp. 146-151, 2005.

[9] S. Goel, A. Kumar, M.A. Bayoumi, “Design of robust, energy-efficient full adders for deep submicrometer design using hybrid-CMOS logic style,” IEEE Transactions on VLSI Systems, vol. 14, no. 12, pp. 1309-1321, 2006.

[10] S.J. Piestrak, T. Nanya, “Towards totally self-checking delay-insensitive systems,” Proc. 25th International Symposium on Fault-Tolerant Computing, pp. 228-237, 1995

[11] J. Sparsø, S. Furber, Principles of asynchronous circuit design: a systemsperspective, Kluwer Academic Publishers, Boston, MA, USA, 2001.

[12] P.Balasubramanian , D.A. Edwards,"Efficient Realization of Strongly Indicating Function Blocks,"in:Proc.IEEE computer Society Annual Symposium on VLSI,2008,pp. 429-432.

[13] P.Balasubramanian , D.A. Edwards,"Efficient Realization of Strongly Indicating Function Blocks,"in:Proc.11th IEEE workshop on Design and Diagnostics of Electronic Circuits and Systems 2008,pp.116-121.

[14] C.F.Brej,J.D.Garside,"Early Output Logic using Antitokens,"in:Proc.12th International Workshop on Logic and Synthesis,2003,pp.302-309.

[15] P.Balasubramanian,"A robust Asynchronous early output full adder,"WSEAS Trans Circuit Syst,July 2011,pp.221-230.

[16] A.J. Martin, “Asynchronous datapaths and the design of an asynchronous adder,” Formal Methods in System Design, vol. 1, no. 1, pp. 117-137, 1992.

[17] P. Balasubramanian, N.E. Mastorakis, “A low power gate level full adder module,” Proc. 3rd International Conference on Circuits, Systems and Signals, Invited Paper, pp. 246-248, 2009.

[18] P. Balasubramanian, D. Dhivyaa, J.P. Jayakirthika, P. Kaviyarasi, K. Prasad, “Low power self-timed carry lookahead adders,” Proc. 56th IEEE International Midwest Symposium on Circuits and Systems, pp. 457-460, 2013.

[19] P. Balasubramanian, N.E. Mastorakis, QDI decomposed DIMS method featuringhomogeneous/heterogeneous data encoding, in: Proc. InternationalConference on Computers, Digital Communications and Computing, 2011,pp. 93–101.