International Journal of advancement in electronics and computer engineering (IJAECE)
Volume 4, Issue 8, Dec. 2015, pp.555-559, ISSN 2278 -1412

Design Area-Efficient and high speed 32bit CSLA

Malti Aryayan[1], Jaikaran Singh Chauhan[2]
[1]*M.Tech Scholar, SSSUT&MS Sehore, malti.aryayan.786@gmail.com, India;*
[2] *Asso. Professor, SSSUT&MS Sehore, jksingh81@gmail.com, India;*

Abstract *– Modern applications demand extremely less area budgets and enhanced speed in computer architectures for battery-operated devices like Laptop and others. In this thesis, the main focus is on the area and provides high speed to the processors. Less area and high speed circuits are becoming more desirable due to growing portable device markets and they are also becoming more applicable today in processors. The main focus in this work is to improve the speed of the 32-bit processor and in this case the carry select adder is the better choice. The second concern in the design of this carry select adder is the reduction of area that is achieved by implementing the internal structure of the ripple carry adder in the branching architecture. The approach used here is to implement these pipelines in a manner that only one pipeline will be activated through which the carry is propagating. Furthermore, RTL level optimizations have also been done to ensure less area during intermediate computations. In this thesis, the entire adder architecture has been implemented using Verilog and simulated using ISE tool suite. It provides a good improvement in speed as well as reduces the area requirement to a greater extent.*

Keywords*: CSLA,RCA,BCD,32 Bit CSLA*

I. Introduction

The design of high speed, low power and as well as minimum area adder architecture has been the main concern of many Very Large Scale Integration (VLSI) researchers and this resulted in a large number of adder architectures. These various available architectures provide the capacity to obtain the gain more accurate and deep instinctive understanding of adder and thus suggest various implementations. Here we are going to discuss about the requirement of adder in processors and will also discuss about different adder architectures. This Design uses a simple and efficient gate-level modification to meaningfully shrink the area and power of Carry Select Adder (CSLA). Based on this modification 16-, 32 Square-Root Carry Select Adder (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. The proposed Work has reduced area, and delay as compared with the regular SQRT CSLA. The CSLA is used in computational systems to shrink the badly behaved delay by autonomously producing multiple carries and then select a carry to generate the sum of carry propagation. In this thesis, the main focus is on the low power consumption and provides high speed to the processors. In processors the main concern is an adder which is required in ALU to perform arithmetic operations as well as to decode and fetch addresses in the memory and from the memory. A pipelining technique is using here to. The carry select adder belongs to the grouping of conditional sum adder as it works over conditional statements. Conditional sum adder means which works on some condition. In this sum and carry are calculated by assuming input carry as 1 and 0 coming before the input carry. When actual carry input arrives in system, the actual calculated values of sum and carry are selected using multiplexer. reduce the area and provide high speed of data processing

II. Theory and Methods

II.1. Ripple Carry Adder

Multiple full adder circuits can be cascaded in parallel to add an n-bit numbers. For an n-bit parallel adder, there should be n-number of full adder circuits. A simple 4-bit RCA is shown in fig.5.1. A ripple carry adder is a logic circuit in that the carry-out of each one full adder is the carry in of the succeeding next most important full adder. It is called a ripple carry adder as each carry bit gets rippled into the next stage. In a ripple carry adder the addition and carry out bits of any half adder step is not valid until the carry in of that step occur. Propagation delays within the logic circuitry are the reason at the back this. Propagation delay is time elapsed among the application of an input and amount of the corresponding output. Consider a NOT gate, When the input is "0″ the result will be "1″ and vice versa. The time in use for the NOT gate's output to become "0″ after the use of logic "1″ to the NOT gate's input is the propagation delay at this point. Similarly the carry propagation delay is the

time elapsed among the application of the carry in pointer and the occurrence of the carry out (Cout) signal. The outline of ripple carry adder is plain, which allows for quick design time; however, the ripple carry adder is quite slow, since each full adder have to wait for the carry bit to be designed from the earlier full adder. The gate delay can easily be designed by examination of the full adder circuit. Each full adder requires three levels of logic. In a 32-bit RCA, here are 32 full adders, so the critical path (worst case) delay is 31 * 2(for carry propagation) + 3(for sum) = 65 gate delays.

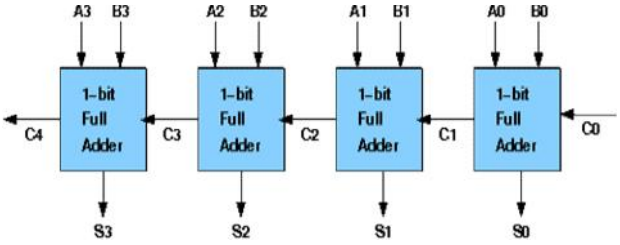

Fig.1. 4-bit Ripple Carry Adder [35]

II.2. Multiplexer

In electronics, a multiplexer (or mux) is a machine that selects one of some analog or digital input signals and ahead the select input into a single line. A multiplexer of 2^n input has *n* select lines that are used to choose which input line to send to the output. Multiplexers are primarily used to enhance the amount of data that can be send over the network in a certain quantity of time and bandwidth. A multiplexer is also called a data selector.

II.3. Basic Adder Unit

The most basic arithmetic operation is the adding up of binary digits i.e. bits. A combinational circuit that add two bits is call the adder. A full adder is that which adds three bit, the third formed due to previous addition operations. The other way of implementing a full adder is to utilize two half adders in its execution. The full adder is the basic structure blocks of all the adders.

II.4. Carry Select Adder

The carry-select adder can be simply define as a combination of two Ripple Carry Adders (RCA), one to generate the sum for carry input C_{in} =1 and the other to generate the sum for carry input C_{in}=0 and a multiplexer stage is connected to choose the correct sum output in which the selection line is taken as the actual carry propagation. Adding two n-bit data with a carry-select adder is done with two ripple carry adders in order to perform the calculation twice, we assume that first time carry will be zero and second time carry will be one. After that the two results, the correct sum and the correct carry are calculated, is then selected with the multiplexer once the correct carry is known.

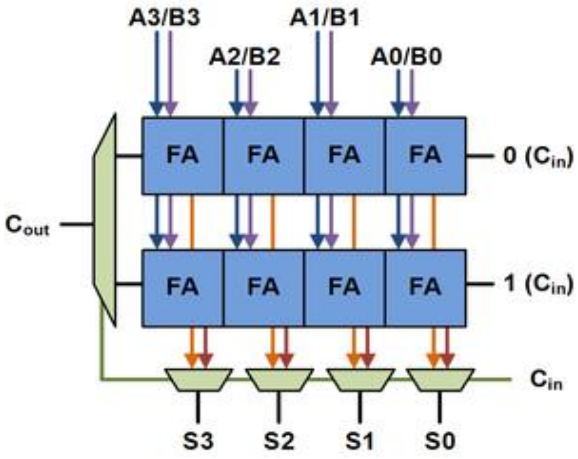

Fig.2: Carry Select Adder [36]

III. Proposed Methodology

The methodology has been implemented in this research work is utilizing the basic concept of the CSLA using the RCA. The simple design flow has been shown in fig.4.1. But the simple concept is implementing here in a different manner. Here the sum is generating linearly but the carry is propagating non-linearly.

III.1. Design Flow of CSLA

In proposed 16-bit, 32BIT carry select adder. Carry Select Adder (CSLA) is one of the fastest adders utilized in numerous data-processing processors to perform fast arithmetic functions. From the formation of the CSLA, it is observe that there is chance for reducing the field area and delay in the CSLA. This part of work uses a simple, easy and efficient gate-level modification to significantly reduce the area and power of the CSLA. The BCD is utilized alternatively of RCA. The basic design flow of the data is shown here. It is providing us the information that how this design has been working. So, the first block is representing the setup block which is signify the initial setup time require to initiate the process. If the block consists of a 4 full adder then that block will add the bits from k to k+3. Here design is 16 bit combine with 4 bit RCA in 4 times. This 16 bit architecture again adds with 16 bit same architecture and design 32 bit architecture. Reparation of same modal gate design reduced, reduced of gate count are is less. After its BCD is use that is reduce delay. Then the simple MUX stage is provided to select the correct sum. Here the selection operation is much faster than time to compute either of the two possible vectors. Implementation of each of these blocks

has been done through HDL level optimization to obtain better performance versus efficiency ratio.

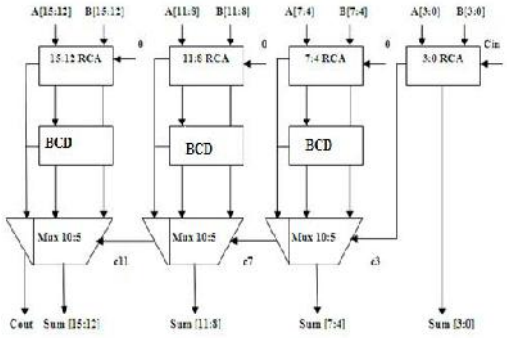

Fig3: Design Flow of CSLA

III.2. 4-bit Linear CSLA

Carry select Adder is a better choice especially in the case of Carry delay. As in a ripple-carry adder, every full adder cell has to wait for the incoming carry before an outgoing carry can be generated. This dependency can be eliminated by pre-calculating i.e. by taking both possible values of the carry input and evaluating the result for both possibilities in advance. Once the real value of the incoming carry is known, the correct result is easily selected with a simple multiplexer stage. The implementation of this idea is called the linear carry select adder and the diagram of the first four bits of the adder is shown in fig 4.3.

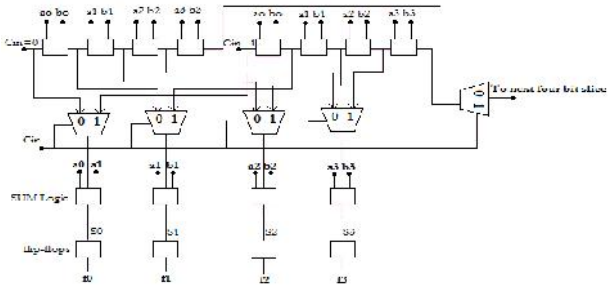

Fig4: Diagram of first 4-bit linear CSLA

In the above diagram all the inputs are given at a time to both the Ci =0 and the Ci =1 carry logic. The carry circuits generate the appropriate carryout's and depending upon the original carry input the appropriate values are selected from the multiplexer and fed to the sum circuit. This is the basic 4- bit carry select adder. Thus for implementing the higher order bit the carry out from the fifth stage multiplexer passes as the carry in for the next 4-bit, while the inputs are given at the same time. It is clear that the delay is reduced to a large extent by performing the carry calculations beforehand, but the disadvantage is that the hardware over head of the carry select adder is restricted to an additional carry path and a multiplexer.

III.3. Implementation of Non-linear Carry propagation

The actual medication which has been made in this proposed CSLA is to use a method which gives more optimum results is to apportion the adder non-linearly. Here, as the computation of 16-bits has been implementing. Thus, to minimize the number of stages require for the computation can reduced only and only if the variable size ripple carry adders are implement. Fig 4.4 shows 16-bit carry select adder design.

This concept can be understood as; if linear carry propagation style is used then it will require the 4-blocks of 4-bit RCA. But if the non-linear style of carry propagation has been used then the implementation of 16-bits can be performed in only 4-stages and in this way delay will be reduced and also the area. For example to design a 16 bit Carry-Select Adder one can use 6 stages of adders with sizes: 4, 4, 4, and 4 = 16 bits. Each stage computes a partial sum; Ripple adders can be used for stage adders.

IV. Result Analysis

This is our proposed result section that is show design architecture which is design with the help Xilinx

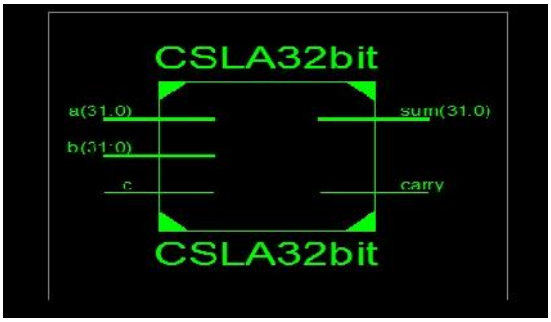

Fig5 : RTL Design 32 Bit CSLA

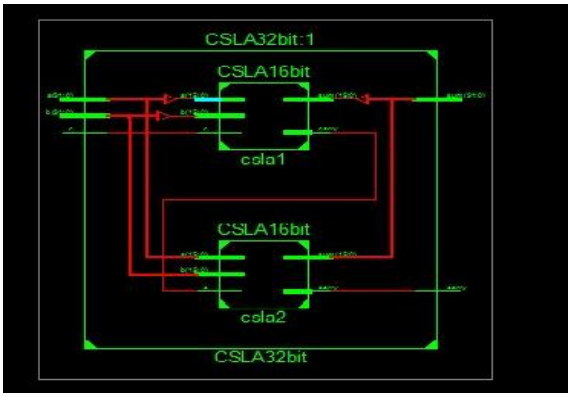

Fig6 : Internal RTL Design Two 16 Bit CSLA

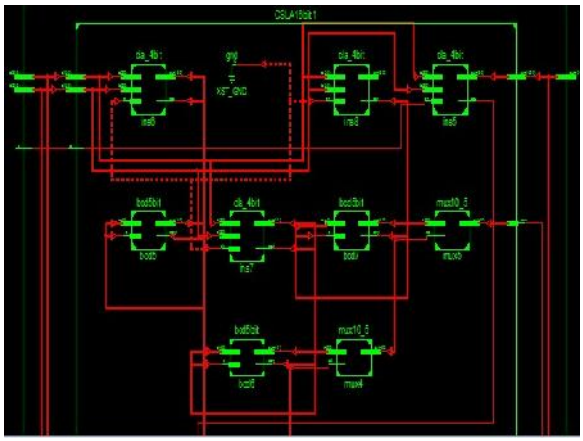

Fig7 : Internal RTL Design Two 4 Bit CSLA

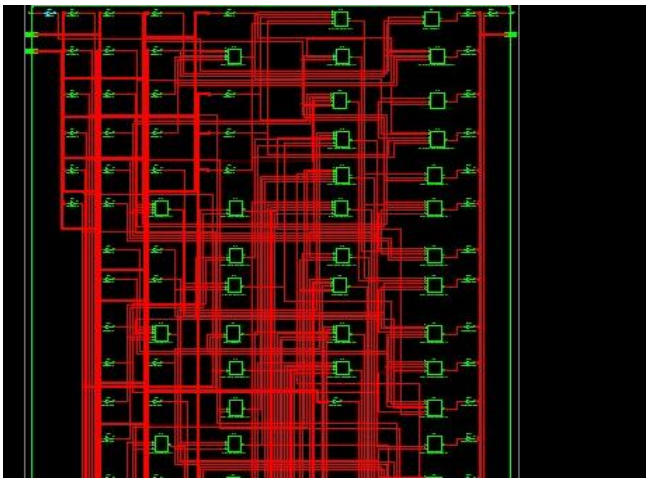

Fig8 : Internal Technology Design Two 24 Bit CSLA

Fig9: Simulation Result

V. Conclusion

In this project, the entire adder architecture has been implemented using VHDL. The structure is then further evaluated in FPGA using Xilinx ISE Design Suite. This thesis has deal with fundamental concepts of addition and optimization. Here we are presented a number of interesting results.

Here the conclusion is obtained that this proposed structure of CSLA is better than all the architectures that have been implemented previous, since every constraints are optimized by using this proposed CSLA. Thus, the obtain results are much better in comparison to previous work in all the respects.

References

[1] R.Priya and J.Senthil Kumar, "Enhanced Area Efficient Architecture for 128 bit Modified CSLA", 20I3 International Conference on Circuits, Power and Computing Technologies [ICCPCT-20I3]

[2] Senthil kumar.A and Kousalya devi.A, "VLSI Implementation of an Efficient Carry Select Adder Architecture" International Journal Of Advance Research In Science And Engineering IJARSE, Vol. No.2, Issue No.4, April, 2013.

[3] Pallavi Saxena, Urvashi Purohit, Priyanka Joshi, "Analysis of Low Power, Area- Efficient and High Speed Fast Adder"International Journal of Advanced Research in Computer and Communication EngineeringVol. 2, Issue 9, September 2013.

[4] J R Padamshri, Christina P A, Kavitha. P and T Mullai, "Modified Area Efficient Carry Select Adder(MA-CSLA)"International Journal of Advanced Electrical and Electronics Engineering (IJAEEE) ISSN (Print) : 2278-8948, Volume-2, Issue-2, 2013.

[5] L.Mugilvannan and S.Ramasamy, "Low-Power And Area-Efficient Carry Select Adder Using Modified Bec-1 Converter" International Journal of Computer Applications in Engineering Sciences Special Issue on National Conference on Information and Communication (NCIC'13) Volume III, Special Issue, August 2013 ISSN: 2231-4946.

[6] Ramkurnar, B. and Harish M Kittur, "Low Power and Area Efficient Carry Select Adder ", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp.I-5, 2012.

[7] Santanu Maity, Bishnu Prasad De, Aditya Kr. Singh, "Design and Implementation of Low-Power High-Performance Carry Select Adder" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-4, April 2012.

[8] Shivani Parmar and Kirat pal Singh,"Design of high speed hybrid carry select adder",IEEE ,2012.

[9] T. RatnaMala, R. Vinay Kumar, T. Chandra Kala, "Design and Verification of Area Efficient High-Speed Carry Select Adder" T. RatnaMala , et al, International Journal of Research in Computer and Communication technology, IJRCCT, ISSN 2278-5841, Vol 1, Issue 6, November 2012.

[10] B. Ramkumar, H.M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," *Eur. J. Sci. Res.*, vol. 42, no. 1, pp. 53–58, 2010.

[11] Padma Devi, Ashima Girdher and Balwinder Singh, "Improved Carry Select Adder with Reduced Area and Low Power Consumption"International Journal of Computer Applications (0975 – 8887) Volume 3 – No.4, June 2010.

[12] Z. Abid, H. El-Razouk and D.A. El-Dib, "Low power multipliers based on new hybrid full adders", Microelectronics Journal, Volume 39, Issue 12, Pages 1509-1515, 2008.

[13] Hasan Krad and Aws Yousif Al-Taie, "Performance Analysis of a 32-Bit Multiplier with a Carry-Look-Ahead Adder and a 32-bit Multiplier with a Ripple Adder using VHDL", Journal of Computer Science 4 (4): 305-308, 2008.

[14] Cadence, "Encounter user guide," Version 6.2.4, March 2008.

[15] Yajuan He, Chip-Hong Chang and Jiangmin Gu, "An Area Efficient 64-bit Square Root Carry-select Adder for Low Power Applications", Proc. IEEE Int. Symp.Circuits Syst., 2005, vol. 4, pp. 4082–4085

[16] Behnam Amelifard, Farzan Fallah, and Massoud Pedram, "Closing the gap between carry select adder and ripple carry adder: a new class of low-power high-performance adders", in Proc. of IEEE International Symposium on Quality Electronic Design (ISQED), 2005.