Efficient Performance Analysis with Literature Survey on Unsigned Multiplier Using CLAA and CSLA

Gaurav Patidar[1], Prof. Sudha Nair[2]
[1]M-Tech Research Scholar, [2]Research Guide, RKDF Institute of Science & Technology, Bhopal

Abstract-A multiplier is one of the key hardware blocks in most digital and high performance systems such as FIR filters, digital signal processors and microprocessors. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier. Designingof such multipliersuitable for various high speed, low power, and compact VLSI implementations.Though area and speed are two conflicting constraints.Therefore improving speed results always in larger areas. Thus here try to find out the best trade off solution among the both of them. In this review we have presented a efficient analysis through a literature survey in order to improve the system performance of unsigned multiplier using CLAA and CSLA.

Keywords: Unsigned Multiplier, Carry Select Adders (CSLA), Carry Look Ahead Adder (CLAA).

I. INTRODUCTION

Multiplication is a mathematical operation that at its simplest is an abbreviated process ofadding an integer a specified number of times. Multiplication is the fundamentalarithmetic operation important in several processors and digital signal processingsystems. Multiplication of two k bit number needed multi operand addition process thatcan be realized in k cycles of shifting and addition with hardware, firmware or software.Multiplication based operations such as multiply and accumulate and innerproduct are among some of the frequently used intensive arithmetic functions currentlyimplemented in many digital signal processing applications such as convolution,fast fourier transform,filtering and in microprocessors in its arithmetic and logicunit.

Under multiplication method two fundamental steps comes, partial product and addition. Therefore in this paper first to design different adders and compare their speed and complexity of circuit i.e. the area occupied. And then to designed Wallace tree multiplier then followed by Booth's Wallace multiplier and have compared the speed and Power consumption in them. While comparing the adders to found out that Ripple Carry Adder had a smaller area while having lesser speed, in contrast to which Carry Select Adders are high speed but posses a larger area. And a Carry Look Ahead Adder is in between the spectrum having a proper tradeoff between time and area complexities. After designing and comparing the adders we turned to multipliers. Initially went for Parallel Multiplier and then Wallace Tree Multiplier. In the mean time learned that the delay amount was considerably reduced when Carry Save Adders were used in Wallace Tree applications. Then turned to Booths Multiplier and designed Radix-4 modified booth multiplier and analyzed the performance of all the multipliers. After that crooked to different methods of power optimization, of which it could only complete a few like it went for designing different recoding schemes and their corresponding partial product generator scheme. After that it designed these recorders and PP generators and found out the time delays and area covered and power consumed by each scheme. It took into consideration that since all the PP generators take a huge amount of area that require to go for simplest of the designs for them and also side by side it need to ensure that it don't have much switching actions in the circuit.

There are two types of multiplier as revealed in figure 1.

i. Serial multiplication algorithms
ii. Parallel multiplication algorithms

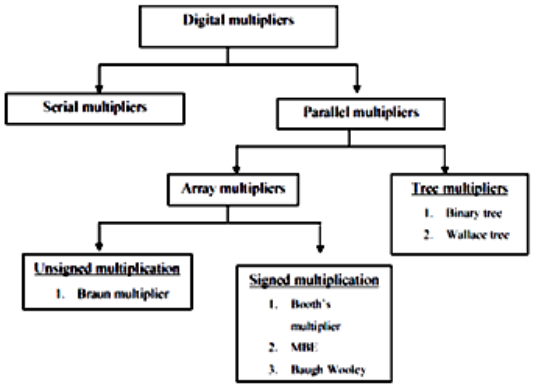

Fig. 1: Classification of multipliers [11]

II. SYSTEM MODEL

Let the product register size be 64 bits. Let the multiplicand registers size be 32 bits. Store the multiplier in the least significant half of the product register. Clear the most significant half of the product register. Repeat the following steps for 32 times:

If the least significant bit of the product register is "1" then add the multiplicand to the most significant half of the product register. 2. Shift the content of the product register one bit to the right (ignore the shifted-out bit.) 3. Shift-in the carry bit into the most significant bit of the product register. Figure 2.Shows a block diagram for such a multiplier [2].

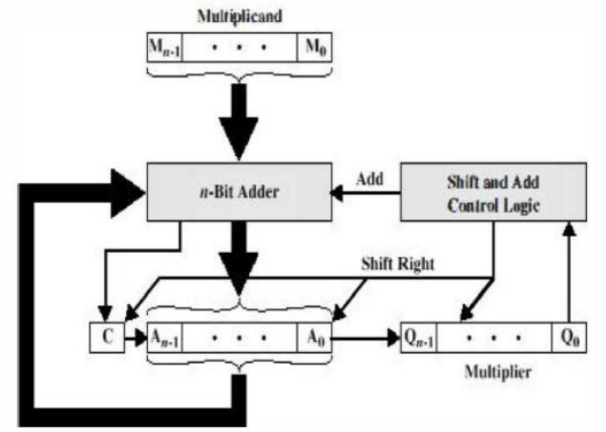

Fig.2. Multiplier of two n-bit values

Carry Select Adders (CSLA)

An 8-bit carry-select adder, built as a cascade from a 1-bit full-adder, a 3-bit carry-selectblock, and a 4-bit carry-select adder. Click the input switches or type the 'a', 'b', 'c' bind keys to control the first-stage adder. The problem of the ripple-carry adder is that eachadder has to wait for the arrival of its carry-input signal before the actual addition canstart. The basic idea of the carry-select adder is to use blocks of two ripple-carry adders,one of which is fed with a constant 0 carry-in while the other is fed with a constantcarry-in. Therefore, both blocks can calculate in parallel. When the actual carry-in signalfor the block arrives, multiplexers are used to select the correct one of both precalculatedpartial sums. Also, the resulting carry-out is selected and propagated to the next carryselectblock. In total, the carry propagation time through an n-bit adder block is reducedfrom O(n) to the number of stages times the delay of the multiplexers. Naturally, using nblocks of 1-bit carry-select adders would incur a complexity of n multiplexers, againresulting in O(n) delay. Therefore, a partition with (slowly) increasing block-size ischosen. In the example, the first (least-significant) block consists of a simple full adder,followed by a 3-bit carry-select block, and finally a 4-bit carry-select block. A commonchoice for a 16-bit carry-select adder is to use a 6-4-3-2-1 bit partitioning. While the delayof the standard ripple-carry adder with n-bits is O(n), the delay through the carry-selectadder behaves as O(sqrt(n)) at a hardware cost of O(3*n).Block diagram of carry selectadder is shown in figure 3.

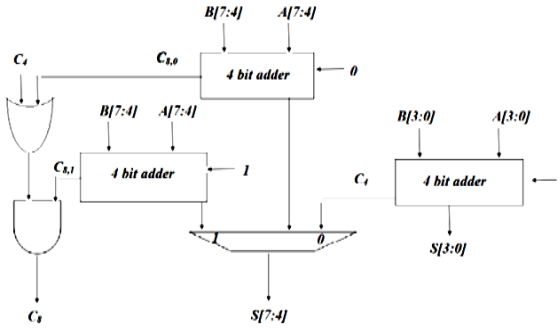

Fig. 3 Carry Select Adder

III. LITERATURE REVIEW

Vijayalakshmi, V., Seshadri, R. and Ramakrishnan, S[1] deal with the comparison of the VLSI design of the carry look-ahead adder based 32-bit unsigned integer multiplier and the VLSI design of the carry select adder based 32-bit unsigned integer multiplier. Both the VLSI design of multiplier multiplies two 32-bit unsigned integer values and gives a product term of 64-bit values. The CLAA based multiplier uses the delay time of 99ns for performing multiplication operation where as in CSLA based multiplier also uses nearly the same delay time for multiplication operation. But the area needed for CLAA multiplier is reduced to 31% by the CSLA based multiplier to complete the multiplication operation.

Shiann-Rong Kuang et al. [2] proposed a simple approach to generate a regular partialproduct array with fewer partial product rows, thereby reducing the area, delay, and power of MBE multipliers. The proposed MBE multiplier combines the advantages of thefollowing two approaches. First one is to add the least significant bit pi0 with negi inadvance and obtained a new least significant bit τi0. And second is to reduce the partialproduct rows from n/2+1 to n/2 by incorporating the last neg bit into the sign extensionbits of the first partial product row .The proposed MBE multiplier combines theadvantages of both of these two approaches to produce a very regular partial productarray, as shown in Figure 4

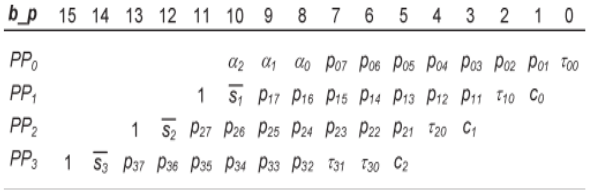

Fig.4. Regular partial product array for 8 × 8 multiplication

This regular array is generated by only slightly modifying the original partial product generation circuits and introducing almost no area and delay overhead.

HaiminChen,et al.[3] proposed the disposal of negative PP based onRadix-4 Booth algorithm. Through recombining PP, it advances a skilled methodavoiding the additive arithmetic of "plus 1" when computing the complement for negativePP, without increasing new PP. The experiment result shows it could obviously improvethe performance of multiplier. This method is of great significance for shortening the keypath of multiplier and then improving its execution speed, which could be applied in allmultipliers based on Radix-4 Booth encoding. The design proposed herer effectivelyavoid the additive arithmetic "plus 1", and without increasing new PP. It not only doesn'tincrease chip resources, but also shorten key path, which could obviously improve theperformance of multiplier. This design could be applied in all the multipliers based onRadix-4 Booth encoding, with great significance of application value.

Wen-Chang Yeh et al. [4] proposed a design methodology for high-speed Boothencoded parallel multiplier. For partial product generation they proposed a new modifiedBooth encoding (MBE) scheme.The conventional MBE partial product array has two drawbacks: 1) an additional partialproduct term at the (n-2) th bit position; 2) poor performance at the LSB-part comparedwith the non-Booth design when using the TDM algorithm. To remedy the two drawbacks, the LSB part of the partial product array is modified. Referring to equation2.1 the Row_LSB (gray circle) and the Neg_cin terms are combined and furthersimplified using Boolean minimization.

O.Saloman et al. [6] researched two algorithms for both a simplified carry save and carryripple addition of 2's complement numbers. The algorithms form the partial products sothat they exclusively have positive coefficients which eliminate the need for the commonsign bit extension. This results in a reduction of circuit area by up to six full adders perrow of adders when partial products are added in an n/2 or Wallace tree. Furthermore, thecapacitive load of the intermediate sum and carry sign bit signals decreases by up to afactor of seven which leads to an appropriate reduction of delay. Although the algorithmsare derived for multipliers they can always be applied to appropriate adder circuits.

Aamir A. Farooqui et al. [8] presented the data-path and VLSI implementation of a32x32 bit signed unsigned multiply accumulate (MAC) unit. In this design they had solved the problem of dealing with signed and unsigned numbers simultaneously, using modified Booth algorithm. This MAC unit can perform 32x32, 32x16, and two 16x16multiplications, on signed unsigned operands with a throughput of 2, 1, and 1 cycle,respectively. In this, MAC unit inputs are applied using 32-bit registers A (multiplier)and B (multiplicand), while the result is stored in 64-bit register C. To balance thepipeline stages of the MAC, the 16x16 multiplier result is produced in carry save formand finally added in the second stage of the MAC. This reduces the clock cycle andpower consumption of the MAC (fewer glitches due to a carry propagate adder) unit.They used special circuitry to accommodate signed unsigned operands and to deal withsign extension. In Booth multiplier the number of Booth encoders and selectorsrequired for m-bits is m/2 and it requires one extra bit to handle unsigned operands.

Therefore in Booth multipliers (for number of bits in power of 2), one extraencoder and selector is required, to deal with unsigned numbers.In this paper they solved this problem in a very simple manner, with minimum hardwareand delay. In this design signed and unsigned multiplication is controlled by two signals.When the multiplier is unsigned and the most significant bit of the multiplier isone then multiplicand is added with the 8-partial products (means multiplication byl), else a zero is added with the partial products. This can be accomplished by usingand gates. To deal with unsigned multiplicand, the multiplicand is sign extended to 17bits. They used Modified Booth algorithm coupled with (three dimension reductionmethod) TDM and sign correction circuitry results in a multiplier, with a delay (partialproduct addition) equivalent to 6 XOR gates.

Fayez Elguibaly [9] worked on a dependence graph (DG) to visualize and describe amerged multiply-accumulate (MAC) hardware that is based on the modified Boothalgorithm (MBA). He used carry-save technique in the Booth encoder, the Boothmultiplier, and the accumulator sections to ensure the fastest possible implementation. Heapply DG MAC data word size and allows designing multiplier structures that are regularand have minimal delay, sign-bit extensions, and data path width. He proposed a fastpipelined implementation by using dependence graph, in which he used an accurate delaymodel for deep submicron CMOS technology. The delay model describes multi-level gatedelays, taking into

account input ramp and output loading. Based on the delay model,pipelined parallel MAC design proposed by him is three times faster than other parallelMAC schemes that are based on the MBA. The speedup resulted from merging theaccumulate and the multiply operations and the wide use of carry-save techniques.

OsmanHasan et al. [10] analyzed a formal synthesis methodology that can beautomated and thus it not only ensures formally verified synthesis results but also is veryeasy to use for end users who do not have any background in formal semantics andreasoning. Their synthesis methodology achieves correctness by construction and thuseliminates the post synthesis verification requirements, which in turn reduces design time.They had demonstrated the practical effectiveness of their methodology by successfullyconstructing an automated tool that is capable of correctly synthesizing WT multipliers ofarbitrary length operands. The proposed formal synthesis methodology is quite generaland can be applied to correctly synthesize any digital circuit. This methodology helps toenhance the library of formally verified correctness-preserving synthesis transformationsand thus formally synthesize a bigger set of combinational digital circuits.

S.SriSakthi et al. [11] presented architecture for efficient reconfiguration is the oneLevel Recursive Architecture. He implemented an n-bit multiplier using four n/2 bitmultipliers, where n is the number of bits. These four n/2 bit multipliers executes inparallel and their results are added up. Power consumption of the multipliers is reducedwith the introduction of power efficient scheme Dynamic Operand Interchange to thereconfigurable booth architecture. Dynamic operand interchange technique interchanges operands dynamically during the execution. The operand with the smallest dynamic rangeis chosen as the multiplier operand. This requires a Dynamic Range Detector (DRD)circuit to detect dynamic range of the operand and a switcher to exchange the operanddynamically. When the operand with smaller dynamic range is encoded with modifiedbooth, there is increased probability of partial products becoming zero. This reduces theswitching activities and thereby reduces power consumption.

S.Saravanan et al. [12] proposed a design approach of a low power Hybrid EncodedBooth Multiplier (HEBM) with Reduced Switching Activity Technique (RSAT). ThisRSAT approach has been applied on the hybrid encoder of the multiplier to reduce thepower consumption. The hybrid encoder in the low power multiplier uses both the Boothand proposed technique. If the number of 1's less than or equal to three, the proposedencoding technique used otherwise go for Booth technique.

IV. PROPOSED SYSTEM

The problem in the existing system is to get only the unsigned bit values, the signed bit values are not obtained. By using the full adder AND gate to produce only the unsigned bit values. The unsigned bit has only positive value. The proposed methodis to use 64-bit unsigned integer multiplier and the VLSI design of the carry select adder based 64-bit unsigned integer multiplier. Both the VLSI design of multiplier multiplies two 32-bit unsigned integer values and gives a product term of 64-bit values. The CLAA based multiplier uses by using the CLAA adders the area will be reduced and it produces the carries faster due to parallel generation of the carry bits by using additional circuitry. The CSLA have small area requirements and shortened computation times.

V. CONCLUSION

Various unsigned multiplier with CSLA and CLAA have beenanalyzed with the help of literature review. The adders structures are studiedwith area-delay, power-delay products are analyzed. VHDL, CSLA multiplier uses fewer resources when compared to all other adders but it has more delay time.While comparing the adders we found out that Ripple Carry Adder had a smaller area while having lesser speed, in contrast to which Carry Select Adders are high speed but posses a larger area. And a Carry Look Ahead Adder is in between the spectrum having a proper tradeoff between time and area complexities.

VI. FUTURE WORK

This 64 bit multiplier can be further extended to 128 bit multiplier and 128 bit multiplier using the proposed method for multiplication operation can be done as future work in order to enhance the system performance.

VII. ACKNOWLEDGEMENT

I would like to express thank the almighty for providing me the strength to work on this subject and coming up with this literature survey paper. We are thankful to my family for supporting us and praying for me. I would like to express my gratitude towards the professors of RKDFIST of Science and Technology for their precious guidance.

REFERENCES

[1] Vijayalakshmi, V.; Seshadri, R.; Ramakrishnan, S., "Design and implementation of 32 bit unsigned multiplier using CLAA and CSLA," Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), 2013 International Conference on , vol., no., pp.1,5, 7-9 Jan. 2013.

[2] S.R. Kuang, J.P. Wang, and C.Y. Guo, “Modified booth multipliers with aregular partial product array,”in IEEE Transaction on circuits and systems-II,vol. 56, no. 5, May 2009, pp. 404-408.

[3] Z. Li, H. Chen, X. Yang “Research on disposal of negative partial products forbooth algorithm,” in Proc. IEEE Conference on information theory and information security (ICITIS), Dec 2010, pp 1115 1117.

[4] W.C. Yeh and C.W. Jen, “High-speed booth encoded parallel multiplierdesign,” in IEEE Transaction on computer society, vol. 49, no. 7, Jul. 2000,pp. 692–701.

[5] J. Y. Kang and J. L. Gaudiot “A simple high-speed multiplier design ,”IEEETransaction on computer society, vol. 55, no. 10, Oct. 2006 ,pp. 1253–1258.

[6] O. Salomon, J.M. Green, and H. Klar, “General algorithms for a simplifiedaddition of 2’s complement numbers,” in IEEE Solid-state circuits, vol. 30,no. 7, Jul. 1995,pp. 839–844.

[7] E.de Angel and E. E. Swartzlander, “Low power parallel multipliers,” IEEE workshop on VLSI signal process. IX, Oct 1996, pp. 199–208.

[8] A. A. Farooqui, V. G. Oklobdzija“General data-path organization of a MACunit for VLSI implementation of DSP processors”. IEEE interactive symposium on ISCAS, vol.2, June 1998.

[9] F. Elguibaly, “A Fast parallel multiplier-accumulator using the modified booth’s algorithm”, IEEE Transactions on analog and digital signalprocessing, vol. 47 Sept. 2000, pp 902-908

[10] O. Hasan, S. Kort “Automated formal synthesis of wallace tree multipliers,”IEEE Transactions on MWSCAS, Aug 2007 pp 293-296.

[11] S. SriSakthi, N. Kayalvizhi, “Power aware and high speed reconfigurablemodified booth’s multiplier”, IEEE International conference on RAICS, Feb. 2011, pp 352-356.

[12] S. Saravanan, M. Madheswaran “Design of hybrid encoded booth’s multiplierwith reduced switching activity technique and low power 0.13μm adder for DSP block in wireless sensor node”, IEEE International conference on wirelesscommunication and sensor computing, Jan. 2010, pp 1-6

[13] F. Lamberti, N. Andrikos, E.Antelo, and P.Montuschi, ”Reducing the computation time in (short bit-width) two’s complement multipliers”, IEEE Transactions on computers, vol. 60, Feb. 2011.