Area Efficient Booth Recoder for Multiply and Accumulation Operation

Munjaji Bhise[1], Suresh Garud[2], Nikhil Hole[3], Prof. A. S. Deshpande[4]
[1, 2, 3, 4] Department of E&TC
[1, 2, 3, 4] ICOER, Wagholi ,Pune

Abstract- *Complex arithmetic operations are most widely used in Digital Signal Processing (DSP) applications. In this project we focus on optimizing design of fused Add-Multiply (FAM) operator for increasing performance. In this project we investigate techniques to implement the direct recoding of the sum of two numbers in its Modified Booth (MB) form.*

We introduce a structured and efficient recoding technique and use three different schemes by incorporating them in Fused add-multiply designs. As we comparing them with the FAM designs which use existing recoding schemes. The proposed technique yields considerable reductions in terms of critical delay, hardware complexity and power consumption on FAM unit.

This proposed method is purely based on modified recoding techniques for DSP applications. This implements a newly designed recoding technique for modified booth recoding. This technique is use for multiplied in its sum modified booth form. The proposed S-MB algorithm is structured, simple and easy to use for signed and unsigned numbers.

Thus FAM operator is optimized to increase the performance of complex DSP operation. In this we use three different techniques of recoding schemes S-MB1, S-MB2,S-MB3. The S-MB technique is implemented by Radix-4 Recoder.

Keywords- FAM Unit, SMB, Radix-4 recoder.

I. INTRODUCTION

Modern consumer electronics make extensive use of Digital Signal Processing (DSP) providing custom accelerators for the domains of multimedia, communications etc. Typical DSP applications carry out a large number of arithmetic operations as their implementation is based on computationally intensive kernels, such as Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Finite Impulse Response(FIR) filters and signals convolution. As expected the performance1of DSP systems is inherently affected by decisions on their design regarding the allocation and the architecture of arithmetic units. Recent research activities in the field of arithmetic optimization have shown that the design of arithmetic component combining operations which share data, can lead to significant performance improvements .Based on the observation that an addition can often be subsequent to a multiplication symmetric FIR filters),the Multiply-Accumulator (MAC) and Multiply-Add (MAD) units were introduced leading to more efficient implementations of DS algorithm compared to the conventional ones which use only primitive resources. Several architectures have been proposed to optimize the performance of the MAC operation in terms of area occupation, critical path delay or power consumption.

As noted in MAC components increase the flexibility of DSP data path synthesis as a large set of arithmetic operations can be efficiently mapped onto them. Except the MAC/MAD operations many DSP applications are based on Add-Multiply (AM) operations (e.g. FFT algorithm). The straight forward design of the AM unit by first allocating an adder and then driving its output to the input of a multiplier increases significantly both area and critical path delay of the circuit. Targeting an optimized design of AM operators, fusion techniques are employed based on the direct recoding of the sum of two numbers(equivalently a number in carry-save representation) in its Modified Booth (MB) form. Thus, the carry-propagate (or carry-look-ahead) adder of the conventional AM design is eliminated resulting in considerable gains of performance. Lyu-and Matula presented a signed-bit MB recoder which transforms redundant binary inputs to their MB recoding form. A special expansion of the preprocessing step of the recoder is needed in order to handle operands in carry-save representation.

In the propose system, two stage recoder which convert number in carry-save form to its MB representation. The first stage transforms the carry save form of the input number into signed digit form which is then recoded in the second stage so that it matches the form that the MB digits request. Recently the technique of has been used for the design of high performance flexible coprocessor architectures targeting the computationally intensive DSP applications. Zimermann and Tranpresent an optimized design of which results in improvements in both area and critical path. In the propose system recoding of a redundant input from its carry save form to the corresponding borrow save form keeping the critical path of the multiplication operation fixed.

Although the direct recoding of the sum of two numbers in its MB form leads to a more efficient implementation of the fused Add-Multiply (FAM) unit compared to the conventional one, existing recoding schemes are based on complex manipulation sine bit-level which are implemented by dedicated circuit sin gate level. This work focuses on the efficient design of FAM operators targeting the optimization of the recoding scheme for direct shaping of the MB form of the sum of two numbers (Sum to MB of S-MB). We propose a new recoding technique which decreases the critical path delay and reduces area and power consumption.

The proposed S-MB algorithm is structured, simple and can be easily modified in order to be applied either in signed (in 2’s complement representation) or unsigned numbers, which comprise of odd or even number of bits. We explore three alternative schemes of the proposed S-MB approach using conventional and signed-bit Full Adders (FAs) and Half Adders (HAs) as building blocks. We evaluated the performance of the proposed S-MB technique by comparing its three different schemes with the state of the art recoding techniques.

II. LITTERATURE SURVEY

Recent research activities in the field of arithmetic optimization [1], [2] have shown that the design of arithmetic components combining operations which share data, can lead to significant performance improvements. Based on the observation that an addition can often be subsequent to a multiplication (e.g.in symmetric FIR filters), the Multiply-Accumulator (MAC) and Multiply-Add (MAD) units were introduced [3]leading to more efficient implementations of DSP algorithms compared to the conventional ones, which use only primitive resources [4]. Several architectures have been proposed to optimize the performance of the MAC operation in terms of area occupation, critical path delay or power consumption [5],[7].

As noted in [8], MAC components increase the flexibility of DSP data path synthesis as a large set of arithmetic operations can be efficiently mapped onto them. Except the MAC/MAD operations, many DSP applications are based on Add-Multiply (AM) operations (e.g., FFT algorithm [9]). The straight forward design of the AM unit by first allocating an adder and then driving its output to the input of a multiplier increases significantly both area and critical path delay of the circuit. Targeting an optimized design of AM operators, fusion techniques [10],[13] are employed based on the direct recoding of the sum of two numbers (equivalently a number in carry save representation[14]) in its Modified Booth (MB) form [15].

Thus the carry propagate (or carry look ahead) adder [16] of the conventional AM design is eliminated resulting in considerable gains of performance. Lyu and Matula [10] presented a signed-bit MB Recoder which transforms redundant binary inputs to their MB recoding form. A special expansion of the pre-processing step of the Recoder is needed in order to handle operands in carry-save representation. In [12], the author proposes a two-stage recoder which converts a number in carry save form to its MB representation.

Array multiplier

As we say Array multiplier is very efficient layout of a combinational multiplier where multiplication of two binary number can be yielded with only one micro operation by employing a combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array. By having flexible structure, this multiplier is depending on the standard add and shift operations very easily. Moreover each partial product is generated by considering multiplicand and one bit of multiplier each time. Similarly impending addition is carried out by efficient high speed array save algorithm and also the final product is yielded employing any of fast adders consequently, number of partial products depends upon the number of multiplier bits used.

Array Multiplier provides more power consumption as well as optimum number of requirement of components, on the other hand provides much delay. As the area is increased requirement of gates is also more so due to this constraint array multiplier is less economical and hardware complexity is high.

Wallace Tree Multiplier

Wallace tree performs faster multiplication of two numbers that employs a three step method in order to complete this task where the bit products are can be formed and its matrix is reduced to a two row matrix in which its sum of the row equals the sum of bit products then the two resulting rows are summed up with efficient fast adder to obtain a final product. A fast adder is at the end produces the final result.

In Wallace tree circuit layout is complex even though speed of operation of it is high due to its irregular structure. Generally it is not preferable in low power applications because of excess wiring usage that result in extra circuitry and increase in power consumption in the multiplier.

Braun Multiplier

Braun multiplier is more efficient due to its regular structure. Due to its simple nature of parallel multiplier that is commonly termed as carry save array multiplier. So it is multiplier that restricted to perform multiplication of two unsigned numbers. Generally speaking about its structure that consists of array of AND gates and adders arranged in iterative structure that does not require any logic registers. Therefore it is termed as the non-additive multiplier since it does not add an additional operand to result of multiplication.
The major shortcomings of the Braun's multiplier are that the number of components required increases quadratically with that of number of bits which will make the multiplier to be inefficient. It cannot stop the switching activity even if the bit coefficient is zero that ultimately results in unnecessary power dissipation.

Booth Multiplier

The booth recoding multiplier is one such multiplier it scans the three bits at a time to reduce the number of partial products. As these three bits the two bits from the current pair and third bit is taken from the high order bit of an adjacent lower order pair. Now after testing each triplet of bits they are converted by booth logic unit into a set of five control signals where it can be used by the adder cells in the array that control the operations performed by the adder cells. Also booth recording reduces the numbers of adders and on the other hand delay required generating the partial sums by observing three bits at a time. In order to speed up the multiplication booth encoding performs many steps of the multiplication simultaneously. Therefore we conclude booth's algorithm takes merit of the consideration that an adder subtracted is nearly as fast and small as a simple adder.

The major drawbacks of booth multiplier are number of add subtract and shift operation becomes variable and become sin convenient for designing parallel multipliers .Also algorithm becomes inefficient when there are some isolated 1's that results in more power consumption due to large number of adders. Thus summing the partially redundant partial products requires as more hardware as representing them in the fully redundant form.

III. PROPOSED SYSTEM

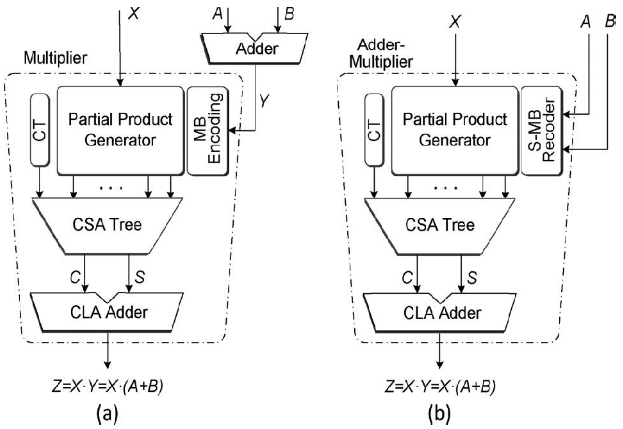

Fig1.Block Diagram

The sum-modified booth (S-MB) recoding technique reduces the number of partial products and increasing speed of calculation. This technique reduces area and power consumption with decreases the critical path delay. The proposed S-MB algorithm is structured, simple and easy to use for signed and unsigned numbers of bits.

Concept Description:-

The optimized design of the AM operator is based on the fusion of adder and MB encoding unit into a single data path block (Fig.1) by direct recoding of the sum Y=A+B to its MB form representation. The FAM component contains only one adder at the end (final adder of the parallel multiplier). As a result, significant area savings are observed and the critical path delay of the recoding process is reduced.FAM Design is a new technique for direct recoding of two numbers in the MB representation of their sum.

S-MB Recoder:-

The S-MB recoder is embedded with encoder block and adder. It is structured with full adders and half adders where the adder and encoding is done in single structure. This block reduces the area of the FAM design.

CSA Tree:-

The carry select adder comes in the category of conditional sum adder. It works on some conditions. The carry and sum are calculated by assuming input carry as 0 and 1 prior the input carry comes. When a carry input arrives, the actual calculated values of carry and sum are selected by using a multiplexer. The conventionl CSA consists of n-bit adder for the LSB and MSB of two n-bit adders. In the MSB adder's, one adder assumes carry input as zero and another assumes carry input as one for performing addition. The carry out calculated from the LSB stage is used to select the actual calculated values of output sum and carry.

IV. RESULT

The simulation result for S-MB1 even is shown below

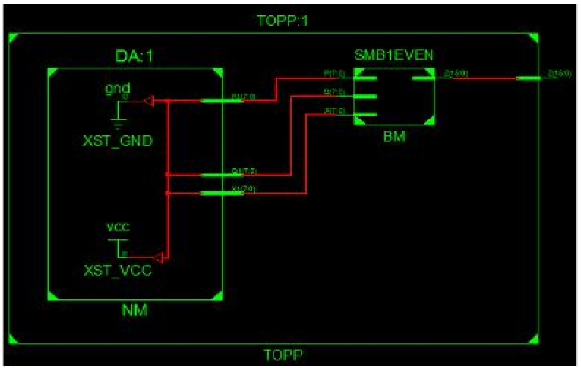

Fig-Schematic diagram

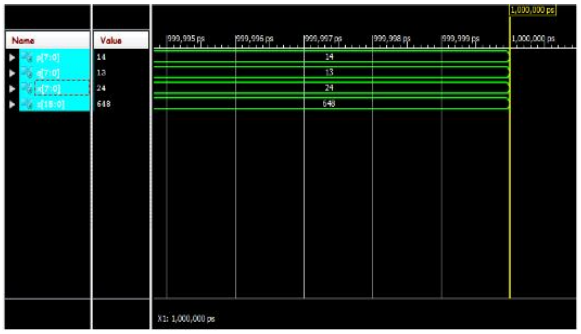

Fig-Simulation Result

ACKNOWLEDGEMENT

A part from the efforts of me, the success of this project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I would like to show my greatest appreciation to Prof.A.S Deshpande i can’t say thank you enough for her tremendous support and help. I feel motivated and encouraged every time. Without her encouragement and guidance this project would not have materialized.

We are also grateful to our H.O.D. Sir Prof. P.R. Badadapure for his kind cooperation offered to us every time regarding the designing and working of the hardware system.
We are also grateful to our principal Sir Dr. S.V.Admane for his kind co operation offered to us every time supporting and motivating us.

We would also like to thank to all the staff members and the laboratory in charges for keeping the labs open and having a number of instruments necessary in the development of the project.

The guidance and support received from all the group members who contributed and are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

V. CONCLUSION

This project we focus on optimizing the design of the Fused-Add Multiply (FAM) operator. We propose a structured technique for the direct recoding of the sum of two numbers to its MB form. We explore three alternative designs of the proposed S-MB recoder and compare them to the existing ones. We expect that the proposed recoding schemes are incorporated in FAM designs, will yield considerable performance improvements in comparison with the most efficient recoding schemes found in literature.

REFERENCES

[1] A. Amaricai, M. Vladutiu, and O. Boncalo, “Design issues and implementations for floating-point divide-add fused,” IEEE Trans. Circuits Syst. II–Exp. Briefs, vol. 57, no. 4, pp. 295–299, Apr. 2010.

[2] E. E. Swartzlander and H. H. M. Saleh, “FFT implementation with fused floating-point oper-ations,” IEEE Trans. Comput., vol. 61, no. 2, pp. 284–288, Feb. 2012.

[3] J. J. F. Cavanagh, Digital Computer Arithmetic. NewYork: McGraw- Hill, 1984.

[4] S. Nikolaidis, E. Karaolis, and E. D. Kyriakis-Bitzaros, “Estimation of signal transition activ-ity in FIR filters implemented by a MAC architecture,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 19, no. 1, pp. 164–169, Jan. 2000.

[5] O. Kwon, K. Nowka, and E. E. Swartzlander, “A 16-bit by 16-bitMAC design using fast 5: 3 compressor cells,” J. VLSI Signal Process. Syst., vol. 31, no. 2, pp. 77–89, Jun. 2002.

[6] L.-H. Chen, O. T.-C. Chen, T.-Y.Wang, and Y.-C. Ma, “A multiplication- accumulation computation unit with optimized compressors and minimized switching

activities," in Proc. IEEE Int, Symp. Circuits and Syst., Kobe, Japan, 2005, vol. 6, pp. 6118–6121.

[7] Y.-H. Seo and D.-W. Kim, "A new VLSI architecture of parallel multiplier–accumulator based on Radix-2 modified Booth algorithm," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp. 201–208, Feb. 2010.

[8] A. Peymandoust and G. de Micheli, "Using symbolic algebra in algorithmic level DSP synthesis," in Proc. Design Automation Conf., Las Vegas, NV, 2001, pp. 277–282.

[9] W.-C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," IEEE Trans. Signal Process., vol. 51, no. 3, pp. 864–874, Mar. 2003.

[10] C. N. Lyu and D. W. Matula, "Redundant binary Booth recoding," in Proc. 12th Symp. Comput. Arithmetic, 1995, pp. 50–57.

[11] J. D. Bruguera and T. Lang, "Implementation of the FFT butterfly with redundant arithmet-ic," IEEE Trans. Circuits Syst. Il, Analog Digit.Signal Process. vol. 43, no. 10, pp. 717–723, Oct. 1996.

[12] W.-C. Yeh, "Arithmetic Module Design and its Application to FFT," Ph.D. dissertation, Dept. Electron. Eng., National Chiao-Tung University, Chiao-Tung, 2001.

[13] R. Zimmermann and D. Q. Tran, "Optimized synthesis of sum-of-products," in Proc. Asi-lomar Conf. Signals, Syst. Comput., Pacific Grove, Washington, DC, 2003, pp. 867–872.

[14] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs. Oxford: Oxford Un-iv. Press, 2000.

[15] O. L. Macsorley, "High-speed arithmetic in binary computers," Proc. IRE, vol. 49, no. 1, pp. 67–91, Jan. 1961.

[16] N. H. E. Weste and D. M. Harris, "Datapath subsystems," in CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Readington:Addison-Wesley, 2010, ch. 11.