International Journal of Scientific & Engineering Research, Volume 4, Issue 8, August 2013
ISSN 2229-5518

ASIC Based Design of High Speed Parallel MAC Based On Radix-4 & Radix-8 Booth Encoding

Reni Kurian, Riboy Cheriyan

Abstract— Real time signal processing applications are increasingly demanded now a day, which are used for multimedia, communication etc. . . . Speed is the major factor for the above application. Parallel MAC is an important element in digital signal processing, because they determine the execution time of the system, i.e. they have the highest delay among basic operational blocks. So in order to improve the speed of the system delay of MAC has to be decreased. This MAC provides high speed in multiplication and multiplication with accumulative addition than general MAC. This paper presents a parallel MAC based on radix-4 & radix-8 booth encodings. In this Paper, several methods to improve the speed of Parallel MAC has been checked, as a result Kogge Stone Ling Adder has been used instead of ordinary CLA to improve the speed.

Index Terms— Carry save adder (CSA) tree, Digital signal processing (DSP), Kogge Stone Ling Adder, Multiplier and- Accumulator(MAC), Radix-4 Booth Encoding, Radix-8 Booth Encoding

1 INTRODUCTION

THE multiplier and multiplier and accumulator (MAC) is an important element of signal processing. When a system is considered the longest delay will be provided by the multiplier. The main parts for a multiplier using Booth algorithm have a Booth encoder, a Wallace tree and a final adder. The most effective way to increase the speed of a multiplier is to reduce the number of the partial products because multiplication precedes a series of additions for the partial products[1],[2] To reduce the number of calculation steps for the partial products, modified Booth algorithm (MBA) has been applied mostly where Wallace tree has taken the role of increasing the speed to add the partial products. In parallel MAC in order to improve the performance compared to serial MAC, multiplication and accumulation is combined and device a hybrid type of carry save adder (CSA). Since the accumulator that has the largest delay in MAC was merged into CSA, thus overall performance was elevated. The CSA tree uses 1's-complement-based radix-4 & radix-8 modified Booth's algorithm (MBA). The CSA generates the least significant bits in advance to decrease the number of the input bits of the final adder. Also, the parallel MAC accumulates the intermediate results in the type of sum and carry bits instead of the output of the final adder.

2 GENERAL MAC STRUCTURE

In this, we discuss basic MAC operation. Basically, multiplier operation can be divided into three operational steps. The first one is booth encoding to generate the partial products. The second one is adder array or partial product compression and the last one is final addition in which final multiplication result is produced. If the multiplication process is extended to accumulate the multiplied result, then MAC consists of four steps. General hardware architecture for MAC is shown in fig1. It executes the multiplication operation by multiplying input multiplier X and input multiplicand Y. After that current multiplication result is added to the previous multiplication result Z as accumulation step [1],[2].

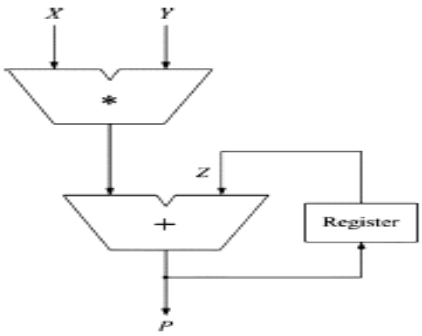

Fig. 1. Hardware architecture of general MAC.

- *Reni Kurian , P.G student, Department of ECE, VLSI & Embedded System, Saintgits College of Engineering, Kottayam, PH-8943047636. E-mail: renikurian88@gmail.com*
- *Riboy Cheriyan ,Asso. Professor, Department of ECE Saintgits College of Engineering, PH-9744600993. E-mail: riboycheriyan@gmail.com*

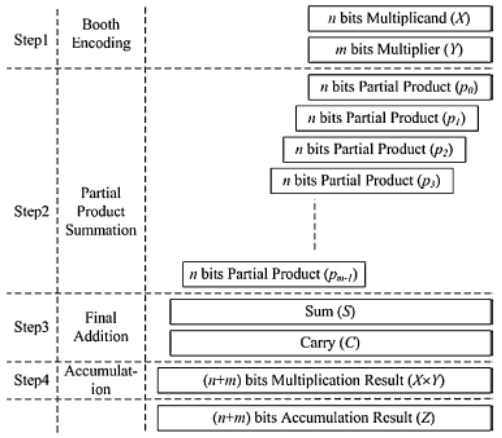

Fig. 2. Basic arithmetic steps for multiplication and accumulation

If N bit data are multiplied, the number of the generated partial products is proportional to N. In order to add them serially, the execution time is also proportional to N. The architecture of a multiplier, which is the fastest, uses Modified Booth encoding that generates partial products and a Wallace tree based on CSA as the adder array to add the partial products. If radix-4 Booth encoding is used, the number of partial products, i.e., the inputs to the Wallace tree, is reduced to half, resulting in the decrease in CSA tree step. In addition, the signed multiplication based on 2's complement numbers is also possible. Due to these reasons, most current used multipliers adopt the Booth encoding[6]-[8].

3.PARALLEL MAC ARCHITECTURE

In this section the derivation of parallel MAC and its architecture is explained

3.1 MAC Architecture

:

If an operation to multiply two N-bit numbers and accumulate into a 2N-bit number is considered, the critical path is determined by the 2N -bit accumulation operation. If a pipeline scheme is applied for each step in the standard design of Fig. 2, the delay of the last accumulator must be reduced in order to improve the performance of the MAC. The overall performance of the parallel MAC is improved by eliminating the accumulator itself by combining it with the CSA function. If the accumulator has been eliminated, the critical path is then determined by the final adder in the multiplier. The basic method to improve the performance of the final adder is to decrease the number of input bits. In order to reduce this number of input bits, the multiple partial products are compressed into a sum and a carry by CSA. The number of bits of sums and carries to be transferred to the final adder is reduced by adding the lower bits of sums and carries in advance within the range in which the overall performance will not be degraded. In addition, to increase the output rate when pipelining is applied, the sums and carrys from the CSA are accumulated instead of the outputs from the final adder in the manner that the sum and carry from the CSA in the previous cycle are inputted to CSA. Due to this feedback of both sum and carry, the number of inputs to CSA increases, compared to general MAC.

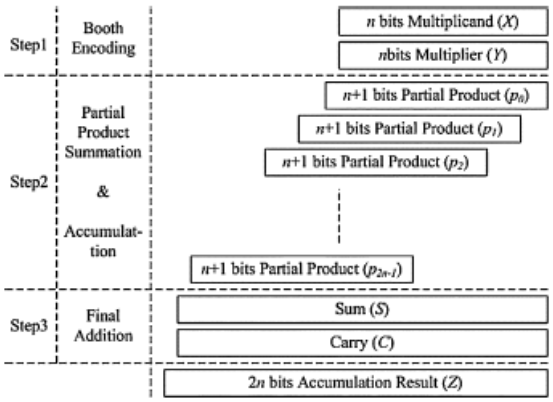

Fig. 3. Arithmetic steps for parallel multiplication and accumulation

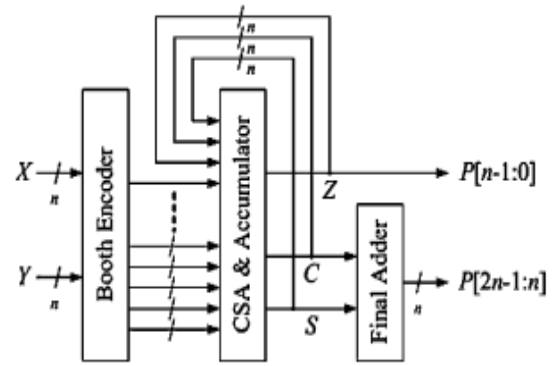

Fig. 4. Hardware architecture of parallel MAC

3.2 Proposed CSA Architecture:

The architecture of the hybrid-type CSA that complies with the operation of the proposed MAC is shown in Fig. 5, Fig 6 which performs 8* 8-bit operation. S_i is to simplify the sign expansion and N_i is to compensate 1's complement number into 2's complement number. S[i] and C[i] correspond to the i^{th} bit of the feedback sum and carry. Z[i] is the i^{th} bit of the sum of the lower bits for each partial product that were added in advance and Z'[i]is the previous result. In addition, $P_i[j]$ corresponds to the i^{th} bit of the j^{th} partial product. Proposed CSA consists of full adders, half adders and carry look ahead adders as shown in Fig 5 and Fig 6. In this, series of CLA adders

are used to generate the lower 8-bits of the final result. The white square in Fig 5 and Fig 6 represents an FA and the black square is a half adder (HA). The rectangular symbol with five inputs is a 2-bit CLA with a carry input. The rectangular symbol with seven inputs is a 3-bit CLA with a carry input.

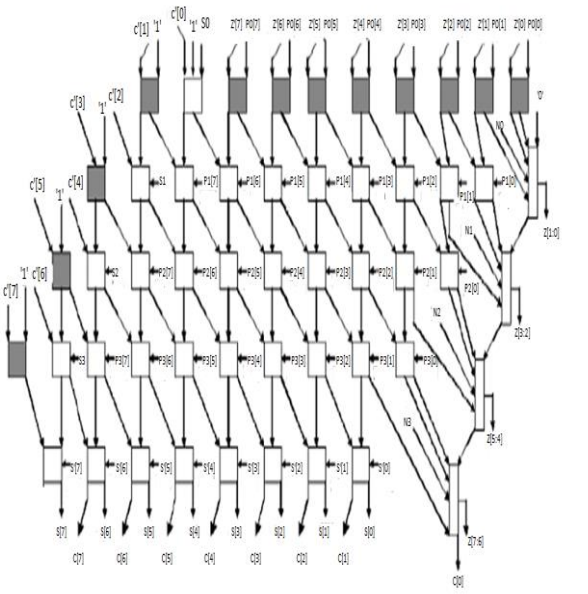

Fig. 5. CSA architecture for 8-bit MAC based on radix-4 booth encoder

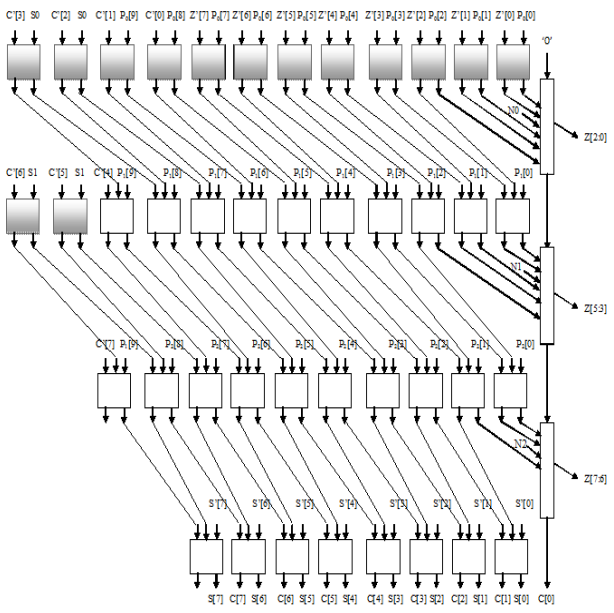

Fig. 6. CSA architecture for 8-bit MAC based on radix-8 booth encoder

4. FINAL ADDER

Final adders are digital circuits that compute the addition of variable binary strings of equivalent or different size. Two types of adder are used for the addition of upper sum and carry bits. These are CLA and Kogge. Traditional CLA is constructed by XOR, AND, and OR gates Stone Ling Adder. They are similar to CLA, main change is that instead of propagating carry a pseudo carry has been propagated. Generation of pseudo carry is faster than carry, so propagation delay of ling adder is less than CLA[9]-[12].

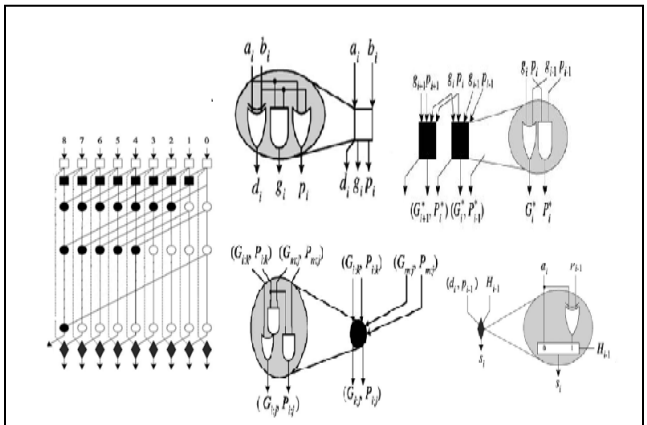

Fig. 7. Kogge Stone Ling Adder

Where white box represent bit generate and propagate, black box represent ling generate and propagate, black circles represent group generate and propagate, and diamond shape represent sum generation part and white circles represent dumpy cells

5. EXPERIMENTAL RESULT

The 8×8 bit parallel MAC based on both the booth encodings (i.e. Radix-4 booth encoding and Radix-8 booth encoding) is designed in VHDL using Modelsim 10.1b and the functionalities of the algorithms are verified by Leonardo Spectrum LS 2009 a-6

Table1.
Comparison result of 8 bit MAC

	MAC 4 CLA	MAC 4 KOGGE STONE LING ADDER	MAC 8 CLA	MAC 8 KOGGE STONE LING ADDER
GATES	1169	1344	1876	2054
DELAY (ns)	10.26	9.43	11.39	10.27

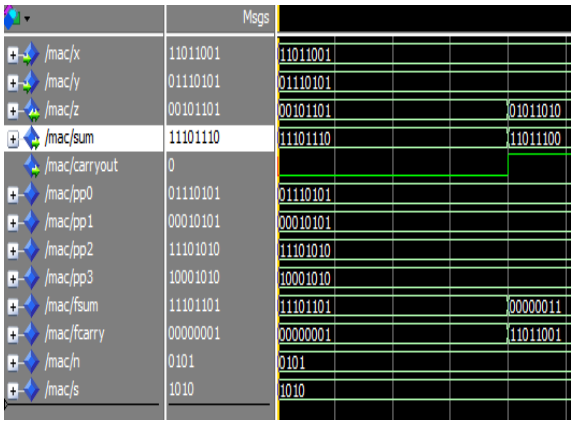

Fig. 8. Simulation result of Pipelined 8-bit MAC based on radix-4 modified booth encoder

6. CONCLUSION

An 8x8 multiplier-accumulator (MAC) is presented in this work. A Radix 4and Radix8 Modified Booth multiplier circuit is used for MAC architecture. Compared to other circuits, the Booth multiplier has the highest operational speed and less hardware count. Parallel MAC based on Radix-4 and Radix-8 is high speed MAC compared to general MAC. By using Radix-4 the number of partial product get reduced to N/2 for N bit and by Radix-8 the number of partial product get reduced to (N+1/3). In this for Radix 4 the number of partial product is four and for Radix 8 it is three. So the CSA array for Radix-8 is smaller than Radix-4. But the encoder for Radix-4 is simpler than Radix-8, so the total performance of Radix-4 is better than Radix-8. In this project an eight bit Kogge stone ling adder has been used as the final adder instead of Carry Look Ahead adder. By using this number of gates gets increased to a small percentage by the time delay gets reduced, i.e. speed get increased.

REFERENCES

[1] Young-Ho Seo, "A New VLSI Architecture of Parallel Multiplier–Accumulator Based on Radix-2 Modified Booth Algorithm" , *IEEE*Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 18, No. 2, February 2010

[2] M.V Sathish, Mrs Sailaja, "*VLSI Architecture of Parallel Multiplier–Accumulator Based on Radix-2 Modified Booth Algorithm*" IJEEE, Volume-1, Issue-1, 2011.

[3] AD Booth, "*A signed binary multiplication technique*" , pp 236-240.1952

[4] C.S.Wallace," *A suggestion for a fast multiplier*"- ,IEEE tran, pp14-17. Feb 1964.

[5] A.R.Cooper," *Parallel architecture modified Booth multipler*," IEEE tran, pp 125-128.1988

[6] Premananthan.G, "*A faster carry save adder in radix-8 booth encoded multiplier*" International Journal of VLSI & Signal Processing Applications, Vol. 2,Issue2,April 2012, ISSN 2231-3133.

[7] Da Huang, Afsaneh Nassery, "*Modified Booth Encoding Radix-4 8-bit Multiplier*".

[8] J.a. hidalgo"*A Radix-8 multiplier unit design for specific purpose*" Dept. de Electrónica, E.T.S.I. Industriales Plaza El Ejido S/N, 29013 Málaga.

[9] Giorgos Dimitrakopoulos and Dimitris Nikolos, "*High-Speed Parallel-Prefix VLSI Ling Adders*" IEEE transactions on computers, vol. 54, no. 2, february 2005.

[10] Bart R. Zeydel, "*Energy-Efficient Design Methodologies: High-Performance VLSI Adders*" Ieee Journal Of Solid-State Circuits, Vol. 45, No. 6, June 2010.

[11] Deepa Yagain1, Dr. Vijaya Krishna A.2, Akansha Baliga3," *Design of High speed adders for Efficient Digital Design Blocks* "Department of Electronics and Communication People's Education Society Institute of Technology

[12] Huey Ling, "High speed adder" *IBM RED DEVELOP volume 25.No 3 May 1981*

[13] Frank k gurkaynak, "*Higher Radix Kogge Stone parallel prefix adderarchitecture*" IEEE international symposium on circuits and systems, may 28-31, 2000,Geneva, Switzerland