Use of Compressors and Ladner Fischer Adder for the Design of Fused Sum Product Unit Using Advanced Booth Recoder

Sneha Vijayan, Haripriya P

PG Student, Department of ECE, Sree Narayana Gurukulam College of Engineering, Kadayiruppu, Kerala, India

Assistant Professor, Department of ECE, Sree Narayana Gurukulam College of Engineering, Kadayiruppu, Kerala,

India

ABSTRACT: This paper presents Radix-8 Modified Booth's Multiplier (MBM) optimized for high speed multiplication by using different compressors and Ladner Fischer Adder (LFA). This paper aims at improvement in the performance of FAM unit. This proposed method is purely based on modified recoding techniques for booth recoding in DSP application An efficient verilog code has been written, successfully simulated on Modelsim 6.3 simulator and Xilinx 13.3 navigator is used for synthesizing the code.The selected device to synthesize the code is xc3s500e-4pq208 of Spartan-3E family. The proposed technique yields considerable reductions in terms of critical delay, area and power consumption of the FAM unit.

KEYWORDS: MBM, LFA , Compressor

I. INTRODUCTION

Digital signal processing is the mathematical manipulation of an information signal. As far as Portable multimedia and digital signal processing (DSP) systems, which may require enhanced processing ability, very low power consumption, and have short design cycle, had become increasingly popular over the past few years. Generally Multimedia as well as DSP applications are very highly multiplication intensive so that the performance and power consumption of these systems are dominated by multipliers.

In general in many applications like specially DSP's kind of things we require arithmetic optimization. Design of arithmetic components combining operations which share data, can lead to significant performance improvements. One way to improve the performance of DSP systems is the fusion techniques. optimized design of the AM operator is based on the fusion of the adder and the MB encoding unit into a single data path block (Fig. 1) by direct recoding of the sum $Y=A+B$ to its MB representation. The fused Add-Multiply (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. The entire performance of the FAM unit is increased by using compressors and parallel prefix adders.

This thesis paper is organized as follows: In Section 2, we discuss the existing design for the Implementation of FAM units and the major drawbacks of the existing design. In Section 3, the proposed scheme is presented. In Section 4 both theoretical analysis and experimental evaluation are given, clearly identifying the advantages of the proposed schemes with respect to area complexity, critical delay. Section 5concludes our work.

II. RELATED WORK

In [1] directly representing the sum of two numbers in its modified booth form by using radix-4 encoding [1] technique. Partial products are added by the carry save adder CSA) and the final stage is carry look ahead (CLA) adder CSA adder takes three inputs and produce sum and carry parallel. There is one CSA. Three partial products are added CSA tree and finally when there are only two outputs. Left out then finally CLA adder is used to produce final result. The main disadvantages of the existing designs are by using CSA reduction tree the number of reduction stages becomes larger so the delay becomes high. In CLA we are calculated the carry bits before the sum. As the number of

bits increases the carry chain becomes more complex and also the area occupation the CLA is very large. The schematics for the existing design is shown in the figure.
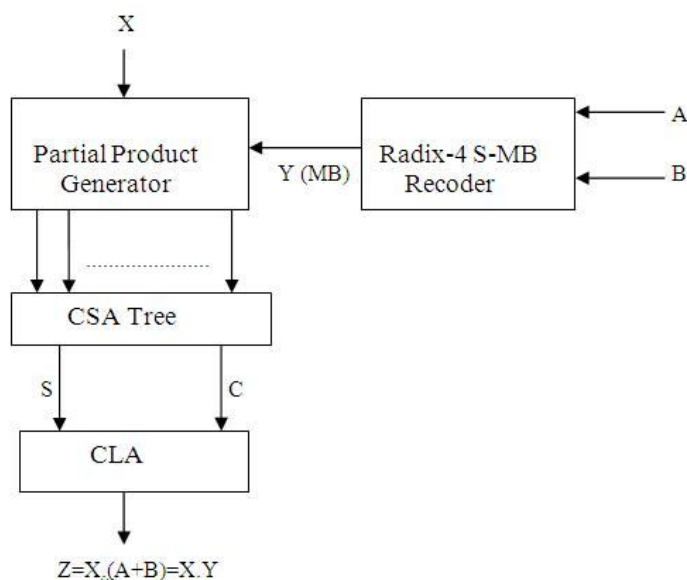

Fig .1 Block Diagram Of Existing Design

III. PROPOSED SYSTEM

In proposed system the S-MB block is implemented by radix-8.The main advantage of radix-8 is that it reduces the number of partial product in multiplication than radix-4 representation .The S-MB algorithm by radix-8 is implemented for odd and even number of bits .The architecture of the proposed system is shown in the figure 2.

In the proposed system partial product reduction done by using 4:2and 5:2 compressors..This different type of compressors reduces the delay and area occupation of the reduction matrix .And finally CLA block is replaced by using a ladner fischer adder to eliminate the high area occupation of CLA.
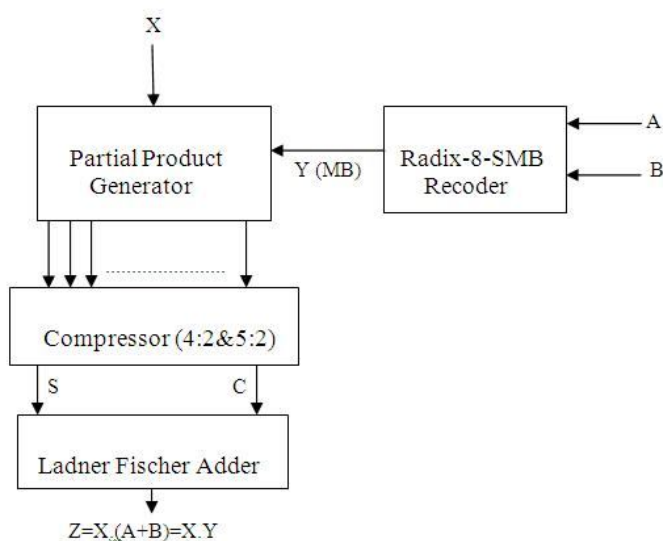

Fig.2. Approximate Compressor, Design 2

A. *Radix-8 S-MB Recoder :*

Booth algorithm (MB) is a prevalent form used in multiplication Radix-8 booth encoder performs the processes of encoding the multiplicand based on the multiplier bits. The sum to modified booth recoder is embedded with adder and encoder block. Fused add multiply design with sum-modified booth recoding technique reduces the number of partial products and increasing speed of calculation. The FAM technique which decreases the critical path delay and reduces area and power consumption, design of this block is structured with conventional half adders, full adders, modified half adders and modified full adders where the adder and encoding is done in single structure. The three S-MB recoding schemes for the design of AM implementation

S-MB1 Recoding Scheme for AM design

S-MB2 Recoding Scheme for AM design

S-MB3 Recoding Scheme for AM design

In each of these recoding techniques we distinguish two cases. First case is bit width of inputs A &B are even and the second case is bit width of A and B are odd. This three types of recoding schemes can be easily applied in either signed (2's complement representation) or unsigned numbers which consist of odd or even number of bits. The main aim of these recoding techniques is to targeting to transform the sum of A and B (Y=A+B) in its MB representation. After representing sum of two numbers in its modified booth form the next steps in the design of the FAM designs are shown in the block diagram. Partial product is generation done according to radix-8 booth encoding and reduction is done by using two types of compressors. Finally LFA as the final product adder.
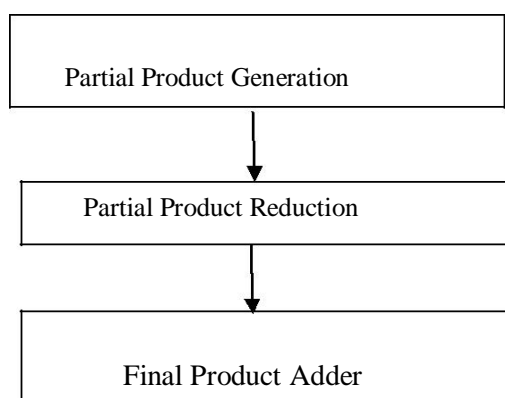

Fig .3.Block Diagram Of Major Steps In Multiplication

B. *Partial Product Generation :*

Partial product is the product formed by multiplying the multiplicand by one digit of the multiplier when the multiplier has more than one digit. Partial product generation is the very first step in binary multiplier. Partial products are used as intermediate steps in calculating larger products .Partial product generator is designed to produce the product. This Booth multiplier is known as radix-8 because it perform the 8 different types of operations on the multiplicand that are +M, +2M, +3M, +4M, –4M, –3M, –2M and –M where M denotes the Multiplicand. To multiply X by Y, the Radix 8 Booth algorithm starts from grouping Y by four bits and encoding into one of {-4,-3,-2, - 1, 0, 1, 2,2,3,4} [7]. The Table 1 shows rules to generate the encoded signals by Modified Booth recoding scheme [8].

In radix-8 Booth Algorithm, multiplier operand Y is partitioned into overlapping group of 4 bits. In firs group, first bit is taken zero and other bits are least Significant two bit of multiplier operand. In second group, first bit is most significant bit of first group and other bits are next two bit of multiplier operand. In third group, first bit is most significant bit of second group and other bits are next two bits of multiplier operand. This process is carried on. For

each group, Partial product is generated using multiplicand operand X. For n-bit multiplier, there are n/2 or n/2+1 groups are formed [9], which generate the partial products as PP [n+m-1], where n, m are the number of bits in multiplier and multiplicand respectively.

Table I. Radix -8 Booth Encoding Table

Yi+2	Yi+1	yi	Yi-1	Multiplier value	Partial product
0	0	0	0	0	Mx0
0	0	0	1	+1	Mx+1
0	0	1	0	+1	Mx+1
0	0	1	1	+2	Mx+2
0	1	0	0	+2	Mx+2
0	1	0	1	+3	Mx+3
0	1	1	0	+3	Mx+3
0	1	1	1	+4	Mx+4
1	0	0	0	-4	Mx-4
1	0	0	1	-3	Mx-3
1	0	1	0	-3	Mx-3
1	0	1	1	-2	Mx-2
1	1	0	0	-2	Mx-2
1	1	0	1	-1	Mx-1
1	1	1	0	-1	Mx-1
1	1	1	1	0	Mx0

.

C. *Compressors*

The conventional adders are the chain of Full adders which generates carries and sum at each level. There was a delay while generating the Final MSB bits of result. During multiplication Booth's Algorithm or any conventional/Modified approach can be used. But during the partial product addition, the conventional adders are not enough to reach the time constraints .The carry travels through the adder to adder. This generates a delay which is consuming for carry propagation and also the efficiency of total circuit gets decreases. During the large operations like large multiplications for DSP applications, large chain of Full adders and half adders will need while following the large chain will lead to generate the larger and larger delays. In our work we design FAM of 16 bits and 17 bits. The latency in the Wallace tree multiplier can be reduced by decreasing the number of adders in the partial products reduction stage. In the proposed architecture, multi bit compressors are used for realizing the reduction in the number of partial product addition stages. The combined factors of low power, low transistor count and minimum delay makes the 5:2 and 4:2compressors, the appropriate choice.

4:2compressor

The 4-2 compressor has 4 inputs A1, A2, A3 and A4 and 2 outputs Sum and Carry along with a Carry in (Cin) and a Carry-out (Cout) as shown in Fig 3. The input Cin is the output from the previous lower significant compressor. The Cout is the output to the compressor in the next significant stage.
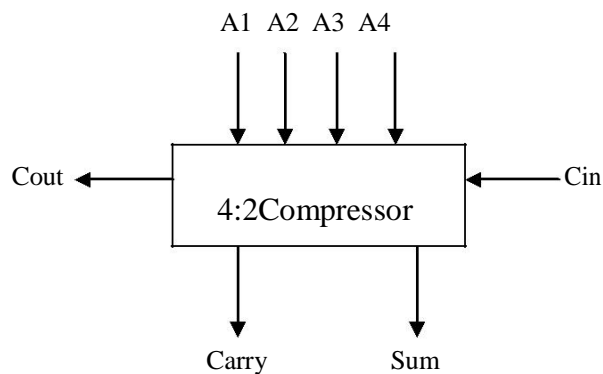
Fig.4. 4:2 compressor

5-2 Compressor

The 5-2 Compressor block has 5 inputsA1,A2, A3,A4,A5 and 2 outputs, Sum and Carry, along with 2 input carry bits (Cin1, Cin2) and 2 output carry bits (Cout1,Cout2) as shown in Fig.4 The input carry bits are the outputs from the previous lesser significant compressor block and the output carry are passed on to the next higher significant compressor block.
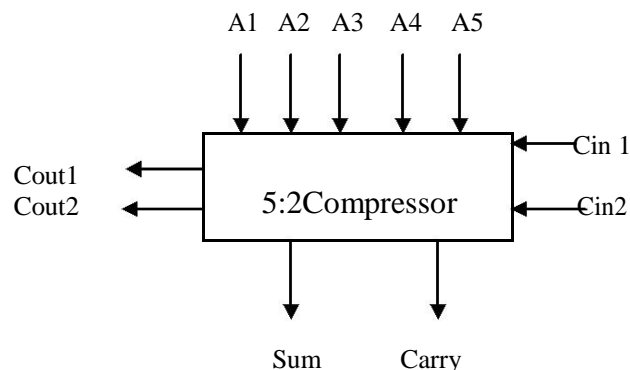

Fig.5. 5:2 compressor

D. *Final Product Adder* :

Parallel Prefix addition is a technique for improving the speed of binary addition. So in our work we used a Ladner fischer adder (LFA) as the final product adder. It is a parallel prefix adder (PPA).A parallel-prefix adder (PPA) gives the best performance in VLSI design. Parallel-Prefix adders perform parallel addition i.e. most important in microprocessors, DSPs, mobile devices and other high speed applications. Parallel-Prefix adder reduces logic complexity and delay thereby enhancing performance with factors like area and power. Therefore the Parallel Prefix adders are requisite element in the high speed arithmetic circuits.

The production of the carriers the prefix adders can be designed in many different ways based on the different requirements. We use tree structure form to increase the speed of arithmetic operation. Parallel prefix adders are faster adders and used for high performance arithmetic structures in industries. Parallel prefix adders are designed from carry look ahead adder as a base. Parallel prefix adders consist of three stages which is shown in the block diagram.
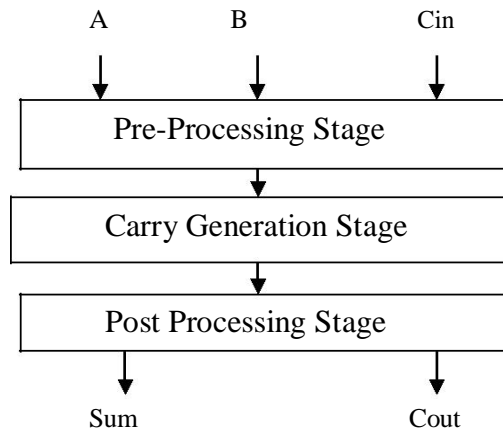
Fig. 6. Block Diagram Of PPA

Pre-processing stage

In the pre-processing stage we compute, the generate and propagate signals are used to generate carry input of each adder .A and B are inputs .Then the propagate and generate signals are given by the equations 1&2.

$$Pi=Ai \wedge Bi \qquad (1)$$

$$GI=Ai \& Bi \qquad (2)$$

Carry generation stage

In this stage we compute carries corresponding to each bit. It uses propagate and generate as intermediate signals which are given by the equations 3&4. Execution is done in parallel form .After the computation of carries in parallel they are divided into smaller pieces.carry operator contain two AND gates , one OR gate.

$$P(i{:}k)=P(i{:}j).P(j{-}1{:}k) \qquad (3)$$

$$G(i{:}k) =G(i{:}j)+(G(j{-}1{:}k) . P(i{:}j)) \qquad (4)$$

Post processing stage

It is the final stage of an efficient Ladner-Fischer adder, the intermediate generate of a first bit is XOR ed with the next bit of propagates then the output is given as sum and it is shown in equation 5.

$$Sum =pi \wedge Gi{-}1 \qquad (5)$$

IV. EXPERIMENTAL RESULTS

This section describes the simulation results of FAM in Verilog using different compressors and LFA .The 13.3 navigator is used for synthesizing the code. The code is synthesized using Spartan3E family with device selected as xc3s500e-4pq208. As compressors form the essential requirement of high speed multipliers .

Table II. Performance Comparison In terms Of Delay

Device utilization	FAM using CSA and CLA	FAM using compressors and LFA	% reduction
Delay calculation	62.14	49.26	37%

Table III. Performance Comparison In Terms Of Area

Device utilization	FAM using CSA and CLA	FAM using compressors and LFA	% reduction
Number of slices	374	298	40%
Number of LUTs	770	690	42%

The results are for the signed and unsigned multiplication of 16-bit multiplicand and 16-bit multiplier depending upon partial product generation by Radix-8 Booth encoder Logic, partial product reduction by 4:2, 5:2 compressors and LFA as final product adder. The choice of optimum multiplier involves two key factors: Area, propagation delay. The MBM reduce the partial products to half to provide the speed advantage. The primary source of propagation delay in circuit is the adder, so compressors used for Wallace tree to add the partial products.

A comparison result is shown in the table II and table III in terms of area and delay for the existing and the proposed algorithm. From the table it is clear that the area is represented by means of number of slices and number of LUT inputs. The area utilization depends upon the number of LUT's (Look Up Table) and SLICES used for synthesizing the code. The proposed design shows 37% reduction in delay and 40% reduction in area than the design of AM using CSA reduction matrix and CLA.

V. CONCLUSION AND FUTURE WORK

Radix-8 Modified Booth Multiplier with FAM technique is designed. Both the delay time and area of FAM is found to be 49.26 ns and 298 slices, 690 LUTs respectively. The delay of proposed FAM is 37% less. There is a trade off existing between proposed and other two in term of delay. The simulation results prove that code is efficient in terms of delay and area. This approach will be well suited for multiplication of numbers with more than 16-bit size for high speed applications. Modified Booth multiplier using Radix-8Modified Booth algorithm and Wallace tree using compressors is very a good technique for high speed applications and its implementation with different logics in VLSI. Further work can be extended for optimization of the given multiplier in terms of both factors as area and delay.

REFERENCES

1. IEEE transactions on circuits and systems—i: regular papers, vol. 61, no. 4, april 2014 1133‖An optimized modified booth recoder for efficient design of the add-multiply operator ― kostas tsoumanis, student member, ieee, sotiris xydis, constantinos efstathiou, nikos moschopoulos, andkiamal pekmestzi
2 .   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.
3 .   E. E. Swartzlander and H. H. M. Saleh, ―FFT implementation with fused floating-point operations, IEEE Trans. Comput., vol. 61, no. 2,pp. 284–288, Feb. 2012.
4.   J. J. F. Cavanagh,Digital Computer Arithmetic. NewYork:McGraw-Hill, 1984.
5.   S. Nikolaidis, E. Karaolis, and E. D. Kyriakis-Bitzaros, ―Estimation of signal transition activity 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
6 . 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.
7 . L.-H. Chen, O. T.-C. Chen, T.-Y.Wang, and Y.-C. Ma, ―A multiplication-accumulation computation unit with optimized compressors and ` Proc. IEEE Int, Symp. Circuits and Syst., Kobe, Japan, 2005, vol. 6, pp. 6118–6121.
8.   Soojin Kim and Kyeongsoon Cho., ―Design of High speed Modified Booth Multipliers Operating at GHz Ranges‖, World Academy of Science, Engineering and Technology,2010.

9.   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.

10.   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.

11.   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.

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

13.   J. D. Bruguera and T. Lang, ―Implementation of the FFT butterfly with redundant arithmetic,‖ IEEE Trans. Circuits Syst. Il,

14.   List I. Abdellatif, E. Mohamed, ―Low-Power Digital VLSI Design, Circuits and Systems,‖ Kluwer Academic Publishers, 1995.

15.   H. Neil. Weste and Kamran Eshraghian,―Principles of CMOS VLSIdesign-A Systems Perspective,‖ Pearson Edition Pvt Ltd. 3rd edition, 2005.

BIOGRAPHY

Sneha Vijayan is an M.Tech student of Electronics and Communication specialized in VLSI & Embedded System, Sree Narayana Gurukulam College Of Engineering, Mahatma Gandhi University. She completed her BTech from Mahatma Gandhi University in 2012. Her research interest area is VLSI design.