An Efficient Design of Sum-Modified Booth Recoder for Fused Add-Multiply Operator

M.Chitra Evangelin Christina
Associate Professor
Department of Electronics and Communication Engineering
Francis Xavier College of Engineering,
Tirunelveli, Tamilnadu, India

P.Maria Jothi Jenifer
Department of Electronics and Communication Engineering
Francis Xavier College of Engineering,
Tiruchendur,Tamilnadu,India

Abstract - This proposed method is purely based on modified recoding techniques for booth recoding in DSP application. The proposed method implements a newly designed recoding technique for modified booth recoding. This technique to implement the direct recoding of the multiplier in its Sum Modified Booth (S-MB) form. The proposed S-MB algorithm is structured, simple and can be easily modified in order to apply either in signed or unsigned numbers, which comprise of odd or even number of bits. Thus Fused Add-Multiply operator is optimized to increase the performance of complex arithmetic operation. It is optimized with three different recoding schemes S-MB1, S-MB2, S-MB3.The sum to modified technique is implemented by Radix-8 Recoder. The proposed technique yields considerable reductions in terms of critical delay, hardware complexity and power consumption of the FAM unit.

I. INTRODUCTION

Digital signal processing (DSP) are widely used in the modern consumer electronics. 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. The performance of the DSP is measured in terms of the amount of hardware and resources required (i,e., space or area); the speed of the execution, which depends on both the throughput and clock rate ;and the amount of the power dissipation or the total energy required to perform a given task. The performance of the DSP application can be affected by a large number of arithmetic operation which requires large architecture.

Recent research activities in the field of arithmetic optimization 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 for efficient implementations of DSP algorithms compared to the conventional ones.

Several architectures have been proposed to optimize the performance of the MAC operation in terms of area occupation, critical path delay or power consumption. 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 . The straightforward 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.

The proposed system optimize the design of AM operators, by introducing fusion techniques which is based on the direct recoding of the sum of two numbers in its Modified Booth (MB) form. 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. The Sum-Modified Booth (S-MB) recoding techniques are efficiently used to implement the direct recoding of the sum of two numbers in its MB form. The proposed technique yields considerable reductions in terms of critical delay, hardware complexity and power consumption of the FAM unit.

II. PROPOSED SYSTEM

A. *Block diagram of proposed system-*

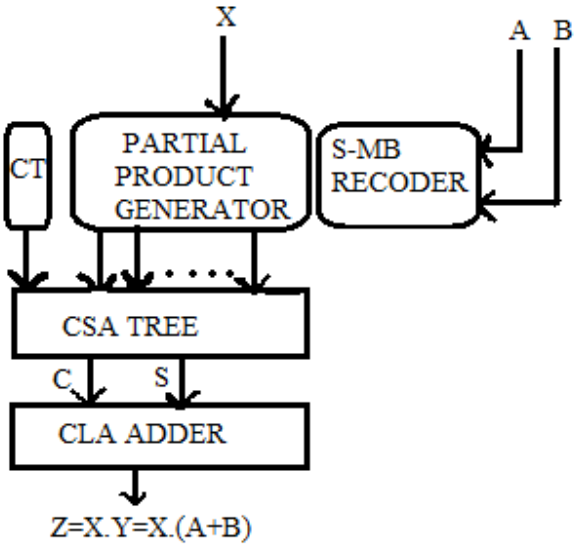

Figure 1.Block diagram

FAM design with sum-modified booth (S-MB) recoding technique reduce 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. 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.

B. *Concept Description-*

An 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.

C. *S-MB Recoder -*

The sum to modified booth recoder is embedded with adder and encoder block .It is structured with half adders and full adders where the adder and encoding is done in single structure.This fused block reduces the area of the FAM design.This S-MB Recoder block is implemented by S-MB Recoder technique.

D. *CSA Tree-*

The carry select adder comes in the category of conditional sum adder. Conditional sum adder works on some condition. Sum and carry are calculated by assuming input carry as 1 and 0 prior the input carry comes. When actual carry input arrives, the actual calculated values of sum and carry are selected using a multiplexer.The conventional carry select adder consists of n- bit adder for the lower half of the bits i.e. least significant bits (LSB's) and for the upper half i.e. most significant bits (MSB's) two n-bit adders. In MSB adder's one adder assumes carry input as one for performing addition and another assumes carry input as zero.

The carry out calculated from the last stage i.e. least significant bit stage is used to select the actual calculated values of output carry and sum. The selection is done by using a multiplexer. This technique of dividing adder in two stages increases the area utilization but addition operation fastens. The basic block diagram for carry select adder is shown in Figure 2.Carry Select Adders (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. The carry select adder partitions the adder into several groups, each of which performs two additions in parallel.

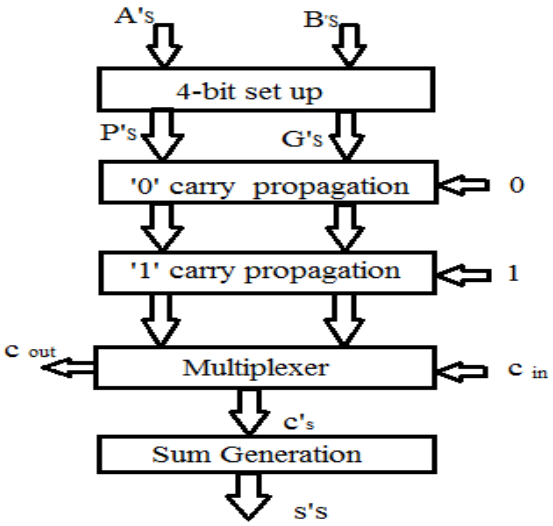

Figure 2.Block Diagram of CSA

Two copies of ripple-carry adder act as carry evaluation block per select stage. One copy evaluates the carry chain assuming the block carry-in is zero, while the other assumes it to be one. Once the carry signals are finally computed, the correct sum and carry-out signals will be simply selected by a set of multiplexers. The 4-bit adder block is RCA. Carry Select Adders acts as a compromise between a small area but longer delay Ripple Carry adder and a large area with shorter delay Carry Look-ahead Adder.

E. CLA Adder

A carry-look-ahead adder (CLA) is a type of adder used in digital logic. A carry-look-ahead adder improves speed by reducing the amount of time required to determine carry bits. The carry-look-ahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits. Most other arithmetic operations, e.g. multiplication and division are implemented using several add/subtract steps. Thus, improving the speed of addition will improve the speed of all other arithmetic operations. Accordingly, reducing the carry propagation delay of adders is of great importance. Different logic design approaches have been employed to overcome the carry propagation problem. One widely used approach employs the principle of carry-look-ahead solves this problem by calculating the carry signals in advance, based on the input signals. This type of adder circuit is called as carry-look-ahead adder (CLA adder).

It is based on the fact that a carry signal will be generated in two cases:

- when both bits Ai and Bi are 1, or
- when one of the two bits is 1 and the carry-in (carry of the previous stage) is 1.

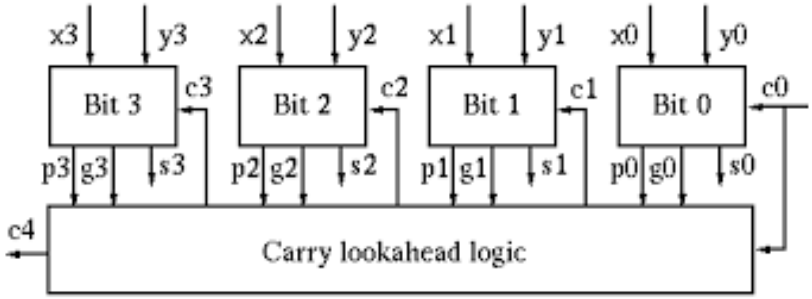

Figure 3.Block Diagram of CLA

F. S-MB Recoding Techniques

Both the conventional and signed HAs and FAs is used to design the three new alternative schemes of the *S-MB* recoding technique by radix-8 Booth encoding. Each of the three schemes can be easily applied in either signed (2's complement representation) or unsigned numbers which consist of odd or even number of bits. Consider that both inputs A and B are in 2's complement form and consist of 2k bits in case of even or 2k+1 bits in case of odd bit-width. Targeting to transform the sum of A and B(Y=A+B) in its MB representation .The three S-MB recoding schemes are:

- S-MB1 Recoding Scheme
- S-MB2 Recoding Scheme
- S-MB3 Recoding Scheme

These S-MB1,S-MB2,S-MB3 Recoding Techniques are implemented by Radix-8 ecoding Techniques.

G. Radix-8 Booth Encoding

Booth algorithm (MB) is a prevalent form used in multiplication and it is a powerful algorithm for signed number multiplication. It treats both the positive and negative number uniformly. Its main advantage is that it reduces by half the number of partial products in multiplication comparing to any other radix-4 representation. Radix-8 booth encoder performs the processes of encoding the multiplicand based on the multiplier bits. Radix-8 Booth recoding is the same algorithm as that of Radix-4. Radix-8 take quartets of bits instead of triplets. The number of partial product can be reduces to n/3 by means of Radix-8 booth encoding where n is the number of multiplier bits.Each quartet is coded as a signed digit using Table 1.

Multiplier Bits				**Radix-8 Booth Encoding**	
$\mathbf{Y_{i+2}}$	$\mathbf{Y_{i+1}}$	$\mathbf{Y_i}$	$\mathbf{Y_{i-1}}$	**Multiplier value**	**Partial Product**
0	0	0	0	0	M x 0
0	0	0	1	+1	M x +1
0	0	1	0	+1	M x +1
0	0	1	1	+2	M x +2

0	1	0	0	+2	M x +2
0	1	0	1	+3	M x +3
0	1	1	0	+3	M x +3
0	1	1	1	+4	M x +4
1	0	0	0	-4	M x -4
1	0	0	1	-3	M x -3
1	0	1	0	-3	M x -3
1	0	1	1	-2	M x -2
1	1	0	0	-2	M x -2
1	1	0	1	-1	M x -1
1	1	1	0	-1	M x -1
1	1	1	1	0	M x 0

Table1:Radix-8 Booth Encoding

H. *Partial Product Generator*

A product formed by multiplying the multiplicand by one digit of the multiplier when the multiplier has more than one digit. Partial products are used as intermediate steps in calculating larger products .Partial product generator is designed to produce the product by multiplying the multiplicand M by 0, 1, -1,2,- 2,-3,-4, 3, 4. For product generator, multiply by zero means the multiplicand is multiplied by "0".Multiply by "1" means the product still remains the same as the multiplicand value. Multiply by "-1" means that the product is the two's complement form of the number. Multiply by "-2" is to shift left one bit the two's complement of the multiplicand value and multiply by "2" means just shift left the multiplicand by one place. . Multiply by "-4" is to shift left two bit the two's complement of the multiplicand value and multiply by "2" means just shift left the multiplicand by two place.

III. EXPERIMENT AND RESULT

Figure 4. Output Of S-MB1 Even

Figure 5. . Output Of S-MB1 Odd

Figure 6.Output of S-MB2 Even

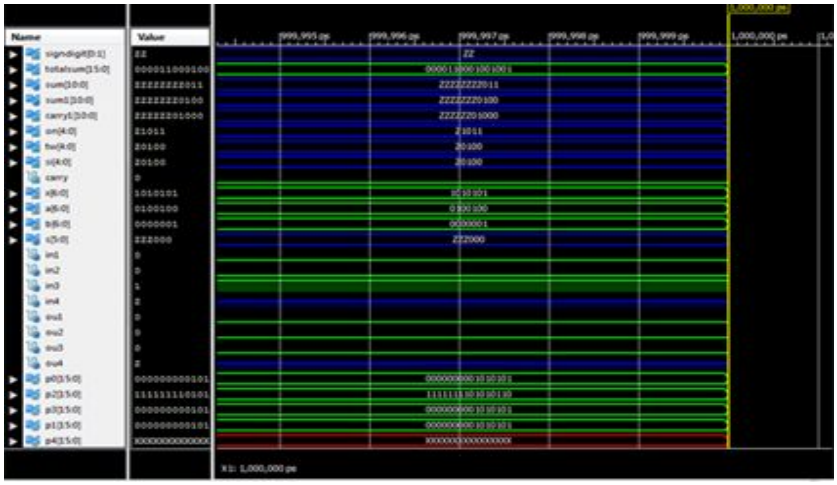

Figure 7.Output of S-MB2 Odd

Figure 8.Output of S-MB3 Even

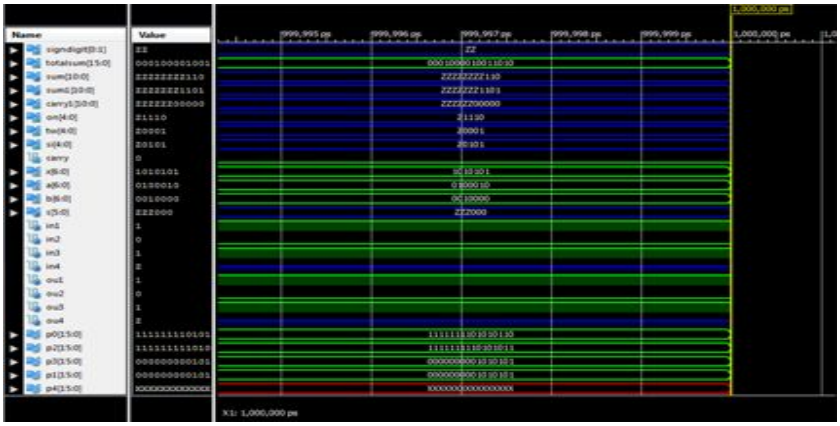

Figure 9.Output of S-MB3 Odd

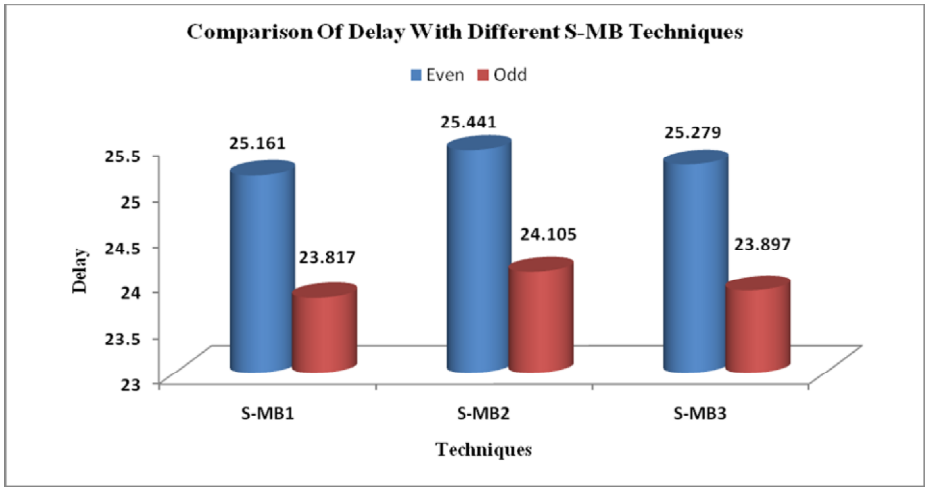

Figure 9.Comparison of Delay with Different S-MB Techniques

IV.CONCLUSION

The Fused Add-Multiply operator is optimized to increase the performance of complex arithmetic operation. It is optimized with three different recoding schemes S-MB1, S-MB2, S-MB3. This proposed system implement the modified booth reconding techniques in radix-8 to achieve the power consumption. The proposed technique yields considerable reductions in terms of critical delay, hardware complexity and power consumption of the FAM unit.

REFERENCES

[1] An Optimized Modified Booth Recoder for EfficientDesign of the Add-Multiply Operator Kostas Tsoumanis, *Student Member, IEEE*, Sotiris Xydis, Constantinos Efstathiou, Nikos Moschopoulos, and Kiamal Pekmestzia IEEE transactions on circuits and systems—i: regular papers, vol. 61, no. 4, april 2014

[2] Amaricai, M. Vladutiu, and O. Boncalo, "Design issues and implementation 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. Swartz lander 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 minimized switching activities," in Proc. IEEE Int, Symp. Circuits and Syst., Kobe, Japan, 2005, vol. 6, pp. 6118–6121.

[8] 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.

[9] 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.