International Journal of Research in Advent Technology, Vol.3, No.2, February 2015
E-ISSN: 2321-9637

A Design of Modified Booth Recoder for Add-Multiply Operator

A.N.Karthigayini[1], A.Arul Rex[2]
P G student, Applied Electronics, Loyola Institute of technology and Science, Nagercoil, India [1]
Email: ankarthiga@yahoo.in
,Assistant Professor, Dept of CSE,Loyola Institute of technology and Science, Nagercoil, India [2]

Abstract-In Digital Signal Processing applicationsComplex arithmetic operations are widely used.Many DSP applications are based on Add-Multiply operation.In the straightforward design of the Add-Multiply (AM) unit, first an adder is used and then the output of adder is given as input to the multiplier. This increases significantly both area and critical path delay of the circuit. To overcome this, fused Add Multiply operator is used.In this paper we focus on the design of the fused Add-Multiply (FAM) unit for increasing performance.Here a technique is used to implement the direct recoding of the sum of two numbers in its Modified Booth (MB) form. Further to improve the overall performance we use modified carry select adder in the multiplier unit instead of carry look ahead adder.

Index Terms-Add-Multiply operation, Modified Booth recoding, arithmetic circuits, fused Add-Multiply operator.

1 INTRODUCTION

In DSP applications large number of arithmetic operations are used as their implementation is based on computationally intensive kernels. The performance of DSP systems are affected by their design regarding the allocation and the architecture of arithmetic units. The design of arithmetic components combining operations which share data, can lead to significant performance improvements. Many DSP applications are based on Add-Multiply (AM) operations. AM unit implement the operation $Z=X\cdot(A+B)$.In conventional design of the AM operator (Fig 1) the inputs A and B are first given as an input to an adder and then the input X and the adder output Y=A+B is driven to a multiplier in order to get the output Z.

In straightforward design of the AM unit, first an adder is used and then its output is given as the input to a multiplier, this increases both area and critical path delay of the circuit. The drawback of using an adder is that it inserts a significant delay in the critical path of the AM. As there are carry signals to be propagated inside the adder, the carry propagation delay is more so the critical path depends on the bit-width of the inputs. In order to decrease this delay, a Carry Look-Ahead adder can be used which, however, increases the area occupation and power dissipation.

To get an efficient design of AM operators, fusion techniques are used based on the direct recoding of the sum Y=A+B to its MB representation. In the fused Add-Multiply (FAM) design it contains only one adder at the end. As a result, significant area savings are observed and the critical path delay of the recoding process is reduced and decoupled from the bit-width of its inputs. In this paper, we present a new technique for direct recoding of two numbers in the MB representation of their sum.

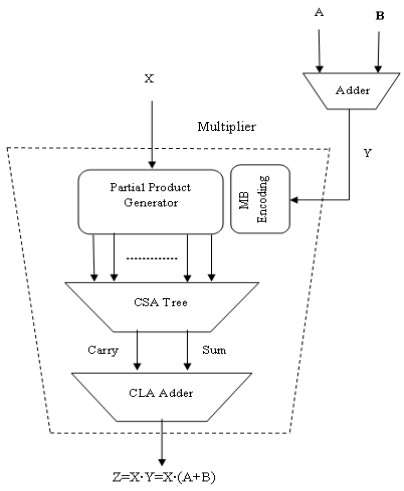

Fig 1 Conventional design of AM operator

1.1 Modified Booth Form

MB is a prevalent form used in multiplication. It is a redundant signed-digit radix-4 en-coding technique. Its advantage is that it reduces the number of partial products by half in multiplication comparing to any other radix-2 representation. Consider the multiplication of 2's complement numbers X and Y with n=2k bits each. The

Multiplicand Y can be represented in MB form as:
$y_j^{MB} = -2_{y2j+1}+y_{2j}+y_{2j-1}$. (1)

Digitsy_j^{MB} can give any one of the values -2,-1,0,+1,+2 where 0≤ j ≤ k-1, correspond to the three consecutive bits y_{2j+1},y_{2j} and y_{2j-1} ,with one bit overlapped and considering that y_{-1} value as zero. Table 1 shows how they are formed by summarizing the MB encoding technique. We can represent each digit by three bits named s, one and two. The sign bit shows if the digit is negative (s=1) or positive (s=0). Signal one shows if the absolute value of a digit is equal to 1 (one=1) or not (one=0). Signal two shows if the absolute value of a digit is equal to 2 (two=1) or not (two=0). Using these three bits we calculate the MB digits by the following relation:
$y_j^{MB} = (-1)^{sj}. [one_j+two_j]$.
Boolean equations to find the MB encoding signals are given as:
$one_j = y_{2j-1} \oplus y_{2j}$
$two_j = (one_j \oplus y_{2j}) \cdot \overline{onej}$
$s_j = y_{2j+1}$ (2)

Table 1 Modified booth encoding table

Binary			y_j^{MB}	MB Encoding			Cin_j
Y_{2j-1}	Y_{2j}	Y_{2j-1}		S_j	one_j	two_j	
0	0	0	0	0	0	0	0
0	0	1	+1	0	1	0	0
0	1	0	+1	0	1	0	0
0	1	1	+2	0	0	1	0
1	0	0	-2	1	0	1	1
1	0	1	-1	1	1	0	1
1	1	0	-1	1	1	0	1
1	1	1	0	1	0	0	0

1.2 FAM Implementation

In the FAM design presented in Fig 2, the multiplier is a parallel one based on the MB algorithm. Let us consider the product X.Y. The term Y is encoded based on the MB algorithm (Section 1.1) and multiplied with X. Both X and Y consist of n=2k bits and are in 2's complement form. The generation of the partial products can be calculated by the equation $pp_j = X.Y_j^{MB}$.After the partial products are generated, they are added, properly weighted, through a Wallace Carry-Save Adder (CSA) tree. Finally, the carry-save output of the Wallace CSA tree is leaded to a fast Carry Look Ahead (CLA) adder to form the final result Z=X·(A+B).

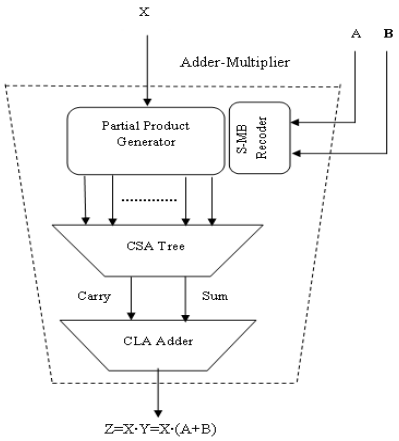

Figure 2 Fused Design of AM Operator

2 OVERVIEW OF SUM TO MODIFIED BOOTH RECODING TECHNIQUE (*S-MB*)

In this recoding technique, we recode the sum of two consecutive bits of the input A (a_{2j},a_{2j+1}) with two consecutive bits of the input B (b_{2j},b_{2j+1}) into one MB digity$_j^{MB}$.As we observe from equation $y_{MB}^j = -2_{y2j+1}+y_{2j}+y_{2j-1}$three bits are included in forming a MB digit.The most significant of them is negatively weighted while the two least significant of them have positive weight.In order to transform the two aforementioned pairs of bits in MB form we need to use signed-bit arithmetic.We design a set of bit-level signed FA to satisfy the above conditions. The inputs and the outputs of the FA are in signed manner.

We use two types of signed FAs which are presented in Table 2 and 3 and fig 3and 4 show the schematics diagram of FA* and FA**. p, q and c_i are the binary inputs andc_0, s are the output carry and sum of the FA.FA* implements the relation $2\cdot c_0$-s = p-q+c_i where the bits s and q are considered negatively signed. Table 2 shows that the output values of FA* are {-1,0,+1,+2}. In FA**, the two inputs p, q are negatively signed and FA** implements the relation $-2\cdot c_0$+s = -p-q+c_i. The output values become {-2,-1, 0, +1} as shown in table 3. As shown in Figure 3 and 4 the signed FAs are implemented using the conventional FA

with the negative inputs and outputs inverted.Equation for calculating carry and sum of FA* are given as:

$$s = p \oplus q \oplus c_i$$

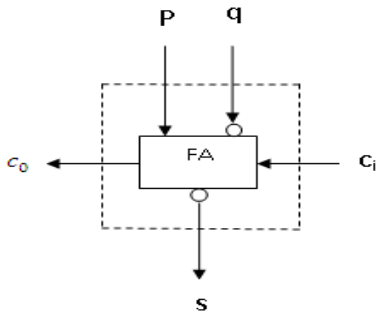

Fig 3 Schematic diagram for signed FA*

Equation for calculating carry and sum FA** are given as:

$$c_0 = ((p \vee q) \wedge \overline{c_i}) \vee (p \wedge q)$$

$$s = p \oplus q \oplus c_i$$

Table 2 Truth table for FA* operation

Inputs			Output value	Outputs	
p (+)	q (-)	c_i (+)		c_0(+)	s(-)
0	0	0	0	0	0
0	0	1	+1	1	1
0	1	0	-1	0	1
0	1	1	0	0	0
1	0	0	+1	1	1
1	0	1	+2	1	0
1	1	0	0	0	0
1	1	1	+1	1	1

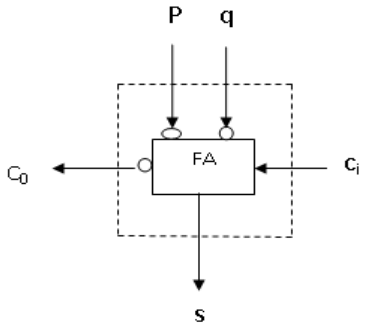

Fig 4 Schematic diagram for signed FA**

2.1 S-MB Recoding technique

Consider thatinputs A and B are in 2's complement

$$c_0 = ((p \vee \overline{q}) \wedge c_i) \vee (p \wedge \overline{q})$$

form. If A and B consist of even number of bits then it contain 2k bits or 2k+1 bits in case of odd bit-width. Consider thatthe bits a_{2j}, a_{2j+1} and $b_{2j}, b_{2j+1}$are the inputs to the j recoding cell in order toas get at its output the three bits that we need to form the MB digit as in "Eq (1)". The sum of A and B is given by the relation $Y=A+B=y_{k\cdot 2^{2k}}+\sum_{j=0}^{k-1} y_j^{MB}\cdot 2^{2j}$

Where $y_{MB}^{j} = -2s_{2j+1}+s_{2j}+c_{2j}$. (3)

Table 3 Truth table for FA** operation

Inputs			Output value	Outputs	
p (-)	q (-)	c_i (+)		c_0(-)	s(+)
0	0	0	0	0	0
0	0	1	+1	0	1
0	1	0	-1	1	1
0	1	1	0	0	0
1	0	0	-1	1	1
1	0	1	0	0	0
1	1	0	-2	1	0
1	1	1	-1	1	1

The encoding of the MB digits y_{MB}^{j}, 0≤j≤k-1, of "Eq (3)" is based on the analysis of section (1.1) "Eq (1)".Bits s_{2j+1} and $s_{2j}$are extracted from the j recoding cell of Fig 5a.

A FA with inputsa_{2j}, $b_{2j}$and b_{2j-1} produces the carry $c_{2j+1}=(a_{2j} \wedge b_{2j}) \vee (b_{2j-1} \wedge (a_{2j} \vee b_{2j}))$ and the sum is given as:$s_{2j} = a_{2j} \oplus b_{2j} \oplus b_{2j-1}$.

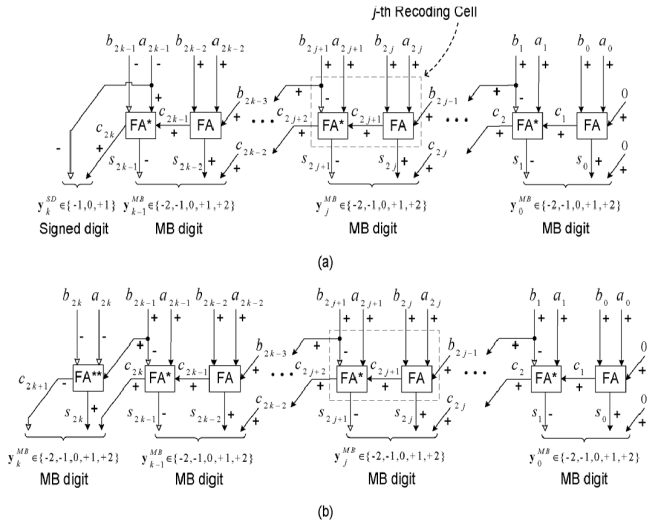

Fig 5 S-MB recoding scheme for (a) even and (b) odd number of bits.

The bit s_{2j+1} must be negatively signed, for that we use a FA* with inputsa_{2j+1}, b_{2j+1}(-) andc_{2j+1}, which produces the carry and the sum denoted as $c_{2j+2}$and s_{2j+1}(-):

$$c_{2j+1} = (a_{2j+1} \wedge \bar{b}_{2j+1}) \vee (c_{2j+1} \wedge (a_{2j+1} \vee \bar{b}_{2j+1}))$$
$$s_{2j+1} = a_{2j} \oplus b_{2j+1} \oplus c_{2j+1} \quad (4)$$

The equation$b_{2j+1} = 2 \cdot b_{2j+1} - b_{2j-1}, b_{2j+1}$is driven to the FA* as negatively signed. We can also use this value with positive sign as an input carry of the subsequent recoding cell.To transform the sum of A and B(Y=A+B) in its MB representationwe consider the initial values of the input to the FA as $b_{-1} = 0$ and$c_0 = 0$.

When we form the MSD of the *S-MB* recoding scheme, we distinguish two cases: In the first case, the bit-width of A and B is even while in the second case, both A and B comprise of odd number of bits. In case of even bits the MSD is a signed digit and is given by the next equation:

$$y^{SD}_{k,even} = -a_{2k-1} + c_{2k} \quad (5)$$

In the second case the MSD $y^{SD}_{k,odd}$ is formed based on$c_{2k+1}, s_{2k} and c_{2k}$. The carry c_{2k+1}(-) and the sum s_{2k} are produced by the FA** with inputs a_{2k}(-), b_{2k}(-) and b_{2k-1}.

2.2 Multiplier design

In the Fused Add-Multiply operator in multiplier unit finally a CLA adder is used but here to reduce the time delay of the circuit we are using the modified carry select adder.After the inputs A and B are encoded based on the S-MB Recoding Technique the recoded values are multiplied with the X value that is the multiplier value to get the partial products.After the partial productsare generated, they are added, properly weighted, through a Wallace Carry-Save Adder (CSA) tree. Finally, the carry-save output of the Wallace CSA tree is leaded to a Carry select adder to form the final result Z=X·(A+B).

3 CONCLUSION

This paper focuses on the design of the Fused-Add Multiply (FAM) operator to make as effective for increasing performance. Here a structured technique for the direct recoding of the sum of two numbers to its MB form is given**.** The recoding scheme is incorporated in FAM design, which yields considerable performance improvements compare to the conventional design. Here we are using a modified carry select adder instead of carry look ahead adder in the multiplier area to decrease the delay.

REFERENCES

[1] 1.E.E.Swartzlander and H.H.M.Saleh. (2012) "FFT implementation with fused floating-point operations,"IEEE Trans. Comput., vol. 61, no. 2, pp. 284–288.

[2] 2.A.Amaricai, M.Vladutiu, and O.Boncalo.(2010) "Design issues and implementations for floating-point divide-add fused," IEEE Trans. CircuitsSyst. II–Exp. Briefs, vol. 57, no. 4, pp. 295–299.

[3] 3.O. Kwon, K. Nowka, and E. E. Swartzlander. (2002) "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.

[4] 4.Y.-H. Seo and D.-W. Kim. (2010) "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.

[5] 5.R. Zimmermann and D. Q. Tran. (2003) "Optimized synthesis of sum-of-products," in Proc. Asilomar Conf. Signals, Syst. Comput.,Pacific Grove, Washington, pp. 867–872.

[6] 6.J. D. Bruguera and T. Lang. (1996) "Implementation of the FFT butterflywith redundant arithmetic," IEEE Trans. Circuits Syst. Il, Analog Digit.Signal Process., vol. 43, no. 10, pp. 717–723.