An Efficient Fused Add-Multiply Unit Design Based on the S-MB Recoder for Signed Digit

Karthikeyan K
PG Scholar, Department of ECE,
Government College of Technology,
Coimbatore, India

Deepa P
Assistant Professor, Department of ECE,
Government College of Technology,
Coimbatore, India

***Abstract*— The basic operations behind the DSP applications are addition and multiplication unit. Here an efficient Fused Add-Multiply (FAM) unit based on the Sum to Modified Booth (S-MB) recoder for signed digit. To perform the addition of numbers in S-MB, signed Half Adder (HA) and signed Full adders(FA) are used. By the use of various combination of Half and Full Adders, the three S-MB recoder technique has been developed, named as S-MB1, S-MB2 and S-MB3. In parallel, the Modified Booth (MB) recoding technique based on Radix-4 encoding has been used to encode the added bits. The Radix-4 algorithm reduces the number of intermediate products by half. So the area and the delay gets reduced. The recoded multiplicand from Radix-4 is multiplied by the multiplier in 2's complement form to generate the intermediate product. The intermediate products are added through a Wallace Carry Save Adder (WCSA) tree along with the correction term to eliminate the errors. The result of the WCSA tree is given to a Fast Carry Look Ahead Adder to obtain the result.**

Keywords*— Add-Multiply operation, arithmetic circuits, Modified Booth recoding, VLSI design

I. INTRODUCTION

An common and very useful combinational logic circuit which can be constructed using just a few basic logic gates. The Adder uses AND and Ex-OR gates and allow us to "add" together single bit binary numbers, to produce outputs called the SUM of the addition and a CARRY called the Carry-out, (C out) bit. One of the main uses for the Binary Adder is in arithmetic and counting circuits.

Fast multipliers are essential parts of DSP systems. The speed of multiply operation is very important in DSP as well as in the general purpose processors. Earlier days, multiplication was done by add, Subtract and shift operations. Multiplication can be done by repeated additions.

Multiply – Accumulator unit has become one of the essential building blocks in digital signal processing applications such as filtering of digital signals, processing of speech signals and in cell phones. Existing MAC unit consists of multiplier and an accumulator that contains the sum of the previous consecutive products. Modern computers may contain a separate MAC, has a separate multiplier implemented in combinational logic followed by an adder and an accumulator stores the outcome. The output is given back to one input of the adder, on every clock cycle, the outcome of the multiplier is added to the content of the accumulator.

A Fused Multiply–Add is a floating-point multiply–add operation performed in single step, with single rounding. An unfused Multiply–Add would compute the product and round it to N significant bits, add the result to another number, and round back to N bits. A fast FMA can be faster and improve the accuracy of many computations that involve the accumulation of products. Fused Multiply–Add can usually be relied on to give more accurate results.

In this paper, a new technique is exploited for direct recoding of two numbers in the MB representation of their sum. An optimized design of the AM operator is based on the fusion of the adder and the Modified Booth (MB) encoding unit into a single data path block by direct recoding of the sum to its MB representation. The Fused Add-Multiply (FAM) component contains only one adder at the end (final adder of the parallel multiplier). So that area savings are observed and the critical path delay of the recoding process is reduced and decoupled from the bit-width of its inputs.

The rest of the paper is organized as follows: In Section II, we discuss about the literature survey. Section III describes on the Fused Add-Multiply (FAM) Unit. Section IV gives the simulation, synthesis and analysis of the FAM unit Section V concludes our work.

II. LITERATURE SURVEY

An array multiplier is a parallel multiplier. It does the shifting and addition operation in a single step. It adds the partial products parallel. The structure of array multiplier is very compact and the resulting layout is very economic This multiplier has simple operation but it expend more time, so it will make the multiplication process very slow [6].

High speed and low power DSP application make use of array multipliers in parallel. One kind of parallel array multiplier is Braun's multiplier. The delay of the multiplier depends on the delay of the full adders and also on the delay of the last stage adder. One of the major disadvantages of the multiplier is that the number of components required increases quadratically with the number of bits which makes the multiplier inefficient. [7].

Baugh-Wooley algorithm is used for binary multiplication of Signed numbers. This algorithm creates AND terms and then they are passed through an array of half-adders and full-adders with the Carry-outs chained to the next most significant bit. Signed operands may also be multiplied using a Baugh-Wooley multiplier.

Combinational Multipliers is used for the multiplication of two unsigned numbers and signed numbers. Generation of intermediate products is easy in binary multiplication. If the multiplier bit is a 1, the product is an correctly shifted copy of the multiplicand; if the multiplier bit is 0 then the product is 0. In most of the systems combinational multipliers are slow and take lot of area.

A Wallace tree multiplier is an efficient hardware implementation of a digital circuit that multiplies two integer values. Wallace tree reduces the number of partial products [5].

Wallace tree has three steps:

- Multiply each bit of multiplier with same bit position of multiplicand. Depending on the position of the multiplier bits generated partial products have different weights.
- Reduce the number of partial products to two by using layers of full and half adders.
- After second step we get two rows of sum and carry, add these rows with conventional adders.

Wallace tree multiplier has small delay. The number of logic levels required to carry out the summation can be reduced with Wallace tree. The major disadvantages of Wallace tree is complex layout and has irregular wires.

The Booth algorithm forms the base of Signed number multiplication algorithms that are easier to implement at the hardware level and speed up signed multiplication. The algorithm developed by Booth is based upon recoding the multiplier, to a recoded value, without changing the multiplicand(x) unchanged. In Booth recoding, each digit of the multiplier may assume signed, unsigned and zero values. It uses a superior notation, called signed digit (SD) encoding, to express these signed digits and is given in table 2.1. In SD encoding +1 is expressed as 1 and 0 as 0, but -1 is expressed as 1.

TABLE I. BOOTH RECODING TABLE FOR RADIX-2

Y i	Yi-1	Recoded value (z)
0	0	0
0	1	+
1	0	-
1	1	0

The Booth recoding procedure is as follows:

- Add 0 to the right of lsb of multiplicand
- Make pairing of 2 bits from right to left and mark corresponding recoded value.
- 00 or 11 → 0(do nothing)
- 01→ +1 (add multiplicand to partial product)
- 10→-1 (subtract multiplicand from partial product)

The Radix-2 booth multiplier has two drawbacks:

- Inconvenient while designing parallel multipliers
- Inefficient when there is isolated number of 1's

The literature survey reveals that there are various add and multiply unit each has its own advantages and disadvantages, But there is no specific fused add.

III. EFFICIENT FUSED ADD-MULTIPLY UNIT

This chapter presents the design of an efficient Fused Add- Multiply (FAM) Unit using Sum to Modified Booth (MB) Recoder for both Signed and Unsigned numbers. Addition and the encoded multiply units are fused into a single block. Hence the propagation delay is reduced. It uses Signed bit Half Adder and Signed bit Full Adder for adding the numbers instead of conventional adders. For multiplication, redundant Signed digit Radix - 4 method is used. It decreases the number of intermediate products by half when compared to the conventional radix-2 booth multiplication method.

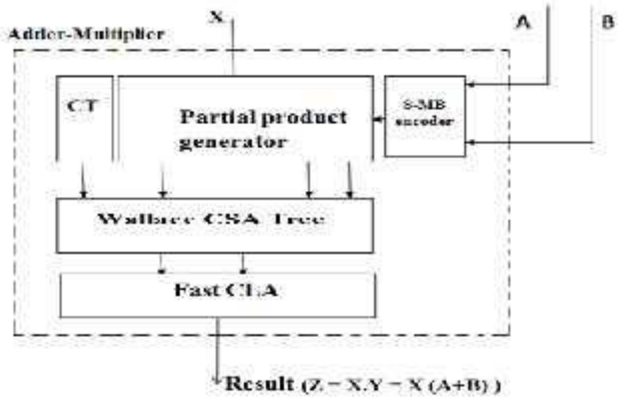

Fig 1. Block diagram of Fused-Add Multiply unit

A. Adder Unit

FAM uses signed Half Adder (HA) and signed Full Adder (FA) for adding the numbers considering that their inputs and outputs are signed.

1. Signed Half Adders

*HA**

The binary inputs are considered as a and b, both are positive .The outputs are represented as carry (c) and sum (s) of a HA. The sum is considered as negative.

TABLE II. TRUTH TABLE FOR HA*

a (+)	b (+)	Sum(s) (-)	Carry (c)(+)
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1

By k-map simplification, the sum and carry equations are given by,

Sum = a exor b (1)

Carry = a + b (2)

HA DUAL OPERATION*

The binary inputs are a and b, both are positive . The sum is considered as negative.

By k-map simplification, the sum and carry equations are given by,   Sum = a exor b (3)
Carry = (~a) . b (4)

TABLE III. TRUTH TABLE FOR HA* DUAL OPERATION

a(-)	**b(-)**	**Sum(+)**	**Carry(-)**
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1

*HA***

The binary inputs are considered as a (signed) and b (unsigned).The outputs are carry (c) and sum (s) of a HA** . Where the sum is considered negatively signed. And the output takes one of the values 0,1.

TABLE IV. TRUTH TABLE FOR HA**

p (-)	**q (+)**	**c(+)**	**s(-)**
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

By k-map simplification, the sum and carry equations are given by,
Sum = p exor q (5)
Carry = (~p) . q (6)

2. *Signed Full Adders*

FA*

The binary inputs are considered as a (signed), b (unsigned),cin (carry in).The outputs are carry (c) and sum (s) of a FA*. Where the sum is considered negatively signed and the carry output is positively. And the output takes one of the values 0,1.

TABLE V. TRUTH TABLE FOR FA*

p (+)	**q (-)**	**cin (+)**	**c(+)**	**s(-)**
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

By k-map simplification, the sum and carry equations are given by,
Sum= a exor b exor cin (7)
Carry = (a+(~b)) cin+ a.(~b) (8)

*FA***

The binary inputs are a , b both are signed, cin(carry in).The outputs are carry (c) and sum (s) of FA. The sum is positive and the carry output is negative. The output takes one of the values {0,±1,±2}.
By k-map simplification, the sum and carry equations are given by,
Sum = a exor b exor cin (9)
Carry =(a+(~b)) cin+ a.(~b) (10)

TABLE VI
TRUTH TABLE FOR FA**

p (-)	q (-)	cin (+)	c (-)	s (+)
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

B. *S-MB Recoding Schemes For Signed & Unsigned Numbers*

The three forms of S-MB recoding scheme based on FA, FA*, FA**, HA, HA, HA** are

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

1. *S-MB1 Recoding Scheme*

The first scheme of the proposed recoding technique is referred as S-MB1 scheme and is given in Fig. 3.2. It uses FA and FA*.For adding two,2 bit numbers one set of FA and FA* used. For adding two,8 bit numbers 4 sets of FA and FA* are used. Sum bits from the FA & FA* and carry out from the FA* are used to form the encoded digits.

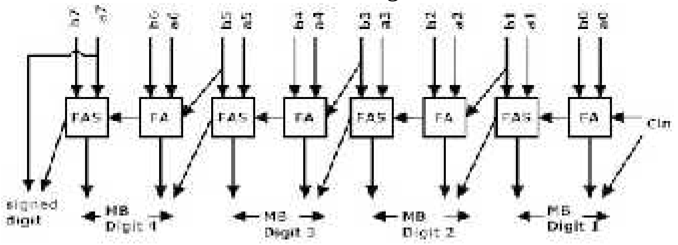

Fig.2. Block diagram for S-MB1 Recoding Scheme for 8 bits

If sign=0, then the result of addition of two numbers is positive. If sign=1, then the result of addition of two numbers is negative.

2. *S-MB2 Recoding Scheme*

The second scheme of the proposed recoding technique is referred as S-MB2 and is given in Fig. 3.4.This scheme uses FA , HA , HA*.For adding two,2 bit numbers one set of FA,HA,HA* are used. For adding two,8 bit numbers 4 sets of FA,HA,HA* are used. Sum bits from the FA & HA* and carry out from the HA* are used to form the encoded digits.

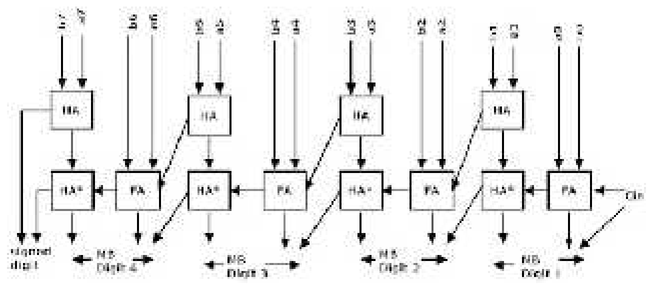

Fig.2. Block diagram for S-MB2 Recoding Scheme for 8 bits

Sign=0, if the result of addition of two numbers is positive. Sign=1,if the result of addition of two numbers is negative.

3. *S-MB3 Recoding Scheme*

The third scheme of the proposed recoding technique is referred as S-MB3 and is given in Fig 3.6 .This scheme uses FA , HA* , HA**. For adding two,2 bit numbers one set of FA,HA*,HA** are used. For summing two,8 bit numbers 4 sets of FA,HA*,HA** are used. Sum bits from the FA & HA** and carry out from the HA** are used to form the encoded digits.

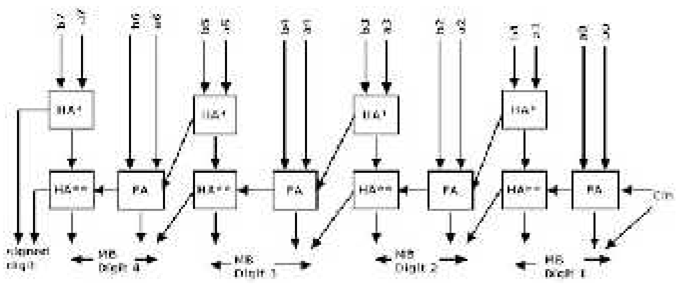

Fig.3. Block diagram for S-MB3 Recoding Scheme for 8 bits

Sign=0, if the result of addition of two numbers is positive. Sign=1,if the result of addition of two numbers is negative.

C. *Modified Booth Form*

Modified Booth (MB) is a prevalent form used in multiplication. It is a redundant signed-digit radix-4 encoding technique. The main advantage is that it reduces by half the number of intermediate products in multiplication comparing to any other radix-2 representation. The basic step of MB recoding is as follows,
Step 1: Each digit is represented by three bits called sign, one and two
Step 2: sign=1, if the digit is negative otherwise sign=0
Step 3: If the value of a digit equals to 1, one=1, else one=0
Step 4: If the value of a digit equals to 2, two=1, else two=0
Equations for finding encoding signals are given by,

$$\mathrm{Sign(S)}=Y_{2j+1} \quad (11)$$

$$\mathrm{One}=Y_{2j-1}\ \mathrm{Xor}\ Y_{2j} \quad (12)$$

$$\mathrm{Two}=(Y_{2j+1}\ \mathrm{Xor}\ Y_{2j})\mathrm{One}^* \quad (13)$$

The table for Modified Booth form is given in TABLE VII.

TABLE VII. MODIFIED BOOTH ENCODING TABLE

Binary value			Output value	Sign	One	Two
y_{2j+1}	y_{2j}	y_{2i-1}				
0	0	0	0	0	0	0
0	0	1	+1	0	1	0
0	1	0	+1	0	1	0
0	1	1	+2	0	0	1
1	0	0	-2	1	0	1
1	0	1	-1	1	1	0
1	1	0	-1	1	1	0
1	1	1	0	1	0	0

D. *Partial Product Generation*

After the encoding, the recode d multiplicand is multiplied with the multiplier and the partial products are produced. Intermediate products are expressed as,

$$PP_j = X.y_j^{mb} = (\sim p_{j,i})2^n + \sum_{i=0}^{n-1} p_{j,i}2^i \quad (14)$$

Where,

X → multiplier
Yj →encoded multiplicand
$p_{j,i}=(x_i \text{ xor } s_j)one_j + (x_{i-1} \text{ xor } s_j)two_j$ (15)
n=2k (input bit width)
i →0 to n-1;j → 0 to k-1
x_i→ Each bit of multiplier (0< i<n-1)
s_j, one_j, two_j→ Encoding signals

E. *Wallace Carry Save Tree and Fast CLA adder*

The multiplier architecture consists of intermediate product generation stage, reduction stage and addition stage. The dormancy of the Wallace tree multiplier can be decreased by scaling down the number of adders in the partial products reduction stage.

After generating the partial products, they are added along with the correction terms through a Wallace carry save adder (CSA Tree).Correction term is used for reducing the mean square error and the truncation error in multt ipliers.
Correction term is expressed as,

$$CT= \sum_{j=0}^{k-1} c_{in,j}.2^{2j} + 2^n(1+\sum_{j=0}^{k-1} 2^{2j+1}) \quad (16)$$

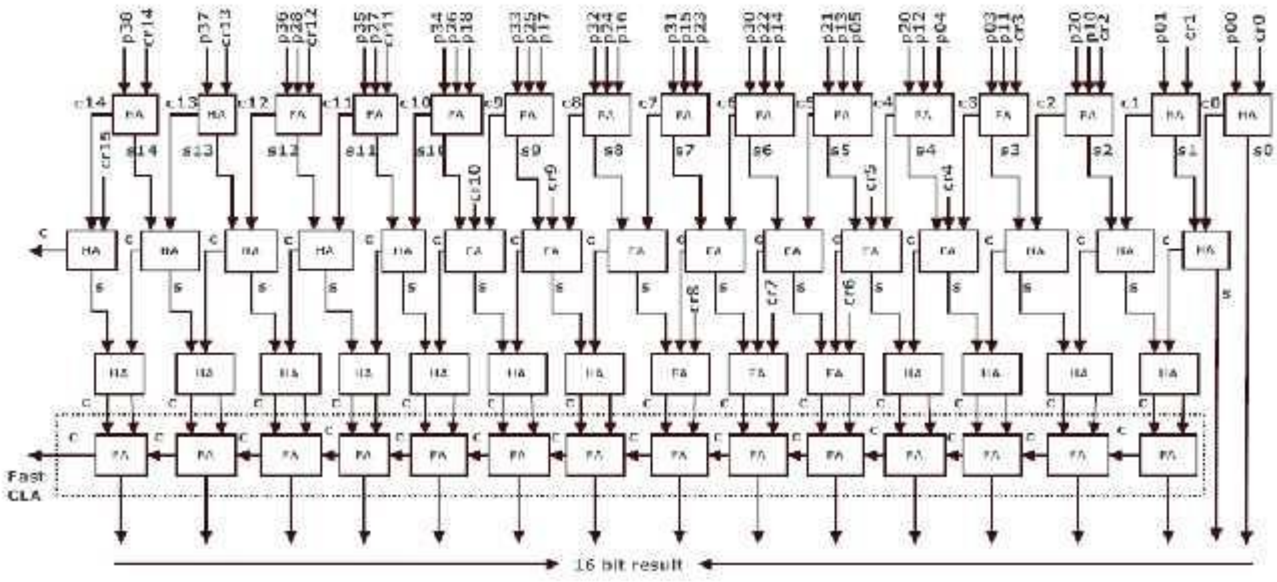

Fig.4. Overall structure of Wallace Carry Save Adder Tree and Fast CLA for adding partial products

Where,

$$c_{in,j} = (one_j + two_j)s_j \tag{17}$$

one_j, two_j, s_j are encoding signals

j ranges from 0 to k-1

The output of Wallace carry save tree is given to the Carry Look Ahead. Propagate and generate terms are calculated and the final result is produced.

The final result (z) is expressed as,

$$z = X.Y = CT + \sum_{j=0}^{k-1} PP_j . 2^{2j} \tag{18}$$

CT→ Correction Term
PPj→ Partial Products

IV. RESULTS AND DISCUSSION

The FAM has been analyzed for area, power and delay and is given in Table 4.1. From the analysis it could be observed that the S-MB1, S-MB2, and S-MB3 show as competitive performance.

TABLE VIII. ANALYSIS OF FAM

Different techniques	Area (um^2)	Power (mW)	Delay (ns)
S-MB1	4264	0.686815	8.573
S-MB2	4258	0.687116	3.004
S-MB3	4258	0.672498	6.566

A. *Comparison between Existing Method and FAM Unit*

FAM is compared with the existing method for area, power and delay and is given in table 4.2. From the table, it could be observed that area and delay is 24.20 % and 63 % less compared to the existing method with a minimum of 52.46% trade-off in power.

TABLE IX. COMPARISON BETWEEN EXISTING METHOD AND FAM UNIT

Parameters	Existing Method	Proposed Method
Area(um^2)	5618	4258
Power(mW)	0.441104	0.672498 (increased due to more number of signals)
Delay(ns)	8.142	3.004

V. CONCLUSION

The FAM units depending on three Sum To Modified Booth Recorder for Signed and unsigned Digit has been successfully simulated using Xilinx ISE Design Suite 14.2 and synthesized by using Cadence Design Tool in 180 nm technology. The Area, Power and Delay of the three FAM units based on three SMB techniques were analyzed and it was compared with the Existing architectures for AM unit. The suggested architecture is found to be efficient in terms of area and delay.

REFERENCES

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

[2] Dr DC Hendry,"Multipliers - Carry Save and Wallace Tree"

[3] VLSI IP, "Booth Multiplier Implementation of Booth's Algorithm using Verilog RTL", http://www.vlsiip.com

[4] NTU, "Booth encoded multiplier",Graduate institute of electronics engineering

[5] Sukhmeet Kaur, Suman and Manpreet Signh Manna, "Implementation of Modified Booth Algorithm (Radix 4) and its Comparison with Booth Algorithm (Radix-2)", Advance in Electronic and Electric Engineering,ISSN 2231-1297, Volume 3, Number 6 (2013), pp. 683-690,© Research India Publications

[6] Soniya, Suresh Kumar,M.Tech. Student U.I.E.T., M.D.U. Rohtak Assistant Professor, U.I.E.T., M.D.U. Rohtak, "A Review of Different Type of Multipliers and Multiplier-Accumulator Unit", IJETTCS, Volume 2, Issue 4, July – August 2013 ISSN 2278-6856

[7] Anitha , Alekhya Nelapati, Lincy Jesima, V. Bagyaveereswaran, "Comparative Study of High performance Braun's Multiplier using FPGAs", IEEE member, VIT University, Vellore, IOSR Journal of Electronics and Communication Engineering (IOSRJECE) ISSN: 2278-2834 Volume 1, Issue 4 (May-June 2012), PP 33-37

[8] Smiksha ECE, UIET A.P.,"Analysis, Simulation and Comparison of Different Multiplier Algorithms", , International Journal of Management, IT & Engineering

[9] Sukhmeet ,ECE, SSIET (P.T.U), Derabassi, Punjab, India,"Implementation of Modified Booth Algorithm (Radix 4) and its Comparison with Booth Algorithm (Radix-2)", Advance in Electronic and Electric Engineering,ISSN 2231-1297, Volume 3, Number 6 (2013), pp. 683-690 © Research India Publications http://www.ripublication.com/aeee.htm

[10] Digital Logic Design, "A Combinational Multiplier Using the Xilinx Spartan II FPGA"

[11] Sandeep Shrivastava, "Implementation of Radix-2 Booth Multiplier and Comparison with Radix-4 Encoder Booth Multiplier", , International Journal on Emerging Technologies 2(1): 14-16(2011) ISSN : 0975-8364

[12] Sakshi Rajput, Asst. Professor, Deptt. of Electronics and Communication, Maharaja Surajmal Institute of Technology, New Delhi, India, "High Speed and Reduced Power – Radix-2 Booth Multiplier", IJCEM International Journal of Computational Engineering & Management, Vol. 16 Issue 2, March 2013,ISSN (Online): 2230-7893www.IJCEM.org.