IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 6, Ver. II (Nov. - Dec. 2016), PP 13-20
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

A New VLSI Architecture for Modified for Add-Multiply Operators using Modified Booth Recoding Technique

K Nunny Praisy[1] and K S N Raju[2]
[1]*M.Tech-VLSID, SVECW (Autonomous), Bhimavaram, Andhra Pradesh.*
[2]*Assistant Professor, SVECW (Autonomous), Bhimavaram, Andhra Pradesh*

***Abstract**: The modified algorithm has made multiplication easy. The algorithm consists of recoding table; it is used to minimize the partial products of the multiplier. We introduced an optimized design of add multiply operator which is based on a single data path. In this three different techniques are introduced (SMB1, SMB2, and SMB3) for the even and odd bit. Modified booth algorithm is mainly used to reduce the number of partial products. This design is used to achieve high performance and to improve the accuracy, reduction in power consumption and critical area. The recoding technique implementations and comparison has done with existing and designed modified booth recoder.*

I. Inroduction

The multipliers are used to perform the multiplication operation of the arithmetic circuit using shift and add operations. The area-efficient which is parallel to sign magnitude of two N-bit numbers produce N-bit product. The performance can be increased by sharing the data in the arithmetic Operations. The Multiply-Accumulator and Multiply-Add is used to increase the efficiency of processor in adder and multiplier. The DSP performance caused with more number of arithmetic operations which can be obtain in large architectures. This booth multiplier contains booth recoder which is used to recode the input of the multiplier. By using adder and multiplier we can reduce the critical path and delay of the circuit.The block diagram of the AM operator is shown in FIG: 1,

FIG:1 AM operator with sum of A and B operator in its MB representation.

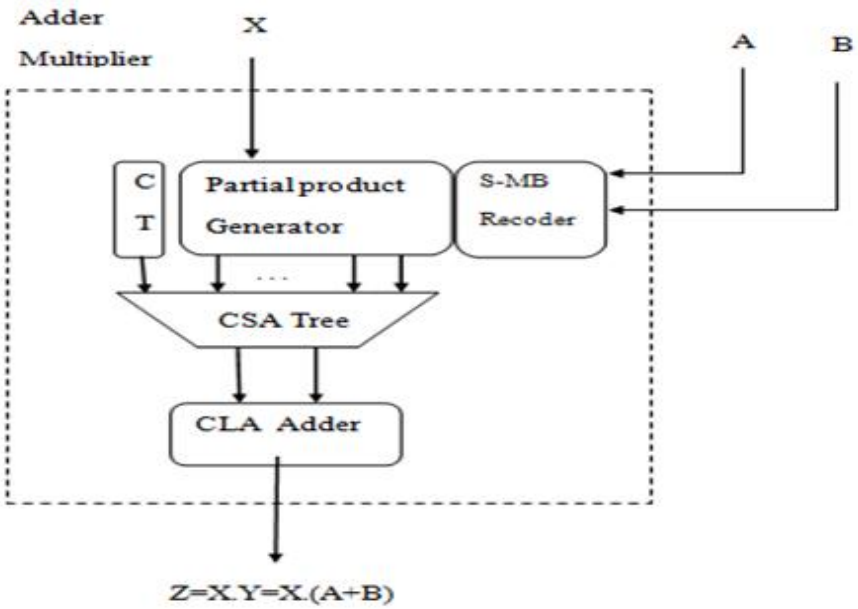


Table 1:

Binary			Y_j^{MB}	MB Encoding			Input Carry
Y_{2j+1}	Y_{2j}	Y_{2j-1}		Sign=s_j	×1=one$_j$	×2=two$_j$	$C_{in,j}$
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

Modified Booth Recoded Table

The modified booth form is used for multiplication and it is a redundant signed-bit of radix-4 encoding. It reduces half the number of partial product in multiplier. If we consider the two numbers X and Y contains n=2k bits and the multiplicand Y is represented in its modified booth form.

$Y_j^{MB} = -2_{y2j+1} + y_{2j} + y_{2j-1}$

$Y_j^{MB} \sum \{-2,-1,0,+1,+2\}$, $0 \le j \le k-1$, it gives the three consecutive bits Y_{2j+1}, Y_{2j} and Y_{2j-1} considering $Y_{-1}=0$ with one bit overlapped the MB encoding table is shown in Table1. Three bits are represented in each digit as s, one, and two. If the digit s is

Negative (s=1) or positive (s=0). The sign shows the exact value of the digit is 1 (one=1) or not (one=0), sign 2 shows the exact value is 2(two=1) or not (two=0). By considering these bits we can Calculate the MB digits as Y_j^{MB}:

$Y_j^{MB} = (-1)^{S_j} [one_j + 2.two_j]$

The Add multiply operator is to design the adder unit in a single data bath block. The three techniques that are introduced to implement the unit. The three techniques are separated with even and odd number of inputs.

A) S-Mb 1 Technique:

This technique is used to implement two full adders with even and odd bits. The number of even bits two Fas is used as FA, the conventional bit used as output FA^*.

$FA^* = 2.c_o - S = p - q + c_i$

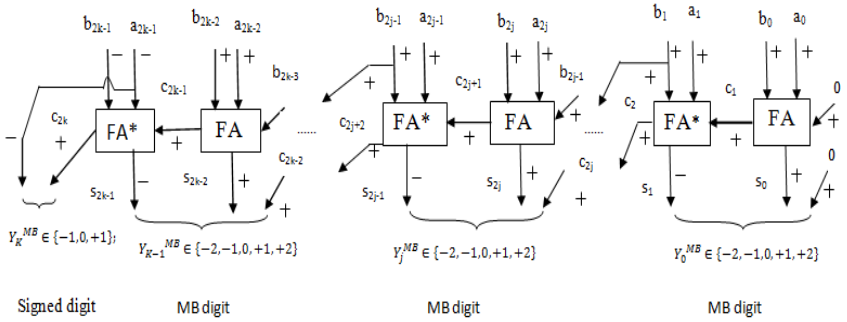


FIG 2: SMB1 Technique for Even Bit Width

For odd bit width the output value is FA^{**} $FA^{**} = -2.C_o + S = -p - q + c_i$

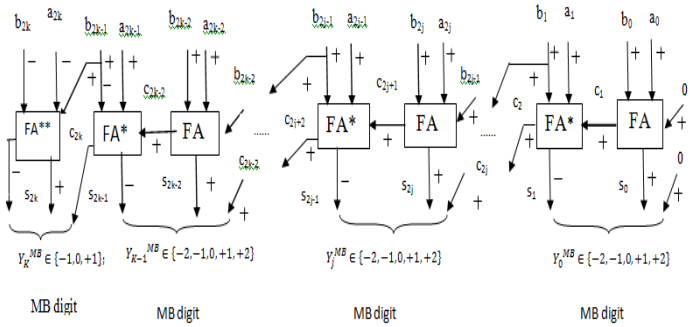


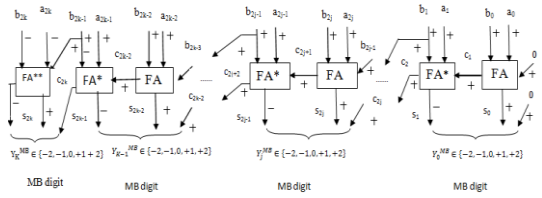
FIG 3: SMB 1 Technique for Odd Bit Width

A) SMB 2 Technique:

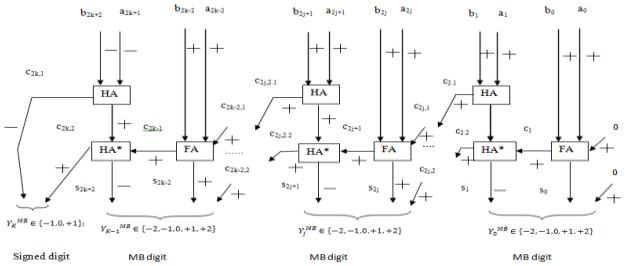
This technique is used for bit width and a conventional full adder is used with two half adders output value as HA^{*}.

HA^{*}= -2. C+S= -p-q

For odd bit width the full adders FA^{*} is used at the end.



(a)



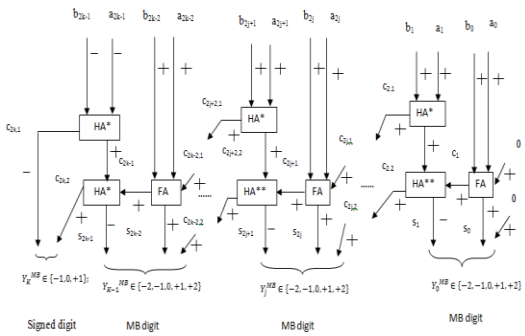
(b)

FIG 4: SMB 2 Technique for (a)Even (b)Odd Number of bits.

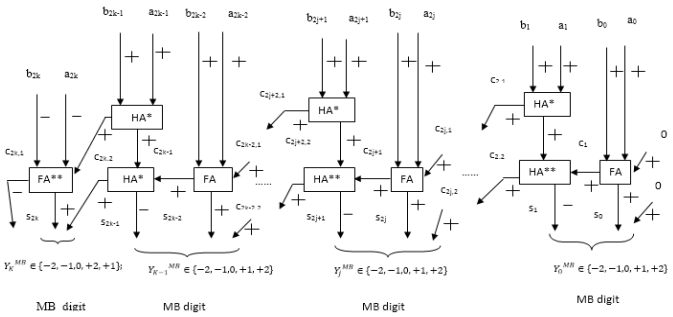
B) SMB 3 Technique:

In this technique even bit width input numbers and a conventional full adders with three half adders he half adders are conventional HA, HA^{*} and HA^{**}. The output value is

HA^{**}= 2.c – s= -p +q



(a)

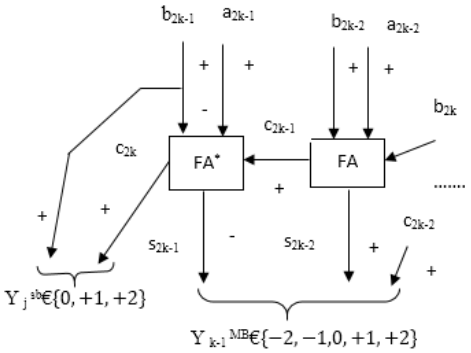


(b)

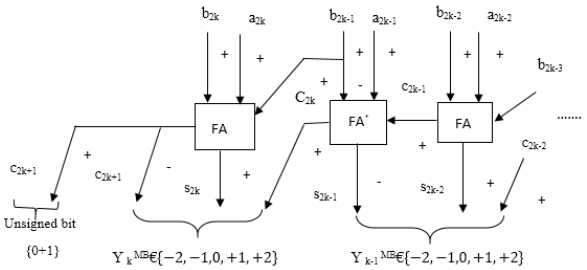
FIG 5: SMB 3 Technique for (a)Even (b)Odd Number of bits.

4)Unsigned Input Numbers:

In this the both inputs A and B are unsigned numbers with the most significant bits as signed numbers. This figure shows the even and odd bits with S-MB implementations.

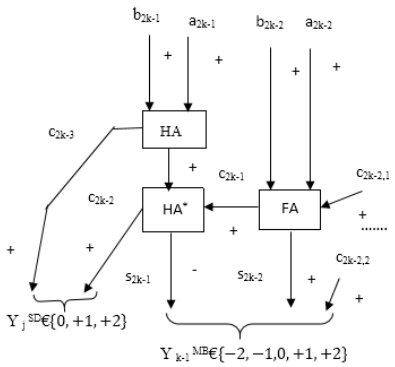


(a)

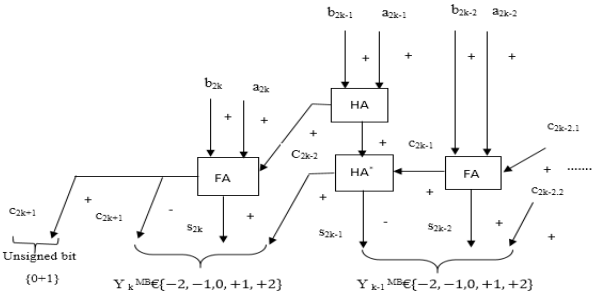


(b)

FIG 8 : Unsigned input numbers of SMB 1 (a)Even (b) Odd

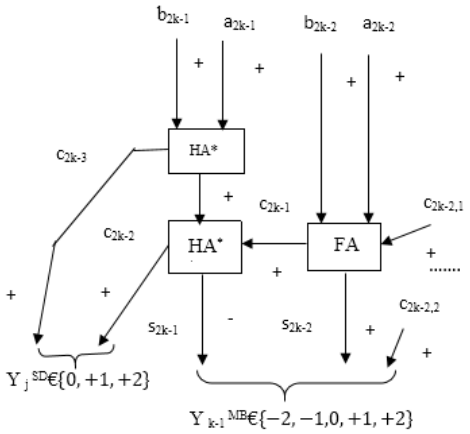


(a)

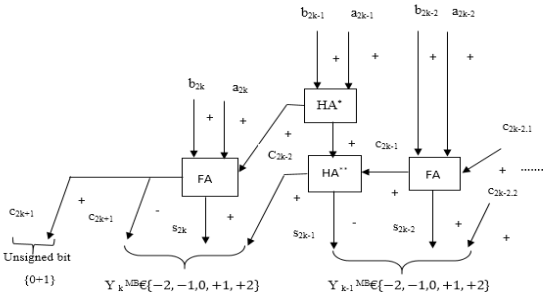


(b)

FIG 9: unsigned input numbers for SMB 2 (a) Even (b) Odd



(a)



(b)

FIG 10: unsigned input numbers for SMB 3 (a) Even (b) Odd

The Sum modified booth recoder output is the sum of two numbers in its direct recoding. The gate level circuit is encoding signals generated by $S_{j,}$ $One_{j,}$ Two_{j} in (fig 6). The partial products are generated by encoded signal with input X. This algorithm consists of half adders and full adders and compressors. By using this ripple carry addition the final multiplication are added in last two rows.

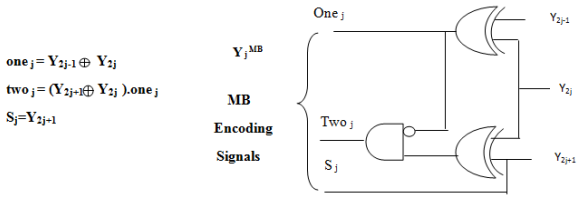


FIG 6: Gate Level Implementation for MB Encoding Signals

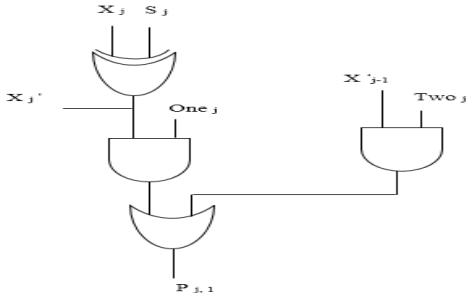
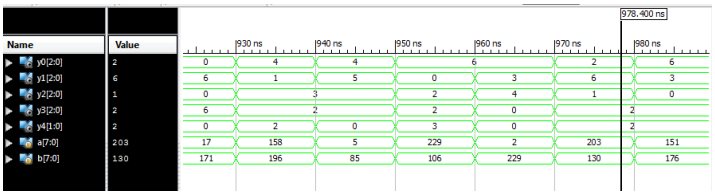
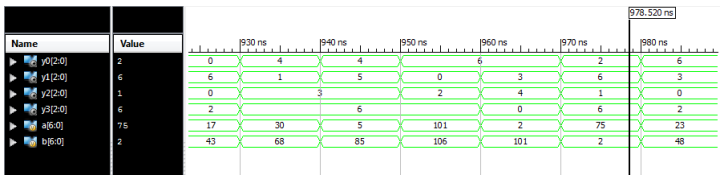


FIG 7: Booth Decoders

II. Simulation Results

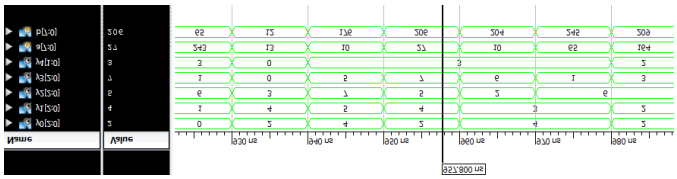


(a)

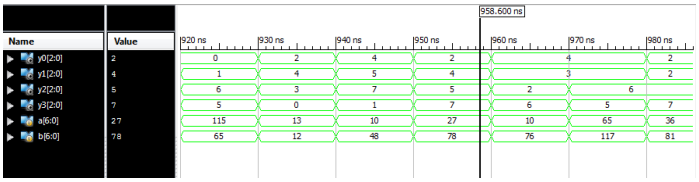


(b)

FIG 11: SMB 1 Technique for (a) Even (b) Odd

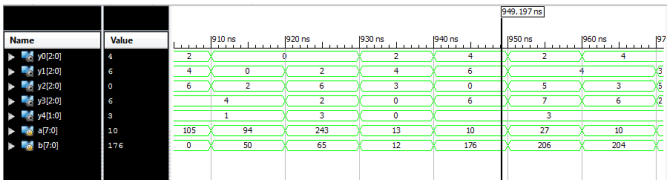


(a)

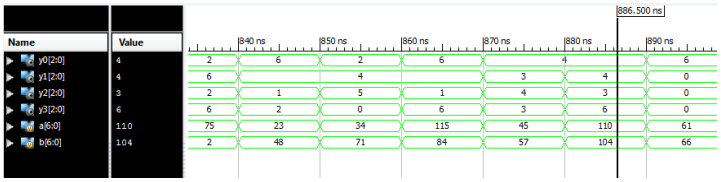


(b)

FIG 12: SMB 2 Technique for (a) Even (b) Odd

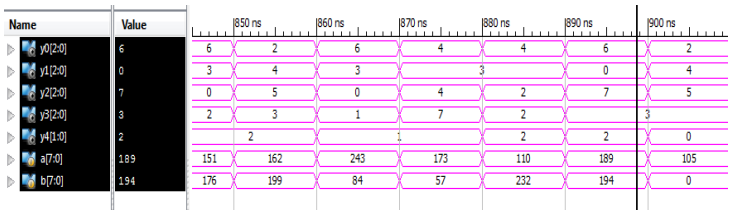


(a)

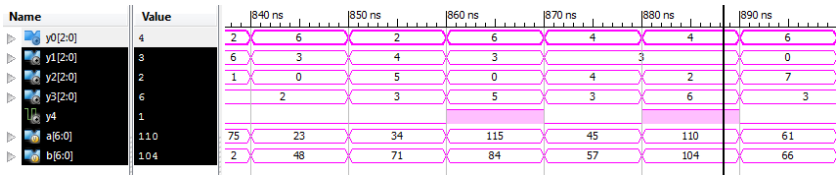


(b)

FIG 13: Simulation Results for SMB 3 Technique (a) Even (b) Odd

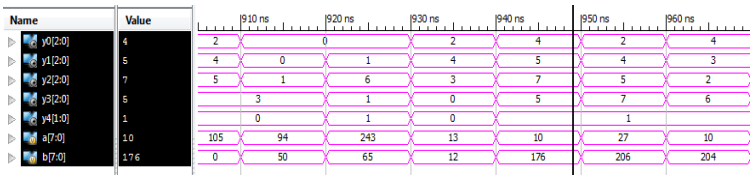


(a)

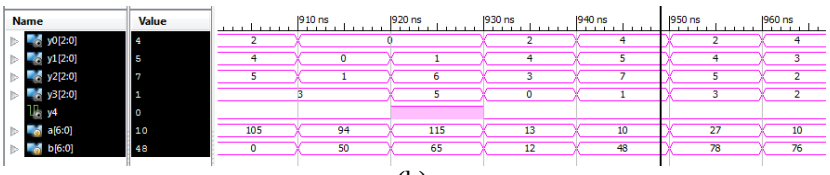


(b)

FIG 11: Simulation Results for unsigned input numbers SMB 1 (a) Even (b) Odd

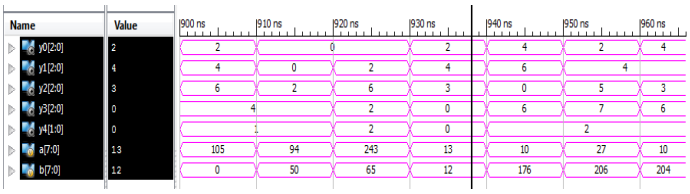


(a)

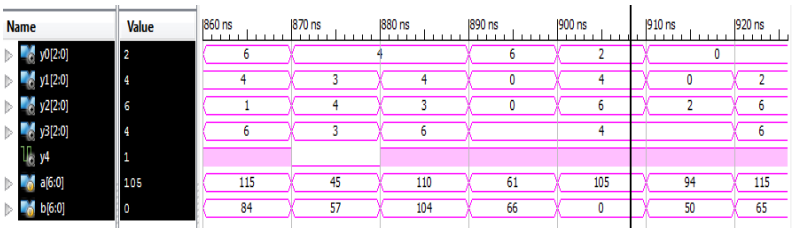


(b)

FIG 12: Simulation Results for unsigned input numbers SMB 2 (a) Even (b) Odd



(a)



(b)

FIG 13: Simulation Results for unsigned input numbers SMB 3 (a) Even (b) Odd

Table 2: Comparison of power

Techniques	Existing	Proposed
SMB 1 Even	0.139	0.82
SMB 1 Odd	0.112	0.88
SMB 2 Even	0.136	0.84
SMB 2 Odd	0.107	0.82
SMB 3 Even	0.108	0.83
SMB 3 Odd	0.872	0.83

III. Conclusion

This design is used to implement the sum of products in direct recoding and sum of two numbers in modified booth (MB) form. This paper tells about the lower frequencies with different timings in terms of area, power and time consumption. The proposed technique is done with reversible logic gates in which the area and power is reduced when compare to existing.

References

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

[2]. 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.

[3]. 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.

[4]. 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.

[5]. no. 2, pp. 201–208, Feb. 2010.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.

[6]. M. Daumas and D. W. Matula, "A Booth multiplier accepting both a redundant or a non redundant input with no additional delay," in Proc. IEEE Int. Conf. on Application-Specific Syst., Architectures, and Processors, 2000, pp. 205–214.