International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016 [ICSSCCET 2016]

ISBN	978-81-929866-6-1
Website	icssccet.org
Received	25 – February – 2016
Article ID	ICSSCCET078

VOL	02
eMail	icssccet@asdf.res.in
Accepted	10 - March – 2016
eAID	ICSSCCET.2016.078

A Novel Approach of an Efficient Booth Encoder for Signal Processing Applications

R R Thirrunavukkarasu[1] & R Satheeshkumar[2]
[1]Assistant Professor, Department of ECE, Karpagam Institute of Technology, Coimbatore
[2]Assistant Professor, Department of ECE, K.S.Rangasamy College of Technology, Tiruchengode

Abstract- *The conventional modified Booth encoding (MBE) generates an irregular partial product array because of the extra partial product bit at the least significant bit position of each partial product row. A simple approach is proposed in which a regular partial product array is generated with fewer partial product rows for reducing power dissipation. Modified Booth Encoding (MBE) circuit is used for the multiplier that generates the partial products. Booth encoder develops a control signal version that designates the partial product. A partial product generating circuit generates a partial product according to the control signal from the encoding circuit. In addition, the partial product array of multiplier is reduced by a Wallace tree scheme and a carry look-ahead adder is used for the final addition. The proposed design is implemented in MICROWIND3.1, DSCH3.1 EDA tools. The circuit has been fabricated using a 65-μm (6 metal with 0.7v, 2.5v) and 90-μm (6 metal with 1.2v, 2.5v) by using deep submicron CMOS technology. The implementation result shows that the power consumption of the proposed MBE scheme is better compared to conventional multiplier.*

Keywords— *Partial products, modified booth, multiplier.*

I. INTRODUCTION

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore, low-power multiplier design has been an important part in low-power VLSI system design. There has been extensive work on low-power multipliers at technology, physical, circuit and logic levels. However, it is difficult to consider application-specific data characteristics in low-level power optimization. The main research hypothesis of this work is that high-level optimization of multiplier designs produces more power-efficient solutions than optimization only at low levels. Specifically, we consider how to optimize the internal algorithm and architecture of multipliers and how to control active multiplier resource to match external data characteristics. The primary objective is power reduction with small area and delay overhead. By using new algorithms or architectures, it is even possible to achieve both power reduction and area/delay reduction, which is strength of high-level optimization. The tradeoff between power, area and delay is also considered in some cases.

Enhancing the processing performance and reducing the power dissipation of the systems are the most important design challenges for multimedia and digital signal processing (DSP) applications, in which multipliers frequently dominate the system's performance and power dissipation. Multiplication consists of three major steps: 1) recoding and generating partial products; 2) reducing the partial products by partial product reduction schemes (e.g., Wallace tree) to two rows; and 3) adding the remaining two rows of partial products by using a carry-propagate adder (e.g., carry look-ahead adder) to obtain the final product. In this brief, we will focus on the first step (i.e., partial product generation) to reduce the area, delay, and power consumption of multipliers.

First chapter gives the brief introduction about multiplier and their power efficient solutions. Second chapter describes and analyzes

Cite this article as: R R Thirrunavukkarasu, R Satheeshkumar. "A Novel Approach of an Efficient Booth Encoder for Signal Processing Applications". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 406-412. Print.

the previous works, mainly Booth encoding and partial product generator. Third chapter deals with the proposed system that overcomes the drawbacks in existing system. Fourth chapter shows the design of new MBE scheme and their result and simulation. Fifth chapter concludes by describing various observations and scope of future work.

II. High Speed Multipliers

For an 8-bit × 8-bit multiplication, a multiplier without MBE will generate eight partial product rows (because there is one partial product row for each bit of the multiplier). However, with MBE, only partial products rows are generated, as shown in the example of Fig. 1.

b_p	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PP_0						$\overline{s_0}$	s_0	s_0	p_{07}	p_{06}	p_{05}	p_{04}	p_{03}	p_{02}	p_{01}	p_{00}
PP_1					1	$\overline{s_1}$	p_{17}	p_{16}	p_{15}	p_{14}	p_{13}	p_{12}	p_{11}	p_{10}		neg_0
PP_2			1	$\overline{s_2}$	p_{27}	p_{26}	p_{25}	p_{24}	p_{23}	p_{22}	p_{21}	p_{20}		neg_1		
PP_3	1	$\overline{s_3}$	p_{37}	p_{36}	p_{35}	p_{34}	p_{33}	p_{32}	p_{31}	p_{30}		neg_2				
PP_4										neg_3						

Figure 1. MBE partial products arrays

However, there are actually partial product rows rather than n/2 because of the last neg signal (neg3 in Fig. 1). The neg signals (neg0, neg1, neg2, and neg3) are needed because MBE may generate a negative encoding ((-1) times the multiplicand or (-2) times the multiplicand).

Having one more partial product row adds at least one more EXOR-delay to the time to reduce the partial products. This fact that one additional partial product row brings delay is even more critical for multiplications of smaller words (8 ~ 16) than with longer operands because of the relatively higher delay effect that this additional row brings. There is also an extra hardware cost since one more carry saving adder stage hardware is necessary.

Removing the last neg signal would prevent the extra partial product row and, thus, save the time of one additional carry save adding stage and the hardware required for the additional carry save adding and also generate a more regularly shaped partial product array, making it a more efficient configuration for VLSI implementation.

Indeed, if MBE encoding generated signals for only +2 × , +1 × , or 0 × the multiplicand, the neg signals would not be necessary (thus, there would not be the additional overhead of adding the last neg). Notice that if somehow produce the two's complement of the multiplicand while the other partial products were produced, there would be no need for the last neg because this neg signal would have already been applied when generating the two's complement of the multiplicand. Therefore, we "only" need to find a faster method to calculate the two's complement of a binary number.

The conventional method (complement a binary number and add 1 to the complemented number) will not work for us because the propagation delay of the carry linearly increases with the word size and it would be much greater than the delay to generate the partial products. Therefore, we need a faster method. Our method is an extension of the well-known algorithm where all the bits after the rightmost "1" in the word are complemented but all the other bits are unchanged. The two's complement of a binary number 0010102 is 1101102 (Fig 2). For this number, the rightmost "1" happens in bit position 1 (the check mark position in Fig. 2). Therefore, values in bit positions 2 to 5 can simply be complemented, while values in bit positions 0 and 1 are kept unchanged. Therefore, two's complementation now comes down to finding the conversion signals that are used for selectively complementing some of the input bits. If the conversion signal at any position is "0" (the crosses in Fig 2), then the value is kept unchanged and, if the conversion signal is "1" (the checks in Fig.2), then the value is complemented. The conversion signals after the rightmost "1" are always 1. They are 0 otherwise. Once a lower order bit has been found to be a "1," the conversion signals for the higher order bits to the left of that bit position should all be "1."

However, this searching for the rightmost "1" could be as time consuming as rippling a carry through to the MSB since the previous bits information must be transferred to the MSB. Therefore, we must find a method to expedite this detection of the rightmost "1."

When grouping two 2n-bits groups, the leftmost conversion signals from the right group contain the accumulative information of its group about whether a "1" ever appeared in any bit position of its group so that a conversion signal should force all the conversion signals from the left group all the way to the "1" if it is itself a "1."

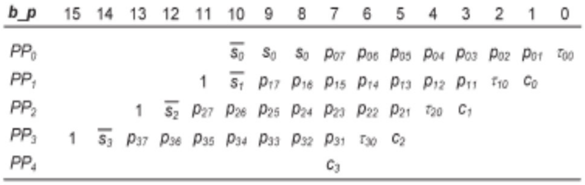
Bit Position	5	4	3	2	1	0	
Input Binary	0	0	1	0	1	0	
					✓		First 1"s Apprearance from LSB
	✓	✓	✓	✓	X	X	Complementation
Two's Complement	1	1	0	1	1	0	

Figure.2 Two's complement conversion example

Cite this article as: R R Thirrunavukkarasu, R Satheeshkumar. "A Novel Approach of an Efficient Booth Encoder for Signal Processing Applications". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 406-412. Print.

For instance, if CS1 (the leftmost conversion signal from the right group) = "1," the conversion signals from the left group (CS2 and CS3) should be forced to a "1," regardless of their previous values. If CS1 ="0," nothing happens to the conversion signals from the left group. Likewise, CS5 may affect conversion signals CS6 and CS7. The same goes for CS3', which may affect the conversion signals (CS7', CS6', CS05', and CS4').

By applying the method we just described for two's complementation, the last partial product row (as in Fig.1) is correctly replaced without the last neg (as in Fig. 3). Now, the multiplication can have a smaller critical path. This avoids having to include one extra carry saving adding stage. It also reduces the time to find the product and saves the hardware corresponding to the carry saving adding stage.



b_p	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PP_0						$\overline{s_0}$	s_0	s_0	p_{07}	p_{06}	p_{05}	p_{04}	p_{03}	p_{02}	p_{01}	τ_{00}
PP_1					1	$\overline{s_1}$	p_{17}	p_{16}	p_{15}	p_{14}	p_{13}	p_{12}	p_{11}	τ_{10}	c_0	
PP_2			1	$\overline{s_2}$	p_{27}	p_{26}	p_{25}	p_{24}	p_{23}	p_{22}	p_{21}	τ_{20}	c_1			
PP_3	1	$\overline{s_3}$	p_{37}	p_{36}	p_{35}	p_{34}	p_{33}	p_{32}	p_{31}	τ_{30}	c_2					
PP_4									c_3							

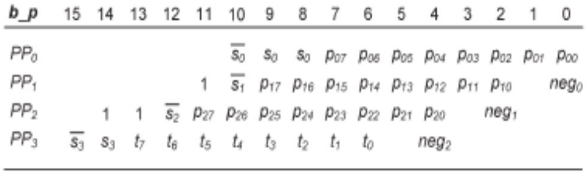
Figure.3. MBE partial products arrays after removing the last neg.

III. Modified Booth Multiplier

The proposed MBE multiplier combines the advantages of these two approaches to produce a very regular partial product array, as shown in Fig. 4.

A. MBE Recoding

In the partial product array, not only each negi is shifted left and replaced by ci but also the last neg bit is removed by using a simple approach described in detail in the following section.



b_p	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PP_0						$\overline{s_0}$	s_0	s_0	p_{07}	p_{06}	p_{05}	p_{04}	p_{03}	p_{02}	p_{01}	p_{00}
PP_1					1	$\overline{s_1}$	p_{17}	p_{16}	p_{15}	p_{14}	p_{13}	p_{12}	p_{11}	p_{10}		neg_0
PP_2		1	1	$\overline{s_2}$	p_{27}	p_{26}	p_{25}	p_{24}	p_{23}	p_{22}	p_{21}	p_{20}		neg_1		
PP_3	$\overline{s_3}$	s_3	t_7	t_6	t_5	t_4	t_3	t_2	t_1	t_0		neg_2				

Figure.4. Proposed MBE partial product array for 8 × 8 multiplication.

For MBE recoding, consider the multiplication of two n-bit integer numbers A (multiplicand) and B (multiplier) in 2's complement representation

According to the encoded results from B, the Booth selectors choose −2A, −A, 0, A, or 2A to generate the partial product rows, as shown in Table 1.

Table 1 MBE Table

b_{2i+1}	b_{2i}	b_{2i-1}	Operation	neg_i	two_i	one_i	p_{ij}
0	0	0	+0	0	0	0	0
0	0	1	+A	0	0	1	a_j
0	1	0	+A	0	0	1	a_j
0	1	1	+2A	0	1	0	a_{j-1}
1	0	0	–2A	1	1	0	$\overline{a_{j-1}}$
1	0	1	–A	1	0	1	$\overline{a_j}$
1	1	0	–A	1	0	1	$\overline{a_j}$
1	1	1	–0	0	0	0	0

The 2A in Table 1 is obtained by left shifting A one bit. Negation operation is achieved by complementing each bit of A (one's complement) and adding "1" to the least significant bit. Adding "1" is implemented as a correction bit neg, which implies that the partial product row is negative (neg= 1) or positive (neg= 0).

The Booth encoder and selector circuits are depicted in Fig.5(a) and (b), respectively. Booth encoder develops a control signal version that designates the partial product.

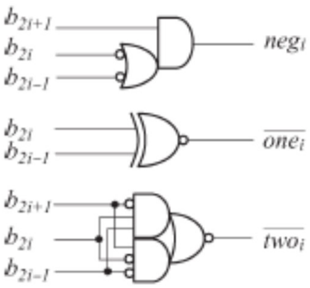


Figure.5 (a). MBE Encoder

A partial product generating circuit generates a partial product according to the control signal from the encoding circuit.

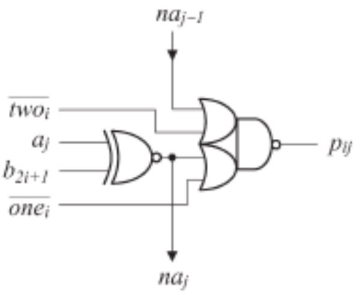


Figure.5 (b). MBE Selector

Multiplier recoding reduces the number of PPs, resulting in less area and power than in binary multiplication. After comparing common recoding schemes, we developed a version neg/two/one-nf ("nf" for neg-first) shown in Table 1. The negation operation is done before the selection between 1X and 2X so that two i and one i set PP i to zero regardless of negi for "-0." To generate an additional "1" for negative PP i, a correction bit c i = y 2i+1(y 2iy 2i-1)' is used. The delay of the recoding and PPG logic is roughly 2TXOR2. The signal paths are more balanced than in other schemes and glitches are, thus, reduced in subsequent PPR logic.

B. New MBE Scheme

In fig. 6, the encoder and the selector circuit receives 3-bit x inputs and n-bit y inputs, respectively.

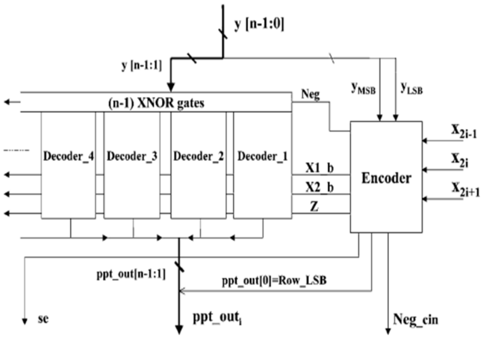


Figure 6. Structure of the New MBE scheme

Here, based on this MBE scheme, the partial products are generated for a 8 x 8 multiplication. An MBE encoder will be used in which the 3-bit multiplier inputs are given and thus the three outputs from this encoder act as control signals. The selector circuit is connected in which 8 decoders were used to obtain the generated partial products. Along with the control signal, the 8-bit multiplicand inputs will be given to those selector circuits. Thus the partial product generated from selector circuit will be noted and compared with various sub-micron cmos technologies. This regular array is generated by only slightly modifying the original partial product generation circuits and introducing almost no area and delay overhead.

C. Final Addition

Finally, the remaining two rows of partial products are added by using the efficient adder to obtain the final product.

A power efficient full adder is designed which offers improved performance when compared to the existing cmos based adders.

Fig 7 and Fig 8 shows the schematic of sum block and carry block of our power-efficient full adder where keeper circuit and pseudo-NMOS are added.

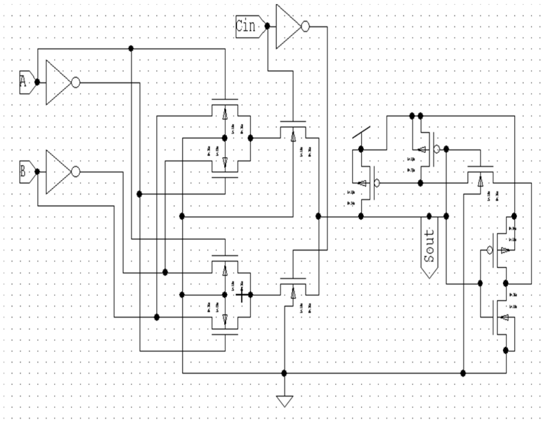


Figure.7 Schematic of sum block of power efficient full adder

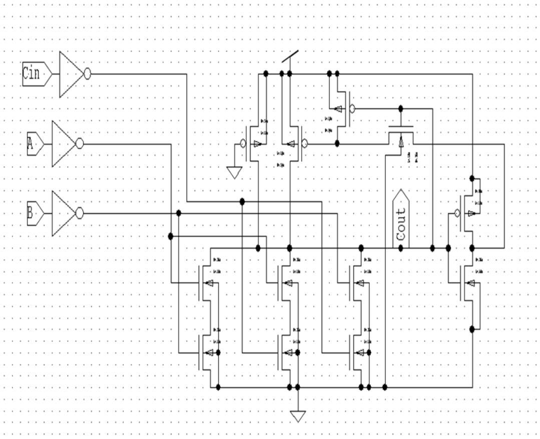


Figure.8 Schematic of carry block of power efficient full adder

IV. Results and Discussion

The complete front end has fully been integrated and fabricated using 120nm, 90nm, 65nm six-metal CMOS technology and thus the simulation results that were generated using MICROWIND 3.1 tool. The layout design of the modified booth multiplier circuit is obtained through the above tool and the performance of the circuit will be achieved based on its power, delay and their product parameters.

For comparison, we have implemented several MBE multipliers whose partial product arrays are generated by using different approaches. Except for a few of partial product bits that are generated by different schemes to regularize the partial product array, the other partial product bits are generated by using the similar method and circuits for all multipliers.

The performance parameters of the multiplier circuit taken for analysis are power, delay and power delay product (PDP) and those parameters are compared in various technologies such as 65nm, 90nm, 120nm and tabulated as follows.

Performance metrics of the New MBE scheme

Technology	Power(µW)	Delay(ps)	PDP(Joule)E-12
65nm	0.11985	410	0.049
90nm	0.13473	448	0.06
120nm	0.15001	682	0.102

Figure 9. Power analysis of MBE structure in various CMOS technologies.

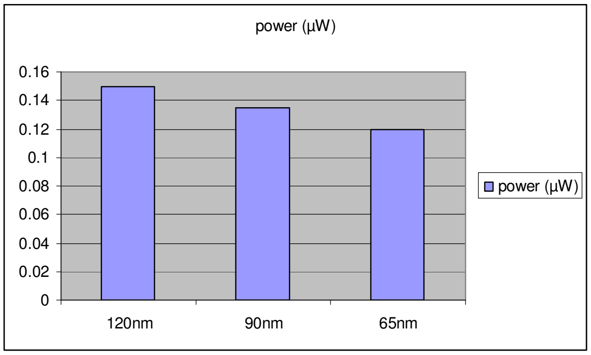


Figure 10. Delay analysis of MBE structure in various CMOS technologies.

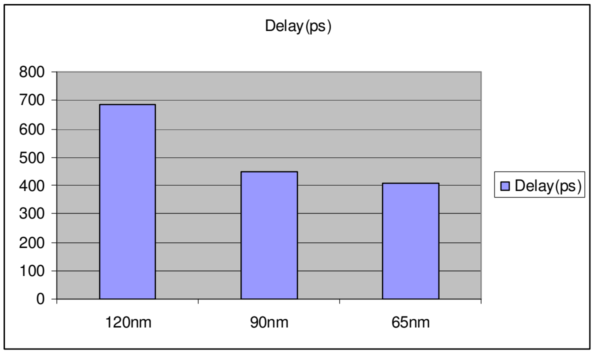


Figure 11. PDP analysis of MBE structure in various CMOS technologies

The circuit has been fabricated using a 65-nm(6 metal with 0.7v,2.5v) , 90-nm (6 metal with 1.2v,2.5) and 120-nm (6 metal with 1.2v,2.5) by using deep submicron CMOS technology.

V. Conclusions

The new MBE scheme is proposed to generate more partial product arrays with fewer partial product rows. Here, the partial product generation is completely focused on achieving low power consumption. More regular partial product array and fewer partial product rows provide a small and fast reduction tree, thus the area, delay and power of MBE multipliers, further reduced. Mainly based on the Booth encoder outputs, the proposed method generates partial products. From the simulation results, the power analysis and delay analysis of each circuit are made at various nanometer technologies. The result shows that proposed MBE scheme can achieve significant improvement in power consumption of the multiplier.

Cite this article as: R R Thirrunavukkarasu, R Satheeshkumar. “A Novel Approach of an Efficient Booth Encoder for Signal Processing Applications”. *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 406-412. Print.

References

1. Chang C.H, Gu J, and Zhang M, (Oct 2004) "Ultra low-voltage low-power CMOS 4–2 and 5–2 compressors for fast arithmetic circuits" IEEE Trans. Circuits Syst. I, Reg. Papers, Vol. 51, No. 10, pp. 1985–1997.
2. Cho K and Song M, (Jan 2001) "Design of a high performance 32 × 32-bit multiplier, with a novel sign select Booth encoder" in Proc. IEEE Int. Symp. Circuits Syst., Vol. 2, pp. 701–704.
3. Elguibaly F, (Sep 2000) "A fast parallel multiplier-accumulator using the modified Booth algorithm" IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process, Vol. 47, No. 9, pp. 902–908.
4. Huang Z and Ercegova M, (Mar 2005) "High-performance low-power left-to-right, array multiplier design" IEEE Trans. Comput., Vol. 54, No.3, pp. 272–283.
5. Kang J.Y and Gaudiot J.L, (Oct 2006) "A simple high-speed multiplier design" IEEE Trans. Comput., Vol. 55, No. 10, pp. 1253–1258.
6. Kim Y.E, Yoon J.O, Cho K.J, Chung J.G, Cho S.I and Choi S.S,(Jun 2006) "Efficient design of modified Booth multipliers for predetermined coefficients" in Proc. IEEE Int. Symp. Circuits Syst., pp. 2717–2720.
7. Lang T and Ercegovac.M.D, Digital Arithmetic. Morgan Kaufmann Publishers, Elsevier Science Ltd., 2004.
8. Salomon O, Green J.-M, and Klar H,(Jul 1995) "General algorithms for a simplified addition of 2's complement numbers," IEEE J. Solid-State Circuits, Vol. 30, No. 7, pp. 839–844.
9. Shiann-rong kuang, Jiun-ping wang, and Cang-yuan guo,(May 2009) "Modified booth multipliers with a regular partial product array," IEEE Trans. on Circuits and Systems, Vol. 56,No. 5.
10. Swartzlander E. E. and Angel E.de, Jr., "Low power parallel multipliers,"in Workshop VLSI Signal process IX, 1996, pp. 199–208.
11. Swartzlander E.E Jr., Bickerstaff K.C, and Schulte M.J.,(2001) "Analysis of column compression multipliers," Proc. 15th IEEE Symp. Computer Arithmetic, pp. 33-29.
12. Villeger.D, Oklobdzija V.G, and Liu S.S,(Mar 1996) "A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach," IEEE Trans Computers, Vol. 45, No. 3, pp. 294-306.
13. Wallace.C.S, (Feb-1964) "A suggestion for parallel multipliers,," IEEE Trans. Electron. Comput., Vol. EC-13, No. 1, pp. 14–17.
14. CIC Referenced Flow for Cell-based IC Design, 2008, Taiwan: Chip Implementation Center, CIC. Document no. CIC-DSD-RD-08-01.

Cite this article as: R R Thirrunavukkarasu, R Satheeshkumar. "A Novel Approach of an Efficient Booth Encoder for Signal Processing Applications". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 406-412. Print.