

Design of 16-bit Multiplier Using Efficient Recoding Techniques

K. N. Narendra Swamy¹ and J. Venkata Suman²

¹PG Student, Dept. of ECE, GMR Institute of Technology

²Assistant Professor, Dept. of ECE, GMRIT, RAJAM
nariswami27@gmail.com, venkatasuman.j@gmrit.org

Abstract

Multiplier is the major component for processing of large amount of data in DSP applications. Using different recoding schemes in Fused Add-Multiply (FAM) design for the reduction of power and look up tables. The performance of 16-bit signed and unsigned multipliers were designed and obtained results are tabulated using Efficient Modified Booth Recoding (EMBR) techniques, which can be used for low power applications.

Keywords: Fused Add Multiply, Signed numbers, unsigned numbers, Modified Booth multiplier, Efficient Modified Booth Recoding Techniques

1. Introduction

In good olden days, multiplication was usually executed via a series of addition, subtraction, and shift operations. Presently, electronic applications make ample use of Digital Signal Processing, based on huge number of arithmetic operations. Multiplication is one of them, reflected as repetitive additions. The number to be added is the multiplicand, the number of times added is the multiplier, and the end result is the product. Multiplication comprises of two basic operations - creation of partial products and accumulation. Process can be in two ways: reducing number of partial products and/or accelerating accumulation. System's presentation primarily depends on multiplier performance. The final Carry Look Ahead (CLA) [8] adder and The Carry Save Adder (CSA) tree used for speediness operation.

2. Motivation

Implementations of shift and add multipliers can elevate performance over software synthesis, but are still slow. The reason is that as each supplementary partial-product is summed a carry must be propagated from the least significant bit (LSB) to the most significant bit (MSB). This carry propagation is overwhelming, and must be repeated for each partial product to be summed. Such a technique was first proposed by Booth [6]. The original Booth's algorithm [1] [6] closes strings of 1's by using the property that: $2 + 2(n-1) + 2(n-2) + \dots + 2m = 2(n+1) - 2(n-m)$. Although Booth's algorithm yields at most $N/2$ encoded partial products from an N bit operand, the number of partial products produced differs. As a result, modified versions of Booth's algorithm for hardware multipliers are used. Modified Booth Encoding, halves the number of partial products to be summed.

3. Modified Booth Multiplier

Modified Booth (MB) [6, 7, 11] is used in multiplication. This generates at most $n/2+1$ partial product; it is a redundant signed-digit radix-4 encoding technique. It decreases the partial products to half in multiplication relating to any other radix-2 version.

3.1. Algorithm

1. Pad the LSB with one zero.
2. Pad the MSB with 2 zeros if n is even and 1 zero if n is odd.
3. Divide the multiplier into overlapping groups of 3-bits.
4. Determine partial product scale factor from modified booth to encoding table.
5. Compute the Multiplicand Multiples.
6. Summing Partial Products.

Encoding can be done by grouping of three bits at a time and must be able to add multiplicand times -2 , -1 , 0 , 1 and 2 . Since Booth recoding [6] got rid of 3's, generating partial products is not that rigid (shifting and negating). After the grouping of partial products, they are added, weighted appropriately, through a Carry-Save Adder (CSA) tree. Carry - Save.

Further, carry look ahead adder, a carry signal will be generated in two cases: (1) when both bits a and bare 1, or (2) when one of the two bits is 1 and the carry-in is 1. The Carry Look Ahead adder (CLA) [8] resolves the carry delay problem by computing the carry signals in advance, centered on the input signals. This addition reduces all partial-products down to a carry-save number by summing them up in an adder tree.

Table 1. Grouping Table

y_{2j+1}	y_{2j}	y_{2j-1}	y_j^{MB}	s_j	one_j	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

$$one_j = y_{2j-1} \oplus y_{2j} \quad two_j = (y_{2j+1} \oplus y_{2j}) \cdot \overline{one_j} \quad s_j = y_{2j+1}$$

4. Summation to Efficient Modified Booth Recoding Technique (S-MB)

4.1. Structured Signed Arithmetic

Increasing multiplier piece is by means of encoding techniques to reduce the number of partial products to be summed. Conventional and signed HAs and FAs [1, 10] are to be used, each of the three schemes can be easily applied in either signed (2's complement representation) or un-signed numbers which consist of odd or even number of bits. In the following techniques, both inputs are in 2's complement form for signed and unsigned consist of bits $2k$ in case of even or $2k+1$ bits in case of odd bit-width [9] considered. In these S-MB recoding techniques, recoding 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}) converted into one MB digit Y_j^{MB} . More precisely, two types of signed HAs used which are referred as HA* and HA**.

$$Y = A + B = y_k \cdot 2^{2k} + \sum_{j=0}^{k-1} y_j^{MB} \cdot 2^{2j} \quad (1)$$

$$\text{Where } y_j^{MB} = -2 s_{2j+1} + s_{2j} + c_{2j}$$

4.2. S-MB Recoding Techniques

4.2.1. Signed Input Numbers

If the input numbers A and B are signed, their MSB is negatively (filled by 1) signed. Below represents are S-MB schemes for even bit-width of A and B. The basic recoding block in all schemes remains unchanged.

1. S-MB Scheme 1

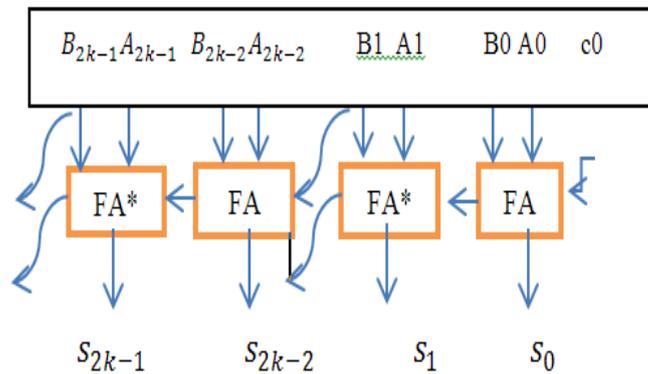


Figure 1. Signed S-MB1 Even Bits

2. S-MB Scheme 2

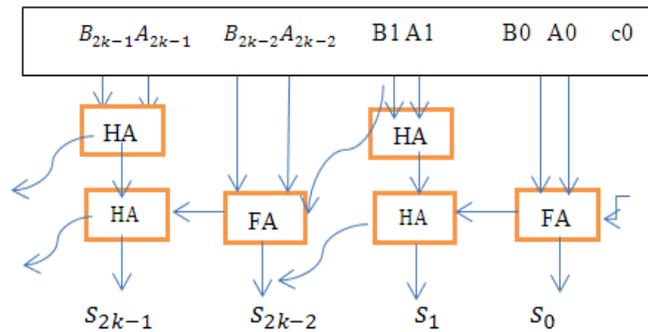


Figure 2. Signed S-MB2 Even Bits

3. S-MB Scheme 3

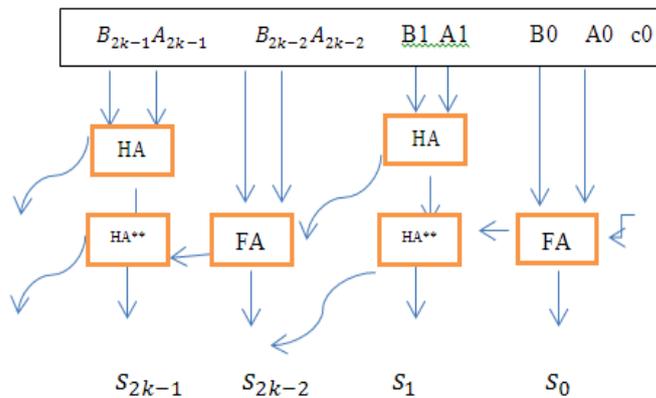


Figure 3. Signed S-MB3 Even Bit

4.2.2. Unsigned Input Numbers

If the input numbers A and B are signed, their MSB is positively (filled by 0) signed. Below represents are S-MB schemes for even bit-width of A and B. The basic recoding block in all schemes remains unchanged.

1. S-MB Scheme 1

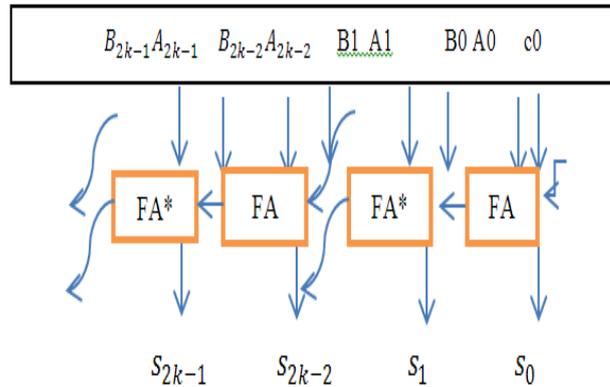


Figure 4. Unsigned S-MB1 Even Bit

2. S-MB Scheme 2

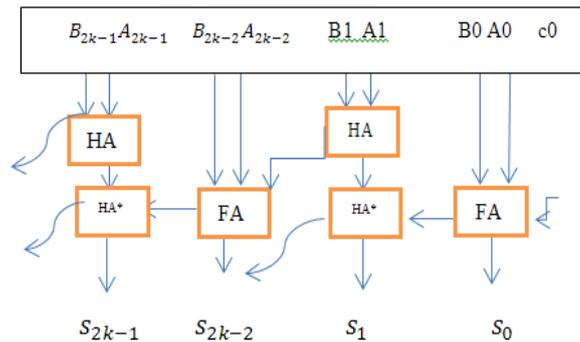


Figure 5. Unsigned S-MB2 Even Bit

3. S-MB Scheme 3

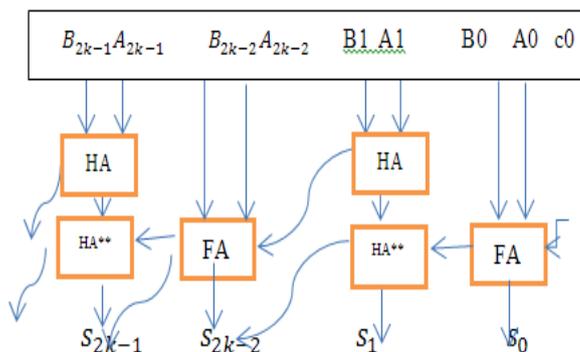


Figure 6. Unsigned S-MB3 Even Bit

5. Fused Add Multiply Implementation

The proposed Fused Add Multiply design [4] represented in Figure 7, the multiplier is a parallel one depends on the MB algorithm. Let us consider the product X. Y. The term

$Y = (y_{n-1}y_{n-2} \dots y_0)$ 2's is encoded based on the MB algorithm [1, 10] and multiplied with $X = (x_{n-1}x_{n-2} \dots x_0)$ 2's. Both X and Y consist of $n=2k$ bits and are in 2's complement type. Equation (2) describes the generation of k partial products:

$$PP_j = X \cdot y_j^{MB} = \bar{p}_j \cdot n2 + \sum_{i=0}^{n-1} p_{j,i} \cdot 2^i \quad (2)$$

The generation of the i- th bit $p_{j,i}$ of the partial product PP_j is given by next expression (3)

$$p_{j,i} = ((x_i \oplus s_i)one_j) + ((x_{i-1} \oplus s_j) + two_j) \quad (3)$$

After generation of partial products, they are added, suitably weighted, through a Wallace Carry-Save Adder (CSA) tree along with the Correction Term (CT) which is specified by the following equation (4):

$$Z = X \cdot Y = CT + \sum_{j=0}^{k-1} PP_j \cdot 2^{2j} \quad (4)$$

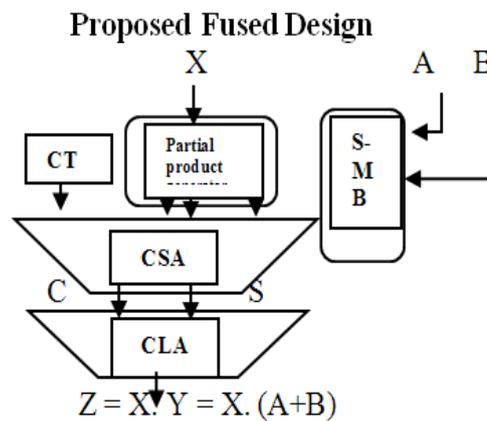


Figure 7. Proposed Fused Design

6. Results

An Efficient Modified Booth Recoding techniques based 16-bit signed and unsigned multipliers were designed in FAM model and simulated using Xilinx tool on Spartan 3E having device specifications as xc3s100E-4tq144 and power 33.61 mW. The performance comparison of 16-bit signed and unsigned multiplier parameters are shown in Tables 2 & 3.

6.1 Results of 16-bit Signed Multiplier

Table 2. 16 Bit Signed Bit Width

16 bit	SMB-1	SMB-2	SMB-3
POWER (mW)	11.95	12.32	11.53
No of LUT's	682	703	658
Delay (ns)	29.535	29.549	28.256
Memory (kb)	146140	147164	148188

According to Table 2, it is clearly observed that SMB-3 is proficient in case of power and number of elements (LUT's) than the other two schemes. The RTL schematic and Simulation results of 16-bit Signed multiplier are shown in Figure 8 and Figure 9.

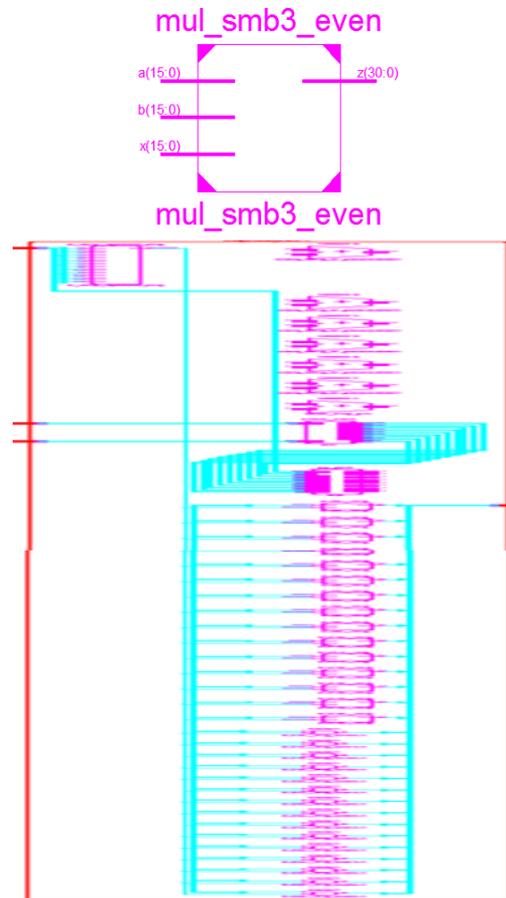


Figure 8. RTL Schematic for 16-bit Multiplier using SMB-3

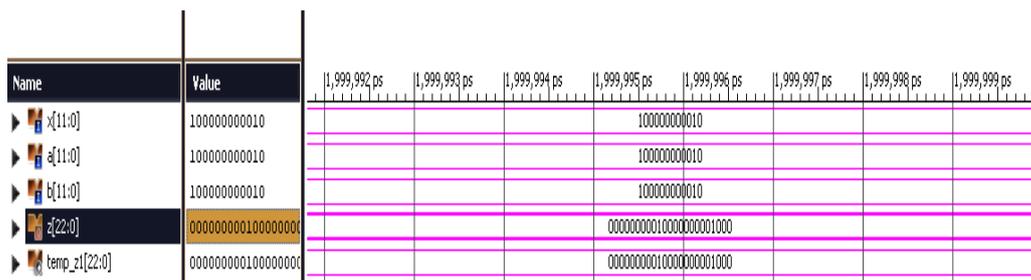


Figure 9. Simulation Results for 16-bit Multiplier using SMB-3

6.2 Results of 16-bit Unsigned Multiplier

Table 3. 16 bit Unsigned Bit Width

16 bit	SMB-1	SMB-2	SMB-3
POWER (mW)	9.747	9.799	9.90
No of LUT's	556	559	565
Delay (ns)	27.737	27.045	26.853
Memory (kb)	146140	147164	148188

According to Table 3, it is clearly observed that SMB-1 is proficient in case of power and number of elements (LUT's) than the other two schemes. The RTL schematic and Simulation results of 16-bit Signed multiplier are shown in Figure 10 and Figure 11.

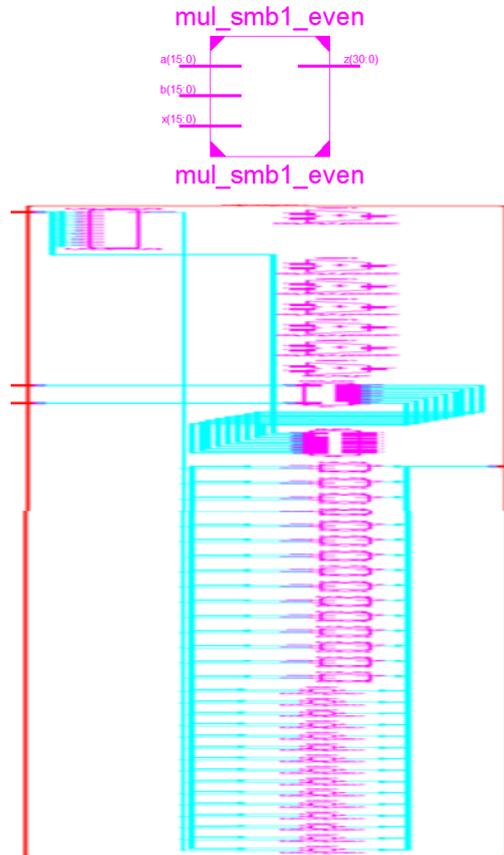


Figure 10. RTL Schematic for 16-bit Multiplier using SMB-1

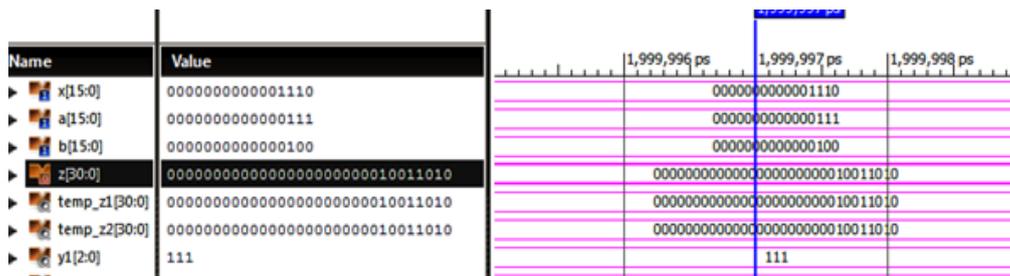


Figure 11. Simulation Results for 16-bit Multiplier using SMB-1

7. Conclusion

In this paper, design of 16-bit signed and unsigned multipliers based on EMBR techniques in fused add-multiply were explored. The proposed designs show major improvement in terms of power consumption, number of elements and delay. From the simulation results, we are clearly concludes that SMB3 will gives the better performance for 16-bit signed multiplier and SMB-1 will gives better performance in design of 16-bit unsigned multiplier in terms of power consumption, number of elements and delay. In future, the proposed EMBR based both signed and unsigned multiplier designs can be improved for more number of even and odd bit widths.

References

- [1] K. Taoumanis, S. Xydis, C. Efstathiou, N. Moschopoulos and K. Pekmestzi, "An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator", IEEE Trans. Computers and Systems-I Regular Papers, vol. 61, no. 4, (2014) April.
- [2] Y. Liao and D. B. Roberts, "A High-Performance and Low-Power 32-bit Multiply-Accumulate Unit with Single-Instruction-Multiple-Data (SIMD) Feature", IEEE Journal of Solid-State Circuits, vol. 37, no. 7, (2002) July.
- [3] T. T. Hoang and P. Larsson-Edefors, "A High-Speed, Energy-Efficient Two-Cycle Multiply-Accumulate (MAC) Architecture and Its Application to a Double-Throughput MAC Unit", IEEE Transactions On Circuits And Systems—I: Regular Papers, December, vol. 57, no. 12, (2010) December.
- [4] A. Amaricai, M. Vladutiu and O. Boncalo, "Design Issues and Implementations for Floating-Point Divide-Add Fused", IEEE Transactions On Circuits And Systems—II: Express Briefs, vol. 57, no. 4, (2010) April.
- [5] J. Mori, *et al.*, "A 10 ns 54 54b parallel structured full array multiplier with 0.5 m units cmos technology", IEEE Journal of Solid-State Circuits, vol. 26, no. 4, (1991), pp. 600-605.
- [6] D. Chandel, G. Kumawat, P. Lahoty, V. V. Chandrodya and S. Sharma, "Booth Multiplier: Ease of multiplication", International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, ISO 9001:2008 Certified Journal, vol. 3, Issue 3, (2014) March.
- [7] S. K. Suman and M. S. Manna, "Implementation of Modified Booth Algorithm (Radix 4) and its Comparison with Booth Algorithm (Radix-2)", Advance in Electronic and Electric Engineering, vol. 3, no. 6, (2013), pp. 683-690.
- [8] R. Uma, V. Vijayan, M. Mohanapriya and S. Paul, "Area, Delay and Power Comparison of Adder Topologies", International Journal of VLSI design & Communication Systems (VLSICS), vol. 3, no. 1, DOI: 10.5121/vlsic.2012.3113 153.
- [9] K. N. N. Swamy and V. Suman, "Design of Efficient and Fast Multiplier using MB recoding Techniques", International Journal of Emerging Research in Management and Technology, vol. 4, Issue 6, (2015) June.
- [10] R. Zimmermann and D. Q. Tran, "Design Ware, Solutions Group, Synopsys, Inc. 2025 NW Cornelius Pass Rd., Hillsboro, OR 97124, 'Optimized Synthesis of Sum-of-Products", Proceedings 37th Asilomar Conference on Signals, Systems, and Computers, IEEE, (2003) November.
- [11] M. Chaudhary and M. S. Narula, "High Speed Modified Booth's Multiplier for Signed and Unsigned Number", ITM University, Gurgaon, Haryana, (Dept. of EECE), ITM University, GurIn thisgaon, Haryana.

Authors



K. N. Narendra Swamy, pursuing M. Tech (VLSI & ES) in GMRIT, Rajam, AP, India and Received B.Tech (ECE) from Avanathi Institute of Engineering and Technology under JNTUK. Major areas of interest include VLSI, DSP, ES.



Jami Venkata Suman, has completed Bachelor's degree from Tontadarya College of Engineering, under VTU, Belgaum, Karnataka. He Received Master's degree in VLSI System Design from Annamacharya Institute of Technology and Sciences, Rajampet, under JNTU, Hyderabad, A.P and Master of Business Administration in HRM and MRKT from A.U, Visakhapatnam, A.P. He is currently working as an Assistant Professor in the Department of Electronics and Communication Engineering at GMR Institute of Technology, Rajam. Major Research areas include Radar Signal Processing and Low Power VLSI. He is Life Member of ISTE and IACSIT Journal.