

An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator

Swathi. B

M.Tech Student, Electronics and Communication Engineering (E.C.E.), J.N.T.University Anantapur,
Andhra Pradesh, India.

E-mail: swathi_bnm@yahoo.com

ABSTRACT

Complex arithmetic operations are widely used in Digital Signal Processing (DSP) applications. In this work, we focus on optimizing the design of the fused Add-Multiply (FAM) operator for increasing performance. We investigate techniques to implement the direct recoding of the sum of two numbers in its Modified Booth (MB) form. We introduce a structured and efficient recoding technique and explore three different schemes by incorporating them in FAM designs. Comparing them with the FAM designs which use existing recoding schemes, the proposed technique yields considerable reductions in terms of critical delay, hardware complexity and power consumption of the FAM unit.

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

I. INTRODUCTION

Modern consumer electronics make extensive use of Digital Signal Processing (DSP) providing custom accelerators for the domains of multimedia, communications etc. Typical DSP applications carry out a large number of arithmetic operations as their implementation is based on computationally intensive kernels, such as Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Finite Impulse Response (FIR) filters and signals' convolution. As expected, the performance of DSP systems is inherently affected by decisions on their design regarding the allocation and the architecture of arithmetic units. Recent research activities in the field of arithmetic optimization [1], [2] have shown that the design of arithmetic components combining operations which share data, can lead to significant performance improvements. Based on the observation that an addition can often be subsequent to a multiplication (e.g., in symmetric FIR filters), the Multiply-Accumulator (MAC) and Multiply-Add

(MAD) units were introduced [3] leading to more efficient implementations of DSP algorithms compared to the conventional ones, which use only primitive resources [4]. Several architectures have been proposed to optimize the performance of the MAC operation in terms of area occupation, critical path delay or power consumption [5]–[7]. As noted in [8], MAC components increase the flexibility of DSP data path synthesis as a large set of arithmetic operations can be efficiently mapped onto them. Except the MAC/MAD operations, many DSP applications are based on Add-Multiply (AM) operations (e.g., FFT algorithm [9]). The straightforward design of the AM unit, by first allocating an adder and then driving its output to the input of a multiplier, increases significantly both the critical path delay of the circuit. Targeting an optimized design of AM operators, fusion techniques [10]–[13], [23] are employed based on the direct recoding of the sum of two numbers (equivalently a number in carry-save representation [14]) in its Modified Booth (MB) form [15]. Thus, the carry-

propagate (or carry-look-ahead) adder [16] of the conventional AM design is eliminated resulting in considerable gains of performance. Lyu and Matula [10] presented a signed-bit MB recoder which transforms redundant binary inputs to their MB recoding form. A special expansion of the preprocessing step of the recoder is needed in order to handle operands in carry-save representation. In [12], the author proposes a two-stage recoder which converts a number in carry-save form to its MB representation. The first stage transforms the carry-save form of the input number into signed-digit form which is then recoded in the second stage so that it matches the form that the MB digits request. Recently, the technique of [12] has been used for the design of high performance flexible coprocessor architectures targeting the computationally intensive DSP applications [17]. Zimmermann and Tran [13] present an optimized design of [10] which results in improvements in both area and critical path. In [23], the authors propose the recoding of a redundant input from its carry-save form to the corresponding borrow-save form keeping the critical path of the multiplication operation fixed. Although the direct recoding of the sum of two numbers in its MB form leads to a more efficient implementation of the fused Add-Multiply (FAM) unit compared to the conventional one, existing recoding schemes are based on complex manipulations in bit-level, which are implemented by dedicated circuits in gate-level. This work focuses on the efficient design of FAM operators, targeting the optimization of the recoding scheme for direct shaping of the MB form of the sum of two numbers (Sum to MB – *S-MB*). More specifically, we propose a new recoding technique which decreases the critical path delay and reduces area and power consumption. The proposed *S-MB* algorithm is structured, simple and can be easily modified in order to be applied either in signed (in 2’s complement representation) or unsigned numbers, which comprise of odd or even number of bits. We explore three alternative schemes of the proposed *S-MB* approach

using conventional and signed-bit Full Adders (FAs) and Half Adders (HAs) as building blocks.

II. FUSED AM IMPLEMENTATION

A. Review of the Modified Booth Form

Modified Booth (MB) is a prevalent form used in multiplication [15], [20], [24]. It is a redundant signed-digit radix-4 encoding technique. Its main advantage is that it reduces by half the number of partial products in multiplication comparing to any other radix-2 representation. Let us consider the multiplication of 2’s complement numbers X and Y with each number consisting of $n = 2k$. The multiplicand can be represented in MB form as:

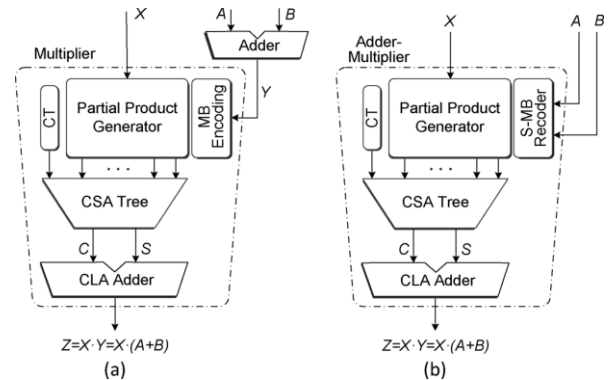


Figure 1. AM operator based on the (a) conventional design and (b) fused design with direct recoding of the sum of operand in its MB representation

The multiplier is a basic parallel multiplier based on the MB algorithm. The terms CT, CSA Tree and CLA Adder are referred to the Correction Term, the Carry-Save Adder Tree and the final Carry-Look-Ahead Adder of the multiplier

$$Y = \langle y_{n-1}y_{n-2} \dots y_1y_0 \rangle_{2^n} = -y_{2k-1} \cdot 2^{2k-1} + \sum_{i=0}^{2k-2} y_i \cdot 2^i$$

$$= \langle y_{k-1}^{MB} y_{k-2}^{MB} \dots y_1^{MB} y_0^{MB} \rangle_{MB} = \sum_{j=0}^{k-1} y_j^{MB} \cdot 2^{2j} \quad (1)$$

$$y_j^{MB} = -2y_{2j+1} + y_{2j} + y_{2j-1} \quad (2)$$

$$y_j^{MB} = (-1)^{s_j} \cdot [one_j + 2 \cdot two_j]$$

B. FAM Implementation;

In the FAM design presented in Figure 1(b), the multiplier is a parallel one based on the MB algorithm. Let us consider the product .The term is encoded based on the MB algorithm (Section II.B) and multiplied with .Both and consist of $n = 2k$ and are in 2's complement form. Equation (4) describes the generation of the partial products:

$$PP_j = X \cdot y_j^{MB} = \bar{p}_{j,n} 2^n + \sum_{i=0}^{n-1} p_{j,i} \cdot 2^i.$$

$$p_{j,i} = ((x_i \oplus s_j) \wedge one_j) \vee ((x_{i-1} \oplus s_j) \wedge two_j).$$

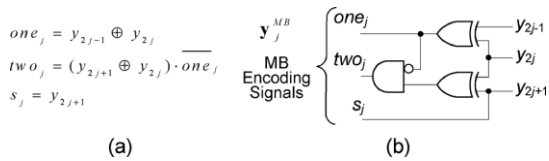


Figure 2. (a) Boolean equations and (b) gate-level schematic for the implementation of the MB encoding signals

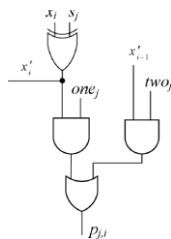


Figure 3. Generation of the i -th bit of the partial product for the conventional MB multiplier

For the computation of the least and the most significant bits of the partial product we consider and respectively. Note that in case that , the number of the resulting partial products is and the most significant MB digit is formed based on sign extension of the initial 2's complement number. After the partial products are generated, they are added, properly weighted, through a Wallace Carry-Save Adder (CSA) tree [21] along with the Correction Term (CT) which is given by the following equations:

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

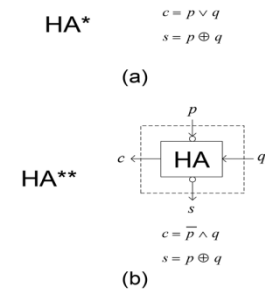


Figure 4. Boolean equations and schematics for signed (a) HA* and (b) HA**

III. SUM TO MODIFIED BOOTH RECODING TECHNIQUE (S-MB)

A. Defining Signed-Bit Full Adders and Half Adders for Structured Signed Arithmetic

In *S-MB* recoding technique, we recode the sum of two consecutive bits of the input () with two consecutive bits of the input () into one MB digit. As we observe from (2), 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. Consequently, in order to transform the two aforementioned pairs of bits in MB form we need to use signed-bit arithmetic. For this purpose, we develop a set of bit-level signed Half Adders (HA) and Full Adders (FA) considering their inputs and outputs to be signed. More specifically, in this work, we use two types of signed HAs which are referred as HA* and HA**.

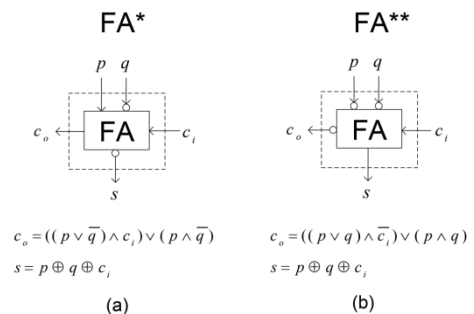


Figure 5. Boolean equations and schematics for signed (a) FA* and (b) FA**.

B. Proposed S-MB Recoding Techniques

We use both conventional and signed HAs and FAs of Section III.A in order to design and explore three new alternative schemes of the *S-MB* recoding technique. Each of the three schemes can be easily applied in either signed (2's complement representation) or unsigned numbers which consist of odd or even number of bits. In all schemes we consider that both inputs and are in 2's complement form and consist of bits in case of even or bits in case of odd bit-width. Targeting to transform the sum of and in its MB representation, we consider the bits , and , as the inputs of the recoding cell in order to get at its output the three bits that we need to form the MB digit y according to (2).

1) S-MB1 Recoding Scheme:The first scheme of the proposed recoding technique is referred as *S-MB1* and is illustrated in detail in Figure 6 for both even (Figure 6(a)) and odd (Figure 6(b)) bit-width of input numbers. As can be seen in Figure 6, the sum of and is given by the next relation:

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

where $y_j^{MB} = -2s_{2j+1} + s_{2j} + c_{2j}$.

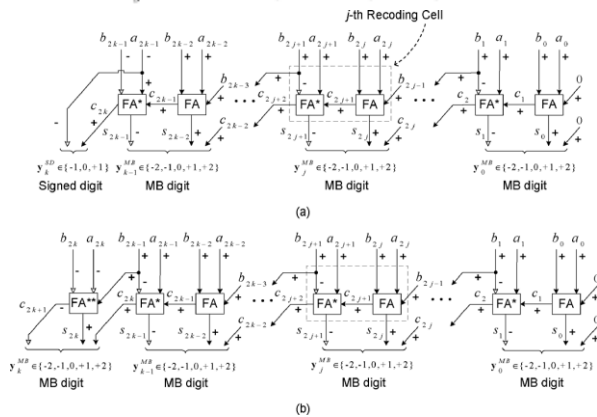


Figure 6. *S-MB1* recoding scheme for (a) even and (b) odd number of bits

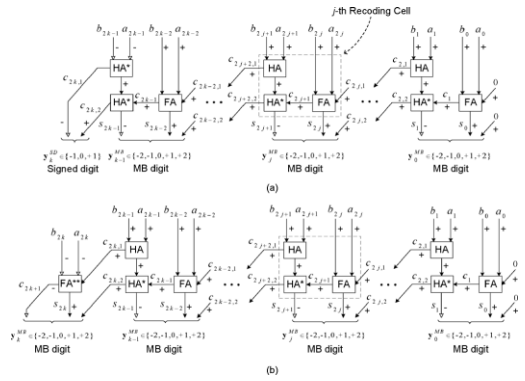


Figure 7. *S-MB2* recoding scheme for (a) even and (b) odd number of bits

2) S-MB2 Recoding Scheme:The second approach of the proposed recoding technique, *S-MB2*, is described in Figure 7 for even (Figure 7(a)) and odd (Figure 7(b)) bit-width of input numbers. We consider the initial values and . The digits , , are formed based on , and according to (8). As in the *S-MB1* recoding scheme, we use a conventional FA to produce the carry and the sum .The inputs of the FA are , and . The bit is the output carry of a conventional HA which is part of the recoding cell and has the bits , as inputs. The bit is the output sum of a HA* (basic operation – Table II, Figure 4(a)) in which we drive and the sum produced by a conventional HA with the bits , as inputs. The HA* is used in order to produce the negatively signed sum and its outputs are given by the following Boolean equations

$$c_{2j+2,2} = c_{2j+1,1} \vee (a_{2j+1} \oplus b_{2j+1})$$

$$s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1,1}$$

$$y_{k,even}^{SD} = -c_{2k,1} + c_{2k,2}$$

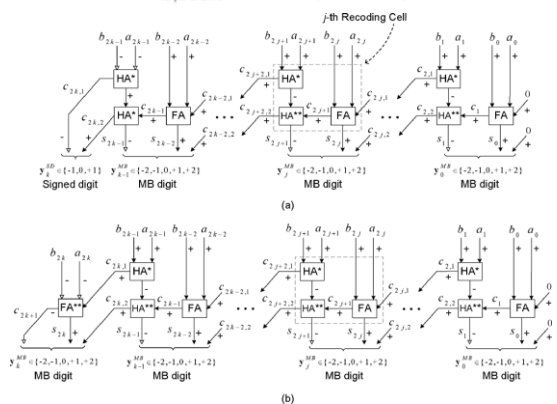


Figure 8. *S-MB3* recoding scheme for (a) even and (b) odd number of bits.

3) S-MB3 Recoding Scheme:The third scheme implementing the proposed recoding technique is *S-MB3*. It is illustrated in detail in Figure 8 for even (Figure 8(a)) and odd (Figure 8(b)) bit-width of input numbers. We consider that and . We build the digits based on, and according to (8). Once more, we use a conventional FA to produce the carry and the sum. The bit is now the output carry of a HA^* (basic operation – Table II, Figure 4(a)), which belongs to the) recoding cell and has the bits , as inputs. The negatively signed bit is produced by a HA^{**} (Table IV, Figure 4(b)) in which we drive and the output sum (negatively signed) of the HA^* of the recoding cell with the bits , as inputs. The carry and sum outputs of the HA^{**} are given by the following Boolean equations:

$$c_{2j+2,2} = c_{2j+1} \wedge (a_{2j+1} \oplus b_{2j+1})$$

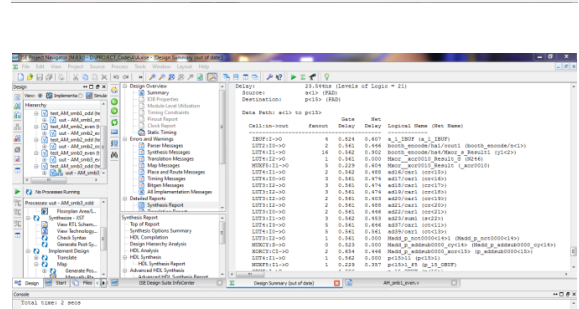
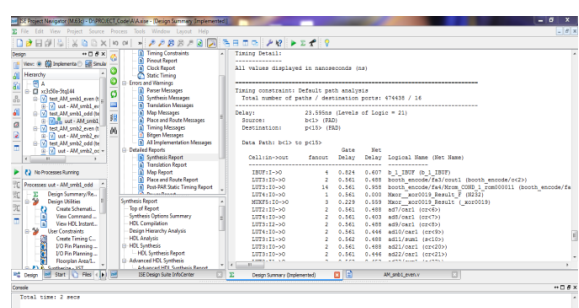
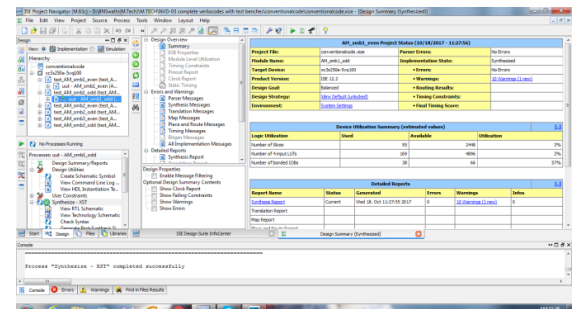
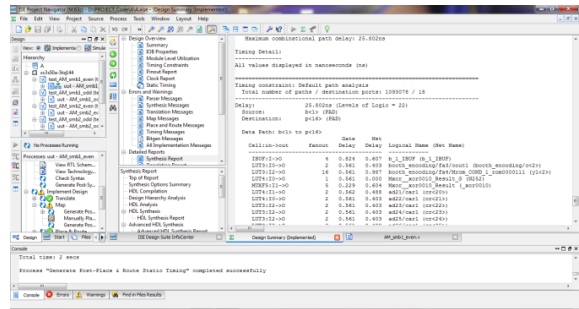
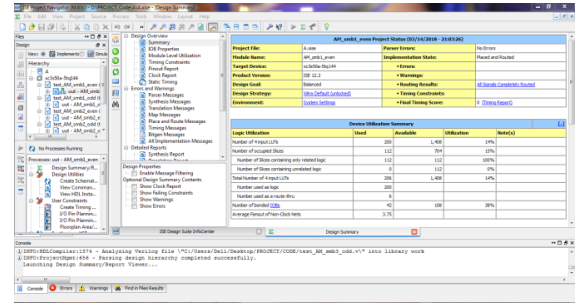
$$s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}$$

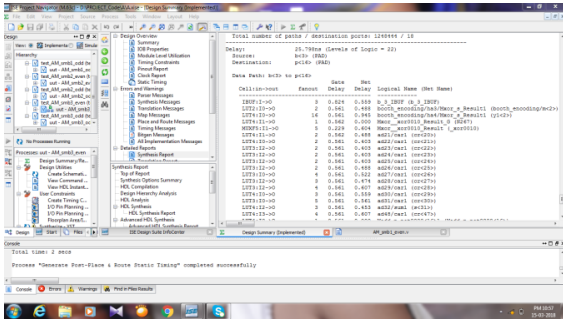
In case that both and comprise of even number of bits (Figure 8(a)), and are negatively weighted and we use the dual implementation of the HA^* (Table III, Figure 4(a)) in the recoding cell. Consequently, the output sum of the HA^* becomes positively weighted and the HA^{**} that follows has to be replaced with a HA^* . The most significant digits for both cases of even and odd bit-width of and , are formed as in *S-MB2* recoding scheme. The critical path delay of *S-MB3* recoding scheme is calculated as follows:

$$T_{S-MB3} = T_{HA^*, carry} + T_{FA, carry} + T_{HA^*, sum}$$

4) Unsigned Input Numbers:In case that the input numbers and are unsigned, their most significant bits are positively signed. Figs. 9–11 present the modifications that we have to make in all *S-MB* schemes for both cases of even (the two most significant digits change) and odd (only the most significant digit change) bit-width of and , regarding the signs of the most significant bits of and . The basic recoding block in all schemes remains unchanged.

IV. SIMULATION RESULTS





V. CONCLUSION

This paper focuses on optimizing the design of the Fused-Add Multiply (FAM) operator. We propose a structured technique for the direct recoding of the sum of two numbers to its MB form. We explore three alternative designs of the proposed *S-MB* recoder and compare them to the existing ones [12], [13] and [23]. The proposed recoding schemes, when they are incorporated in FAMdesigns, yield considerable performance improvements in comparison with the most efficient recoding schemes found in literature.

VI. REFERENCES

[1]. A. Amaricai, M. Vladutiu, and O. Boncalo, "Design issues and implementations for floating-point divide-add fused," IEEE Trans. Circuits Syst. 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]. J.J.F. Cavanagh, Digital Computer Arithmetic. New York: McGrawHill, 1984.

[4]. S. Nikolaidis, E. Karaolis, and E. D. Kyriakis-Bitaros, "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.

[5]. O. Kwon, K. Nowka, and E. E. Swartzlander, "A 16-bit by 16-bit MAC design using fast 5: 3 compressor cells," J. VLSI Signal Process. Syst., vol. 31, no. 2, pp. 77-89, Jun. 2002.

[6]. L.-H. Chen, O. T.-C. Chen, T.-Y. Wang, and Y.-C. Ma, "A multiplication-accumulation computation unit with optimized compressors and minimized switching activities," in Proc. IEEE Int. Symp. Circuits and Syst., Kobe, Japan, 2005, vol. 6, pp. 6118-6121.