The Modified Booth Encoder 2's complement multiplier

M. Martina

1 Introduction

In order to explain how the Modified Booth Encoder (MBE) multiplier works it is convenient to start from simple manipulation of 2's complement representation, which is used in Radix-2 multipliers. Let **a** and **b** be two 2's complement values, each represented with n bits:

$$\mathbf{a} = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i, \tag{1}$$

$$\mathbf{b} = -b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} b_j 2^j. \tag{2}$$

Let $\mathbf{c} = \mathbf{a} \cdot \mathbf{b}$, then $\mathbf{c}$ is represented with $2n$ bits. Stemming from (1) and (2) we can write $\mathbf{c}$ as:

$$\begin{aligned} \mathbf{c} = {} & a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} a_i b_j 2^{i+j} + \\ & - \left(a_{n-1}2^{n-1}\sum_{j=0}^{n-2} b_j 2^j + b_{n-1}2^{n-1}\sum_{i=0}^{n-2} a_i 2^i \right). \end{aligned} \tag{3}$$

The expression in (3) is depicted in the top part of Fig. 1, where the first n rows represent the first line of (3), i.e. elements, which are added together. The last two rows of the top part of Fig. 1 represent the second line of (3) and must be subtracted. This operation can be achieved inverting all the bits and adding '1', as shown in the middle part of Fig. 1. Then, adding the ones we obtain the final solution, which is shown in the bottom part of Fig. 1, where there are two ones placed at 2^n and 2^{2n-1}, respectively. As it can be observed, the Radix-2 solution produces partial products which are in the form $p_j = a \cdot b_j$ or

$$p_j = \begin{cases} 0 & \text{if } \bar{b}_j \\ a & \text{if } b_j \end{cases} \tag{4}$$

where $a = [a_{n-1}a_{n-2}\dots a_1 a_0]$.

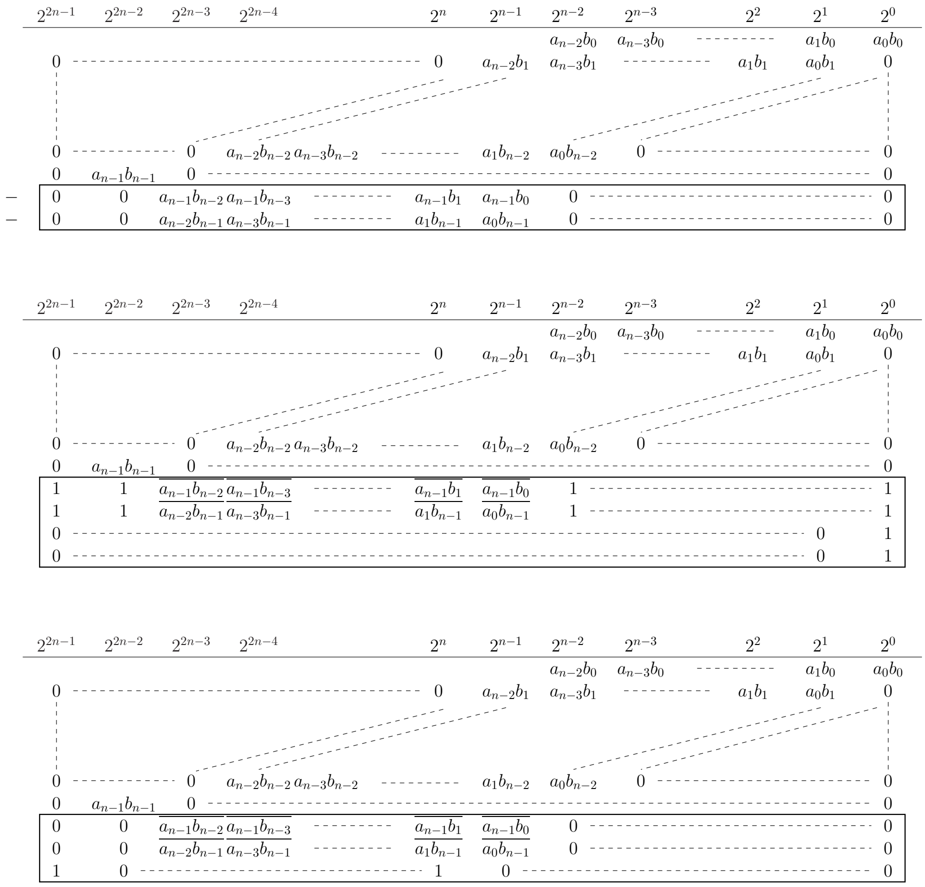

Figure 1: Derivation of the Radix-2 multiplier.

Table 1: Modified Booth Encoding.

$b_{2j+1}b_{2j}b_{2j-1}$	p_j
000	0
001	a
010	a
011	2a
100	-2a
101	-a
110	-a
111	0

2 Modified Booth Enconding (MBE)

MBE is an extension of the Radix-2 approach, namely instead of considering the multiplier on a bit-by-bit basis, more bits are analyzed simultaneously. Usually, MBE is a Radix-4 approach namely it produces half partial products with respect to the Radix-2 solution. This is achieved by dividing the multiplier in 3 bit slices (with $b_{-1} = 0$), where two consecutive slices feature a 1-bit overlap (see Fig. 2). If n is odd the multiplier must be sign extended to have “complete” triplets of bits. Then, each triplet of bits is exploited to encode the mutiplicand according to Table 1. As a consequence, the expression describing partial products, which can be derived from direct inspection of Table 1, is more complex in MBE than in Radix-2 solutions, namely $p_j = (b_{2j+1} \oplus q_j) + b_{2j+1}$, where

$$q_j = \begin{cases} 0 & \textbf{if } \left(\overline{b_{2j} \oplus b_{2j-1}}\right)\left(\overline{b_{2j+1} \oplus b_{2j}}\right) \\ a & \textbf{if } b_{2j} \oplus b_{2j-1} \\ 2a & \textbf{if } \left(\overline{b_{2j} \oplus b_{2j-1}}\right)\left(b_{2j+1} \oplus b_{2j}\right) \end{cases} \tag{5}$$

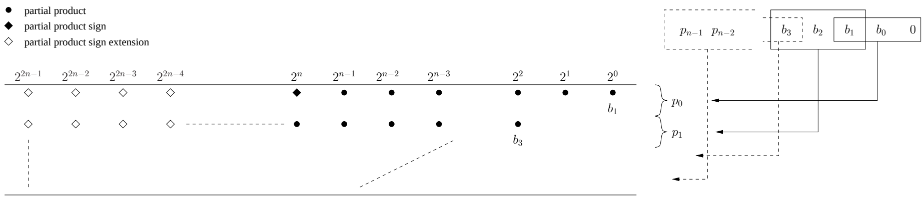

Figure 2: General scheme of a parallel MBE-based multiplier.

3 Adding partial products

A general scheme showing how partial products are added in an MBE-based multiplier is depicted in Fig. 2. The coverage of the dots can be made with any suited structure, including Wallace tree, Dadda tree, etc. As it can be observed, sign extension is needed to correctly add partial products. Unfortunately, this requires adders to properly cover all the dots. A simple and effective technique to reduce the number of adders required for covering partial product sign extension dots is presented in [1].

References

[1] M. Roorda. Method to reduce the sign bit extension in a multiplier that uses the modifieed Booth algorithm. *Electronics Letters*, 22(20):1061–1062, Sep 1986.