

Design Of High Accuracy And Hardware Efficient Fixed-Width Multiplier

K. Sindhuja

K. Suryakamatchi

S. Thangamani

R. Yamunapriyadharshini

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

V.S.B. ENGINEERING COLLEGE, KARUR

ABSTRACT

In this project we introduce a advanced error compensation circuit by using the triple group minor input correction vector to provide less error. By using minor input correction vector and by constructing the error compensation circuit mainly from the "outer" partial products, the hardware complexity not increased much as the multiplier input bits increase. In the proposed 16 x16 bits fixed-width multiplier, the truncation error can be lowered as compared with the direct-truncated multiplier and the transistor count can be reduced as compared with the full-length multiplier.

1. INTRODUCTION

In most high-speed digital signal processing (DSP) and multimedia applications, the multiplier plays an important role because it dominates the chip power consumption and operation speed. In DSP applications, in order to avoid infinite growth of multiplication bit width, we usually have to reduce the number of multiplication products. Cutting off n-bit less significant bit (LSB) output can construct a fixed-width multiplier with n-bit input and n-bit output. However, truncating the LSB part leads to a large number of truncation errors. Many truncation error compensation techniques [1]–[10] have been Presented to design an error compensation circuit with less truncation error and less hardware overhead. The compensation methods can be divided into two categories: compensation with constant correction value [1]–[3] and compensation with variable correction value [4]–[10]. The circuit complexity to compensate with constant corrected value can be simpler than that of variable correction value; however, the variable correction approaches usually can be more precise. In the approaches with

variable correction value, literature [4] proposed an input-dependent method by using probability, statistics, and linear regression analysis to find the approximate compensation value. The error compensation circuit is constructed by the partial product terms with the most-significant weight in the least-significant segment. The compensation value is dependent on the input number and thus has less truncation error. In [5], the error compensation algorithm made use of binominal distribution instead of uniform distribution used in [4] to model the probability of occurrence of multiplier inputs. This modification can bring a more precise error compensation result. Moreover, the compensation vector in [5] can directly inject into the fixed-width multiplier as compensation, which does not need extra compensation logic gates. Therefore, the fixed-width multiplier area can be smaller than [4]. In [6], a two-dimensional conditional estimation method was proposed to compensate truncated error based on both the dependency among the partial product terms and multiplication inputs. The error compensation in [6] can be more precise; however, the hardware is too complex. In [7], [8], multiple-input error compensation vector designs were proposed to further enhance the error compensation precision. Unlike [4] or [5] to set the same weight for each partial product terms in the input correction vector, they applied different weights to each input correction vector element. In [8], "inner" partial products were designed to have a higher weight with respect to "outer" partial products. To take into account different weights of input correction (IC) partial products, the IC vector was divided into two disjointed sets with dual addition trees to compute the error compensation value. In this way, the compensation value can be more approximated to the expected results. Hence it performed better results in terms of error

compensation. Recently, the design in [8] was further extended in [9] and [10]. In [9], a parallel configurable error-compensation circuit was proposed to perform nearly the same error compensation precision as [8], but with lower computation delay. In [10], a variable correction to include the partialproducts of LSB part was proposed to trade-off between hardware complexity and error compensation precision. Nowadays [8]–[10] are the state-of-the-art fixed-width multiplier designs that can perform lower error with efficient hardware. In this paper, we consider the impact of truncated products with the second most significant bits on the error compensation, which is similar to [10] but with lower hardware complexity. We propose a new error compensation circuit by using the dual group minor input correction (MIC) vector to further lower IC vector compensation error in [8]. By utilizing the symmetric property of MIC, fan-in can be lowered to half and hardware in up-MIC and down-MIC can be shared. Therefore, the hardware complexity of error compensation circuit can be lowered. Moreover, the hardware complexity just increases slightly as the multiplier input bits increase because we construct the proposed error compensation circuit mainly by the “outer” partial products. As compared with the state-of-the-art design in [8]–[10], the proposed fixed-width multiplier not only performs with lower compensation error but also with less hardware complexity, as multiplier input bits increase.

2. BOOTH MULTIPLIER WITH CSA

Multiplication can be divided into three steps:

- Generating partial products, summing up all partial
- Partial product reduction using carry save addition, until only two rows remain,
- Adding the last two rows of partial products by using a carry propagation adder.

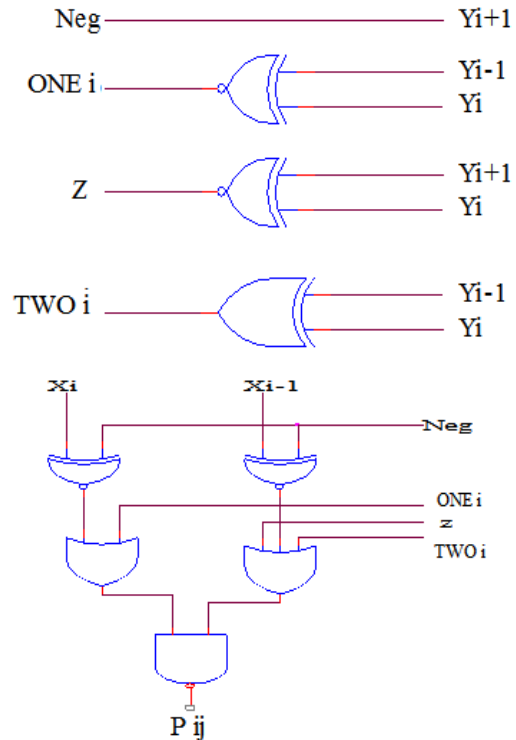


Figure1: Circuit of Encoder and Decoder

The existing MBE partial product array for an 8 x 8 multiplier uses the sign extension circuitry. The existing MBE partial product array has two disadvantages:

- 1) An additional partial product term at the position of (n-2)th bit,
- 2) Performance is low at the LSB-part compared with the non-Booth design when using the TDM algorithm.

To remedy the two disadvantages, the LSB part of the partial product array is modified. Using Boolean minimization the R_LSB and the N_cin terms are joined together and is further simplified. The new equations for the R_LSB and N_cin can be written as equations 1 and 2 as,

$$R_LSB_i = Y_{LSB} * (x_{2i-1} \oplus x_{2i}) \tag{1}$$

$$N_Cin_i = x_{2i+1} * x_{2i+1} + x_{2i} * x_{2i-1} + y_{LSB} * x_{2i} + y_{LSB} \tag{2}$$

By combining the proposed new MBE decoder and the modified partial product array, the MBE-based multiplier can have better perform than the non-Booth multipliers . For the final adder, a new algorithm that optimizes final adder incrementally is proposed. The proposed algorithm solves final adder problem efficiently for any size and shows performance improvement up to 25 percent for the final adder.

2.1. PARTIAL PRODUCT GENERATION

Modified Booth Multiplier

Three steps are there in this Multiplication.

First step is to generate the partial products. Second step is to add the generated partial products until the last two rows are remained. Third step is to compute the final multiplication results by adding the last two rows.

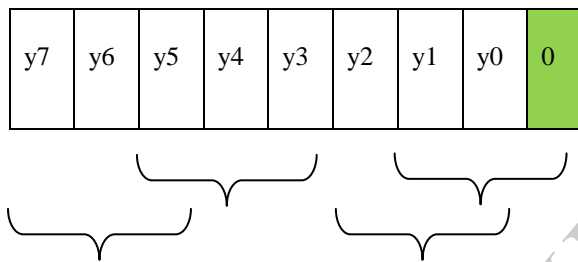


Figure 2 : 8-Bit Modified Booth Encoding

The figure 2 shows the grouping of 8-bit multipliers into 4 groups. The Modified Booth 2 or the radix 4 booth encoding table 1 shows the way of groping the even bit multiplier. Here by default a 0 is padded with LSB of the multiplier for grouping.

Modified Booth Encoding Technique

| Y _{i+1} | Y _i | Y _{i-1} | ACTION | NEG | 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 | A _{j-1} ' |
| 1 | 0 | 1 | -A | 1 | 0 | 1 | A _j ' |
| 1 | 1 | 0 | -A | 1 | 0 | 1 | A _j ' |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

The modified Booth algorithm reduces the no of partial products by half in the first step. The Modified Booth Encoding scheme proposed here is the most efficient Booth encoding and decoding scheme. To multiply X (multiplicand) by Y (multiplier) using the modified Booth algorithm starts from grouping Y by three bits and encoding into one of {-2, -1, 0, 1, 2}.

Table show the rules to generate the encoded signals by MBE (Modified Booth Encoding) scheme and Figure 10 shows the corresponding logic diagram. The final multiplication results are generated by adding the last two rows. The Carry Propagation Adder is used in this step to add final two rows with the accumulator value.

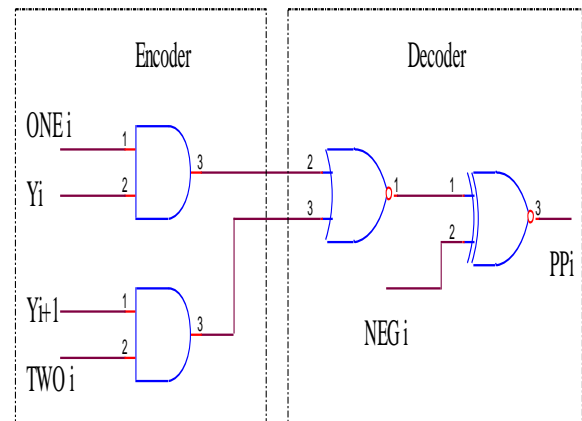


Figure 3 : Modified Booth Encoder and Decoder Circuits

2.1.2. Carry Save Adders

The Carry Save Adders are the efficient adders to reduce delay of critical path in partial products. Because the disadvantages of RCAs are

- Not very efficient when large number bit numbers are used.
- One of the most serious drawbacks of this adder is that the delay increases linearly with the bit length.

In many cases we need to add several operands of larger length together, carry save adders are good ideal for this type of addition. A carry save adder has a ladder of standalone full adders, and performs number of partial additions. The principal idea is that the carry has a higher power of 2 and thus is routed to the next column. Performing additions with Carry save adder saves time.

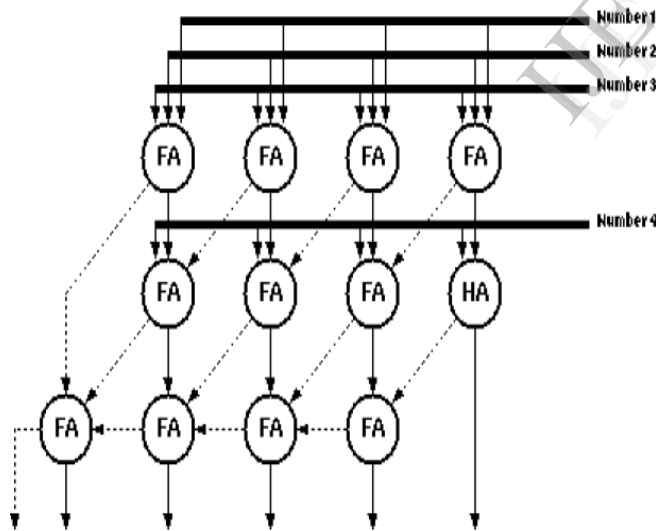


Figure 4 : Carry Save Adder for as Four Bit Numbers

In this method, for the first 3 numbers a row of full adders are used. Then a row of full adders are added for each additional number. The final results obtained in the form of two numbers are SUM and CARRY, then summed up with a carry propagate adder . An example of 4 numbers addition is shown in figure 5.

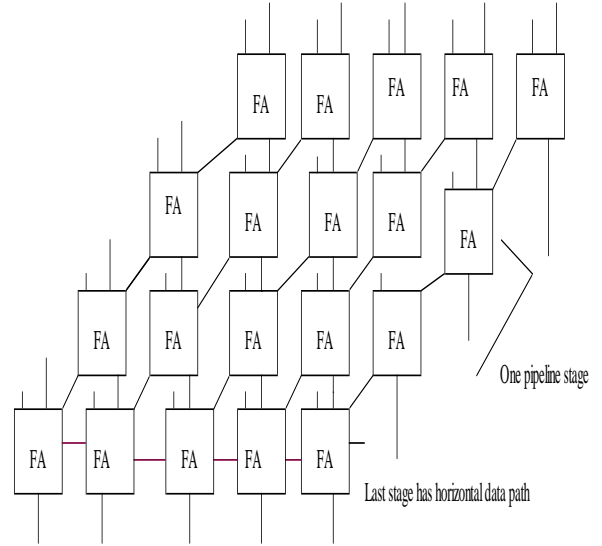


Figure 5 : Parallel Array Multiplier using CSA

3.MINOR INPUT CORRECTION (MIC)

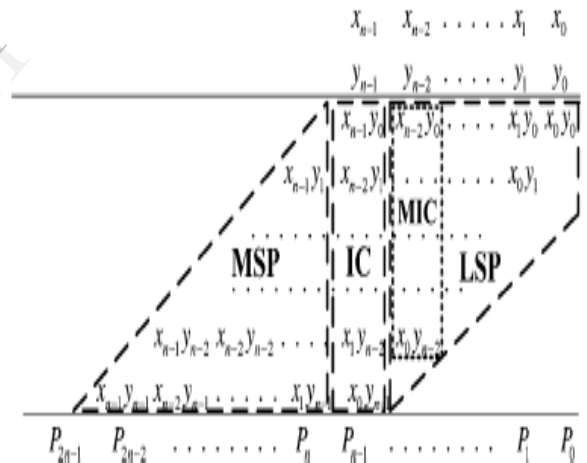


Fig. 1. *n*-bit Baugh-Wooley multiplier partial product array consists of MSP, IC, and LSP, in which MIC is the partial product vector with the most significant bits of LSP.

Signed multiplication is a difficult process. With unsigned multiplication there is no need to take the sign of the number into consideration. However in signed multiplication the same process cannot be applied because the signed number is in a 2's complement form which would yield an incorrect result if multiplied in a similar fashion to unsigned multiplication. But Booth's algorithm preserves the sign of the result.

4. CONCLUSION

In this paper, a low-error and area-efficient fixed-width multiplier by using the dual group minor input correction vector is presented. The proposed fixed-width multiplier performs with less compensation error but also with less hardware complexity, especially when multiplier input bits increase, time delay is reduced, lower power, higher speed.

5. REFERENCES

1. Abdelgawad.A and M. Bayoumi (2007): 'High speed and area-efficient (MAC) multiply accumulate unit for digital signal processing applications,' in Proc. IEEE International Symposium on Circuits And Systems. (ISCAS), pp. 3199–3202.
2. Hatamian.M., and G. L. Cash (1986): 'A 70-MHz 8-bit x 8-bit parallel pipelined multiplier in 2.5- μ m CMOS,' IEEE J. Solid-State Circuits, Vol. JSSC-21, No. 4, pp. 505–513.
3. Hoang. T. T., M. Sjalander, and P. Larsson-Edefors (2009): 'High-speed, energy-efficient 2-cycle multiply-accumulate architecture,' in Proc. IEEE International. SOC Conf. (SOC), pp.119–122.
4. Kim.S, C. H. Ziesler, and M. C. Papaefthymiou(2003): 'Fine-grain real-time reconfigurable pipelining,' IBM J. Research and Development, Vol. 47, No. 5-6, pp. 599–609.
5. Shiann-Rong Kuang, Jiun-Ping Wang, and Cang-Yuan Guo(2009): 'Modified Booth Multipliers With a Regular Partial Product Array,' IEEE Transactions on circuits and systems, Vol. 56, No. 5, pp-404-408.
6. Sjalander. M., and P. Larsson-Edefors (2009): 'Multiplication acceleration through twin precision,' IEEE Transactions. Very Large Scale Integr. (VLSI) Syst., Vol. 17, No.1, pp. 1233–1246.
7. Sreehari Veeramachaneni, Kirthi Krishna M, Lingamneni Avinash, Sreekanth Reddy Puppala, M.B. Srinivas (2007): 'Novel Architecture for High Speed and Low Power 3-2,4-2 and 5-2 Compressors,' IEEE Conference on VLSI Design, pp-33-39.
8. Tung Thanh Hoang, Magnus Sjalander and Per Larsson-Edefors, (2010) : '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, Vol. 57, No. 12, pp. 3073-3081.
9. Villeger D and V.G. Oklobdzija (1993): 'Analysis of Booth Encoding Efficiency in Parallel Multipliers Using Compressors for Reduction of Partial Products,' Proc. IEEE 27th Asilomar Conference on Signals, Systems, and Computer, Vol. 1, pp. 781-784.
10. Villeger D and V.G. Oklobdzija (1993): 'Evaluation of Booth Encoding Techniques for Parallel Multiplier Implementation,' Electronics Letters, Vol. 29, No. 23, pp. 2,016-2,017.