ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 2, Issue 6, June 2015

High Speed Speculative Multiplier with Correction Unit

Meenu I Chandran

Mar-Athanasius College of Engineering/ECE Department, Ernakulam, India
meenuichandran@gmail.com

Abstract*— Integer Multiplication is considered as the basic fundamental building block of digital design .It deeply affects the performance of the DSP activities. This paper proposes a novel approach to build integer multiplication circuits that is based on speculate method. This is a technique which performs a faster-but occasionally wrong-operation. The rare case of error is corrected using a multi-cycle error correction circuit. The proposed speculative multiplier uses a three dimensional method reduction tree. The method is implemented using three steps: partial products recoding, partial products partitioning, speculative compression. It also uses speculative counters, that are faster than a conventional tree using full-adder and half-adders. Comparisons with conventional multiplier show that speculation is more effective when high speed is required. Speculative multipliers allow reaching a higher speed when compared with its conventional counterparts and are also quite effective in terms of power dissipation, when a high speed operation is required.

***Index Terms*— Digital arithmetic operations, multiplication, speculative functional units, speculative multipliers.**

I. INTRODUCTION

A system's performance is generally determined by the performance of the multiplier units because the multiplier is generally considered as the slowest element in the system. Moreover, it is also generally the most area consuming unit. Hence, optimizing the speed and area of the multiplier is a major design concern. However, area and speed are usually conflicting constraints so that improving speed results mostly in large areas.

Some of the recent results in the technical literature [1],[2] points out that, faster circuits can be obtained by adopting a speculative approach. Speculative circuits are based on the idea of performing a faster-but occasionally wrong-operation.A multi-cycle error correction circuit is used only in the rare case of error. This approach is fairly general and can be applied whenever there are several functional units involved in some kind of dependency relationships.

In this paper we propose a novel approach to build a high speed multiplier using some speculative method. The proposed speculative multiplier is implemented using three steps: partial products recoding, partial products partitioning, speculative compression. The speculative multiplier uses speculative counters. The speculative counters are faster than conventional counters.

II. SPECULATIVE MULTIPLIER ARCHITECTURE

Let us consider two inputs A and B that needs to be multiplied. Consider A as the multiplicand and B as the multiplier. The multiplication result is termed as Y.

$$A = a_{n-1}2^{n-1} + \ldots\ldots\ldots\ldots + a_0 \quad (1)$$

$$B = b_{n-1}2^{n-1} + \ldots\ldots\ldots\ldots + b_0 \quad (2)$$

$$Y = A\,.\,B = y_{2n-1}2^{2n-1} + \ldots\ldots\ldots\ldots + y_0 \quad (3)$$

$$= \sum\sum a_i\, b_j\, 2^{i+j}$$

Hence the computation of requires the summation of the partial products a_ib_j according to their weights 2^{i+j}.

To introduce the proposed technique, we assume that input bits a_i and b_j are independent and equally likely. As the consequence of our assumption, the probability of being one for each partial product is 0.25.

Fig. 1 shows us the arrangement of partial products matrix (PPM) for a 16 X 16 multiplier. It may be observed that, the rightmost and the leftmost columns of the PPM comprise a small number of partial products, whereas the inner columns have higher number of partial products. The critical path of the circuit is related to the height of the PPM. The critical path decides the delay. The higher the matrix, the higher the multiplier delays. In principle, a speculative carry-save reduction tree shall be obtained by deleting

some partial products from the inner columns of the PPM. However, blind deletion of some partial products will increase the overall error probability and hence such an approach would be unacceptable. Thus, we propose a new technique to generate a speculative carry-save reduction tree for multiplication based on three steps: partial products recoding, partial products partitioning, speculative compression.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 2, Issue 6, June 2015

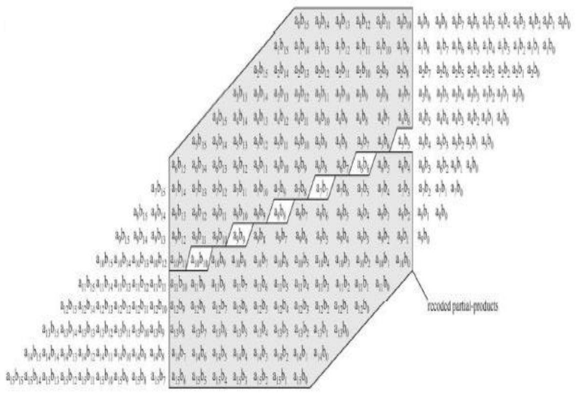

Fig. 1. 16 X 16 bit Multiplier partial products matrixes

A. Partial Product Recoding

Let us consider two partial products a_ib_j and a_jb_i of the i+j-th column of the PPM and let us introduce the following two modified partial products:

$$A_{i,j} = a_ib_j \text{ AND } a_jb_i \qquad (4)$$
$$O_{i,j} = a_ib_j \text{ OR } a_jb_i$$

The terms $A_{i,j}$ and $O_{i,j}$ are called the modified partial products. The modified partial products are such that $A_{i,j} + O_{i,j} = a_ib_j + a_jb_i$.. Thus , in the i + j-th column of the PPM, we can replace the couple of partial products a_ib_j and a_jb_i with the modified partial products $A_{i,j}$ and $O_{i,j}$. The advantage of this recoding is the introduction of low probability terms in the PPM.

B. Partial Product Partitioning

Only low-probability $A_{i,j}$,terms are included in the speculative method tree. We recode only the partial products belonging to the large columns of the PPM. An example is shown in Fig. 1(b). Here we use only the partial products of the columns 11,12,..............22 are recoded.

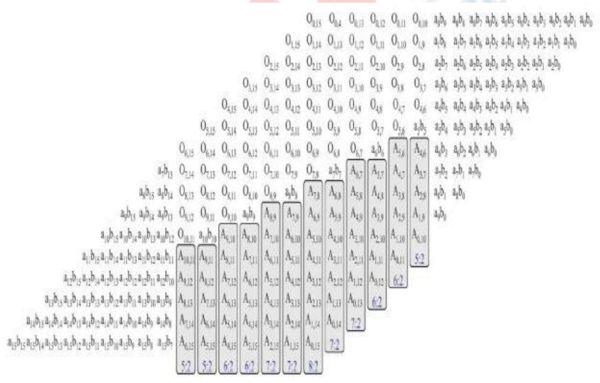

Fig 2 .Partial products matrix after recoding.

C. Speculative Compression

Although the probability of $A_{i,j}$ is reduced with respect to original partial products, simple deletion $A_{i,j}$ of terms would still introduce some error probability. Thus, instead of deleting $A_{i,j}$ terms, we sum them in an approximate way. The summation is done using speculative counters.

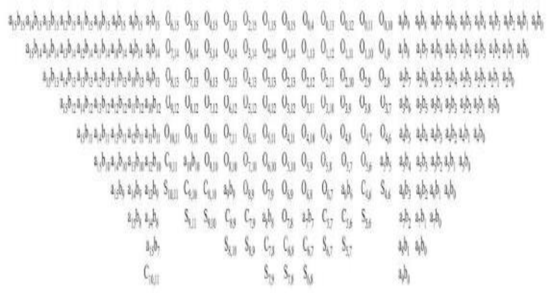

Fig .3 Partial products matrix after speculative compression

An *(m :2)* speculative counter has m inputs $(x_0,x_1,........x_{m-1})$ and only two outputs: Sum and Carry . The speculative counter counts the number of input bits that are "1" and encode the result on *C* and *S*, assuming that no more than three inputs are high. Similarly to full-adders and half-adders, the output *C* has a weight doubled with respect to *S* so that:

$$2C + S = x_0 + x_1 + ...+x_{m-1} \leq 3 \qquad (5)$$

For m=2 and m=3 , the speculative counter is designed to give the correct count of high input bit and hence corresponds to either to a half-adder (m=2) or to a full-adder (m=3). For the case : m > 3, it is impossible to represent the sum $x_0 + x_1 + ... x_{m-1}$ by using only *C* and *S* signals for all the possible input configurations.

The speculative counter performs the output calculations assuming that no more than three inputs are high: if this condition is not verified an error occurs and the multiplication result is wrong and it must be corrected in the next cycle. Compared to conventional counters, speculative counters are simpler as they produce a lower number of output bits (2 instead of $[\log_2(m+1)]$) and hence are faster. This explains the improvements in multiplier performance.

III. MULTIPLIER ARCHITECTURE

The complete architecture of the proposed speculative multiplier is outlined in Fig. 4. The multiplier inputs are firstly processed by the partial products generation and recoding block. This block computes all the partial products a_ib_j and recodes those belonging to the inner columns of the PPM, generating

$A_{i,j}$ and $O_{i,j}$ and recoded partial products. The Ai,j terms are

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 2, Issue 6, June 2015

processed using the speculative counters obtaining the reduced $S_{i,j}$ and $C_{i,j}$ terms. The reduced $S_{i,j}$,$C_{i,j}$ terms, the un-recoded a_ib_j and the recoded partial products $O_{i,j}$ are summed together by using the TDM carry-save tree. The TDM considers the different arrival times of various inputs and tries to make proper connections to full adders so that the delay throughout each path is approximately the same. Hence, the late arriving outputs of the speculative counters are connected to the shortest delay path in the TDM. Generally, at the TDM outputs we obtain the delay profile similar to the one of a conventional multiplier. The maximum delay, however, is reduced compared to conventional multipliers. This is due to the use of speculative counters.

The two outputs obtained from the TDM tree are added by using a speculative adder, obtaining the speculative result Y_s. It may be noted that any speculative adder can be employed in this architecture. The one proposed in [3][4] is used in this architecture.

As in any speculative functional unit, no information loss can be tolerated at the output of the proposed multiplier. Since each speculative compressor can introduce an error, a correction block is required for each speculative compressor. This correction block receives the same inputs as that of the speculative compressor and computes two outputs: an error flag (E) and a suitable correction word (EW) . The error flag (E) is high if four or more inputs of the speculative compressor are "1." The correction word (EW) is computed so that by adding (EW) to the speculative compressor output (2C +S) we obtain the correct result.

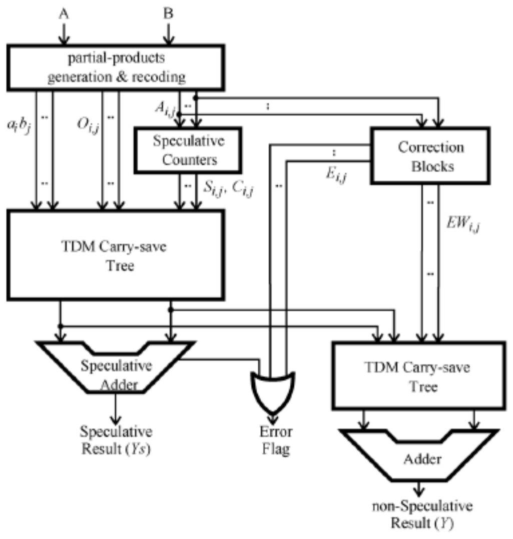

Fig. 4 Architecture of proposed Speculative Multiplier.

As it can be observed in the right-hand part of Fig. 4, all error flags produced by each correction block are OR-ed together and OR-ed with the error flag of the speculative adder, to obtain the error flag of the speculative multiplier. In case of error, the non speculative multiplication result is computed by summing the correction words with the outputs of the TDM carry-save tree included in the speculative part of the multiplier.

It is interesting to observe that in this architecture the error correction part of the speculative adder is not needed—only the error flag is required. In case of misprediction, in fact, the multiplier output is , which is computed independently of the speculative adder.

A. (m:2) Speculative Counter Implementation

We recall that an (m:2) speculative counter, defined in previous section, is a component with *m* inputs(x_0,x_1,. . . . . . x_{m-1}) and two outputs: sum(S) and Carry(C) . The value (2C+S) is equal to the arithmetic sum of the inputs when no more than three inputs are high. From this definition, the S output can be simply computed as a tree of XOR gates of the inputs.

The calculation of the signal, on the other hand, poses a few difficulties. Let f $\geq_2$(x_0,x_1,. . . . . .x_{m-1})be the binary function that is high if there are at least two input signals xi high. The function f $\geq_2$(x_0,x_1,. . . . . .x_{m-1}) clearly corresponds to the C output of the (m:2)speculative compressor. In the case of two, three, and four inputs, the function $f_{\geq 2}$ can be easily and efficiently computed. In the case of two inputs we readily have:

$$f_{\geq 2}(x_0 , x_1) = x_0 \cdot x_1 \qquad (6)$$

The function $f_{\geq 2}$ applied to three inputs corresponds to the carry function of a conventional full-adder :

$$f_{\geq 2}(x_0 , x_1 ,x_2) = x_0 \cdot x_1 + x_0 \cdot x_2 + x_1 \cdot x_2 \qquad (7)$$

The function $f_{\geq 2}$ applied to four inputs can be computed as:

$$f_{\geq 2}(x_0 , x_1 ,x_2)$$
$$= x_0 \cdot x_1 + x_0 \cdot x_2 + (x_0 +x_1)(x_2 + x_1) \qquad (8)$$

B. Correction Block Implementation

The correction circuits are not critical for the overall multiplier. The error correction words are summed together in a two cycle sub circuit, while the error flags are only ORed together to generate the single-cycle error output. This OR is normally outside the critical path that goes through the compressor tree and the VMA. While the error correction blocks may be described in a behavioural way and automatically synthesized, the function they realize can be easily described in terms of basic components such as standard logic gates and full-adders and half-adders. For instance, the (4:2) error correction circuit must yield a single-bit word with weight 2, being high only if all input signals are high, i.e., an AND4 gate. This bit also acts as the error flag . The correction block, in this case, must assert the error flag when 4 or 5 inputs of th compressor are high. This condition is false if at least two inputs of the compressor are zero. Hence, the error flag can be computed as

$$E = \overline{f_{\geq 2}(\overline{x}_0, \overline{x}_1, \overline{x}_2, \overline{x}_3, \overline{x}_4)} \quad (9)$$

IV. CONCLUSION

In the paper a novel speculative multiplier for high-speed application is proposed. The circuit recodes some of the multiplier partial products and uses a speculative compression tree to sum the recoded partial products. A speculative adder is used in the final carry-propagate addition. The speculative multipliers can be useful in critical applications justifying the more complex architecture and the handling of multi-cycle paths. In cases where the multiplier speed is not critical, the use of speculative units appears unjustified. Additional work may further improve the performance of speculative multipliers: the implementation of the output of speculative counters could be optimized by considering don't care conditions. This work investigates only non-Booth multipliers. For large input widths, Booth multipliers may yield an advantage in terms of speed, area, and power.

REFERENCES

[1] L. S. Wallace, "A suggestion for fast multipliers," *IEEE Trans. Computer.*, vol. EC-13, pp. 14–17, Feb. 1964.

[2] L. Dadda, "Some schemes for parallel multipliers," *Alta Frequenza*, vol. 34, Mar. 1965

[3] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in *Proc. Design, Autom., Test Eur. (DATE)*, Mar. 2008.

[4] A. Cilardo, "A new speculative addition architecture suitable for two's complement operations," in *Proc. Design, Autom., Test Eur. Conf. Exhib. (DATE)*, Apr. 2009, pp. 664–669.

[5] S. Veeramachaneni, K. Krishna, L. Avinash, S. Puppala, and M. B. Srinivas, "Novel architectures for high-speed and low-power 3-2, 4-2 and 5-2 compressors," in *Proc. 20th IEEE/ACM Int. Conf. VLSI Design (VLSID 2007)*, Jan. 2007.