© 2018 IJSRSET | Volume 4 | Issue 7 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

A Modified Partial Product Generator Using RADIX-4 to Remove ECW

Neha R.Dhodre[1], Prof. Sunil R. Gupta[2]
[1]PG Scholar, Department of E &TC, JD College of Engineering and Management, Nagpur, Maharashtra, India
[2] H.O.D., Department of E & TC, JD College of Engineering and Management, Nagpur, Maharashtra, India

ABSTRACT

Adders are the most important element of the arithmetic unit especially fast parallel adder. Redundant binary signed digit (RBSD) adders are designed to perform high speed arithmetic operations. Generally in a high radix modified booth encoding algorithm the partial products are reduced in multiplication process. While designing high performance multipliers a redundant binary (RB) representation can be used due to its high modularity and carry-free addition, The traditional RB multiplier requires an extra RB partial product (RBPP) row. Redundant binary representation (RBR) is a numeral system that uses more bits than needed to represent a single binary digit so that most numbers have several representation because an error-correcting word (ECW) is generated by both the radix-4 Modified Booth encoding (MBE) and the RB encoding. This results in an additional RBPP accumulation stage for the MBE multiplier. In this thesis, a new RB modified partial product generator (RBMPPG) is proposed; it removes the extra ECW therefore saves one RBPP accumulation stage. Hence the proposed work generates fewer partial product rows than a traditional RB MBE multiplier. Operation of the system over time show that the proposed work based designs considerably improve the area and power consumption when the word length of each operand in the multiplier is at least 32 bits; these reductions over previous NB multiplier designs incur in a modest delay increase (approximately 5 percent). By using the proposed RB multiplier design the power-delay product can be reduced by up to 59 percent when compared with existing RB multipliers.

Keywords : Redundant binary, modified Booth encoding, RB partial product generator, RB multiplier.

I. INTRODUCTION

The digital multiplier is a existing arithmetic unit in microprocessors, digital signal processors, and emerging media processors. It is also a unchanged operator in application specific data path of video and audio codes, digital filters, computer graphics, and embedded systems. Compared with many other arithmetic operations, multiplication is time as well as power consuming. The critical paths dominated by digital multipliers often impose a speed limit on the entire design. Hence, VLSI design for high-speed multipliers, with low energy dissipation, is still a popular research subject. Redundant binary (RBR) representation is one of the signed digit representations first introduced by Avizienis in 1961 which provides carry-propagation-free addition for fast parallel arithmetic. Many algorithms and architectures have been proposed to design high-speed and low-power multipliers. A normal multiplication by digital circuits is a three step process. In the first step, partial products are generated i.e the product of one term of a multiplicand and one term of its multiplier. in the second step, all partial products are added by a partial product reduction tree until two partial product rows remain. In the third step, the two partial product rows are added by a fast carry propagation adder or

carry- lookahed adder. Two methods have been used to perform the second step for the partial product reduction. A first method uses 4-2 compressors, while a second method uses redundant binary (RB) numbers. Both methods allow the partial product reduction tree to be reduced at a rate of 2 to 1. The RB addition is carry-free, making it a favourable substitute for 2's complement multi-operand addition in a tree-structured multiplier. Similar to a normal binary (NB) multiplier, an RB multiplier is dissect into three stages and consists of four modules: the Booth encoder, decoder), RB partial product accumulator, and RB-to-NB converter. A Radix-4 Booth encoding or a modified Booth encoding (MBE) is usually used in the partial product generator of parallel multipliers to reduce the number of partial product rows by half [5-6] [10-13]. A RBPP row can be obtained from two adjacent NB partial product rows by inverting one of the pair rows [5-6]; an N-bit conventional RB MBE (CRBBE-2) multiplier requires N/4 RBPP rows. An additional error-correcting word (ECW) is also required by both the RB and the Booth encoding [5-6] [14]; therefore, the number of RBPP accumulation stages (NRBPPAS) required by a power-of-two word-length (i.e., 2 -bit) multiplier is given by: NRBPPAS = log (N /4 + 1)

= n – 1, if N = 2^n.

This paper focuses on the RBPP generator for designing a 2 - bit RB multiplier with fewer partial product rows by eliminating the extra ECW. A new RB modified partial product generator based on MBE (RBMPPG-2) is proposed. In the proposed RBMPPG-2, the ECW of each row is moved to its next neighbor row. Furthermore, the extra ECW generated by the last partial product row is combined with both the two most significant bits (MSBs) of the first partial product row and the two least significant bits (LSBs) of the last partial product row by logic simplification. Therefore, the proposed method reduces the number of RBPP rows from N /4 + 1to N /4, i.e., a RBPP accumulation stage is saved. The proposed method is applied to 8×8-bit, 16×16-bit, 32×32-bit, and 64×64-bit RB multiplier designs; the designs are synthesized using the NanGate 45nm Open Cell Library. The proposed designs achieve significant reductions in area and power consumption compared with existing multipliers when the word length of each of the operands is at least 32 bits.

II. LITERATURE SURVEY

A high-radix Booth encoding technique can reduce the number of partial products. However, the number of expensive hard multiples (i.e., a multiple that is not a power of two and the operation cannot be per-formed by simple shifting and/or complementation) increases too. Besli et al. noticed that some hard multiples can be obtained by the differences of two simple power-of-two multiplies. A new radix-16 Booth en-coding (RBBE-4) technique without ECW has been pro-posed, it avoids the issue of hard multiples. A radix-16 RB Booth encoder can be used to overcome the hard multiple problem and avoid the extra ECW, but at the cost of doubling the number of RBPP rows. Therefore, the number of radix-16 RBPP rows is the same as in the radix-4 MBE. However, the RBPP generator based on a radix-16 Booth encoding has a complex circuit structure and a lower speed compared with the MBE partial product generator when requiring the same number of partial products.

III. FAST PARTIAL PRODUCT GENERATOR

The proposed partial product generator generates RB partial products, without any carry propagation delay or any additional hardware. For a multiplicand 'y' the radix-4 Booth encoder will have five different NB partial products{–2y,– y,0, y,2y}. Instead of generating –2y and –y in two's complement form the multiplier retains the partial products in their one's complement form and introduces an extra bit '1' along with the partial products. The NB partial product –y obtained from Booth encoder is expressed as(y,1) , where y is the one's complement of y. The set of partial products obtained from Booth encoder is represented as {(2y,1),(y,1),(0,0),(y,0),(2y,0)}.

A NB partial product A can be represented as A = (A*+a)

TABLE I. ENCODING FOR NB PARTIAL PRODUCTS

A	B	a	b	Z
+	+	0	0	-1
+	-	0	1	0
-	+	1	0	0
-	-	1	1	1

Where A* = A and a = 1, when A is negative and A* = A and a = 0, when A is positive. The sum of two NB partial products A and B can now be expressed as

$$A + B = (A^{*} + a) + (B^{*} + b)$$
$$= A^{*} + B^{*} + a + b$$
$$= (\overline{A^{*}} - B^{*}) - 1 + a + b$$

Using the RB Encoding 2 shown in Table I, the above equation can be expressed as

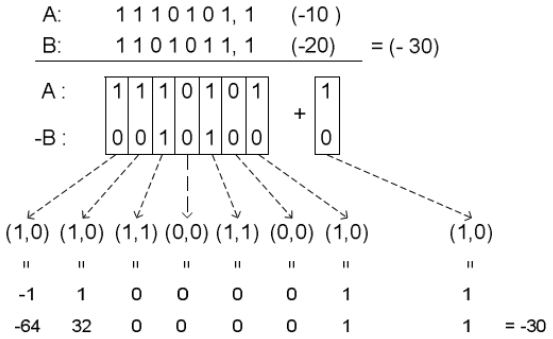
$$A + B = (A^{*}, B^{*}) - 1 + a + b$$

For different positive and negative numbers A and B, the values of a and b will be chosen according to Table II. It can be observed that a and b are nothing but the sign bits of A and B respectively. If Z = a + b - 1, Equation 6 can be modified as

$$A + B = (A^{*}, B^{*}) + Z$$

Where Z can be coded according to Table II.

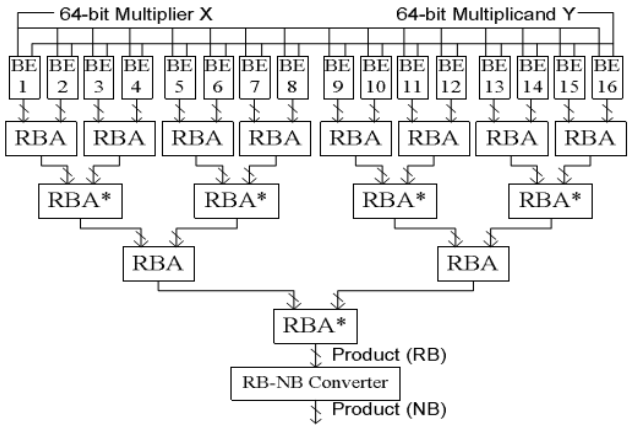
The extra RB digit from each RB operand forms an extra operand, which can be fed into the next partial product accumulation stage as shown in Makino [5]. This correction word will be having the format …0Z000Z000Z, where Z ϵ {1, 0, -1}. The addition of two NB partial products A = -10 and B = -20 according to Table II encoding is shown in Fig. 1. The two partial products are grouped along with the extra bit. In this case both numbers are expressed in their one's complement representations. The extra bits are '1' for both, and are shown separately in Fig. 1. The bit pair (A,B) is also shown in Fig 1. These bit pairs represent the sum A+B in RB notation. The equivalent RB number can be obtained using Encoding 2 in Table I and is shown at the bottom. The extra bit position is also assigned unit weight. The RB result obtained can be reconverted into its equivalent decimal value using a negative weight for the MSB bit. This results in the final sum of –30.



Example of RBPPG using one's complement arithmetic

The above method avoids any kind of carry propagate operation during partial product generation, and simply expresses the partial products in one's complement NB format for a negative number. The extra bit for each NB partial product is same as the sign bit of each operand. Contrary to Kim's technique [10], the correction bit Z is found directly from the grouping, instead of a combination of RB and Booth recoding terms. Also, the correction digit is limited to one per RB partial product when compared to one per NB partial product in earlier designs. The RBPPG does not use any gates (including inverters) for obtaining the corrected RB partial product.

The comparison of various PPG blocks for a 54x54-bit and 64x64-bit multiplier is shown in Table III. The number of partial products after the encoding is shown for each multiplier. Our PPG design exhibits the highest amount of reduction among 54x54bit multipliers. The details regarding the number of partial products for 54-bit multipliers was given in [5, 8, 10], whereas those for 64-bit multipliers were computed by us. It may be noted that in the case of 64-bit multipliers all the earlier multiplier formats exceed the optimum number of partial products for a 4 stage partial product accumulator.



64-bit multiplier architecture

IV. CONCLUSION

A new design for Booth multipliers has been proposed by removing the carry propagation delay introduced while generating the negative partial products in two's complement form. This is achieved by presenting the partial products in one's complement form together with an additional bit. This technique replaces the error correcting word in earlier designs with one error digit per RB operand, which can be added along with the RBA tree. The multiplier width can now be increased to perfect powers of 2, without increasing the number of stages of RBAs in the partial product accumulation stage. The use of radix-4 Booth encoding combined with our technique results in 78% reduction in the number of partial products generated. The selection of the particular RB encoding also allows us to take advantage of a faster RBA cell, thereby speeding up multiplication from all fronts.

V. REFERENCES

[1]. I Koren, Computer Arithmetic Algorithms, Prentice Hall, New York, 1993.

[2]. AD. Booth, A signed binary multiplication technique, Quarterly Journal of Mechanics and Applied Mathematics 4 (1951) 236–240.

[3]. O.L. McSorley, High-speed arithmetic in binary computers, Proceedings of the Institute of Radio Engineers 49 (1961) 67–91.

[4]. Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa, N. Takagi, A high-speed multiplier using a redundant binary adder tree, IEEE Journal of Solid-State Circuits 22 (1987) 28–34.

[5]. H Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, K. Mashiko, An 8.8–11 s 54 54-bit multiplier with high-speed redundant binary architecture, IEEE Journal of Solid-State Circuits 31 (1996) 773–783.

[6]. SM. Yen, C.S. Laih, C.H. Chen, J.Y. Lee, An efficient redundant-binary number to binary number converter, IEEE Journal of Solid-State Circuits 27 (1992) 109–112.

[7]. N Besli, R.G. Deshmukh, A novel redundant binary signed-digit (RBSD) Booth's encoding, in: Proceedings of the IEEE Southeast Conference, Columbia, SC, 2002, pp. 426–431.

[8]. N. Besli, R.G. Deshmukh, A 54 54-bit multiplier with a new redundant binary Booth's encoding, Proceedings of the Canadian Conference on Electrical and Computer Engineering 2 (2002) 597–602 (Winnipeg, Canada).

[9]. S. Lee, S. Bae, H. Park, A compact radix-64 54 54 CMOS redundant binary parallel multiplier, IEICE Transactions on Electronics E 85-C (2002) 1342–1350.

[10]. Y. Kim, B. Song, J. Grosspietsch, S. Gillig, A carry-free 54 54-bit multiplier using equivalent bit conversion algorithm, IEEE Journal of Solid-State Circuits 36 (2001) 1538–1545.