EXECUTION OF REDUNDANT BINARY MULTIPLIERS WITH MODIFIED PARTIAL PRODUCT GENERATOR

[1]K.SURESH, [2]R.VIJAY KUMAR

[1]M.TECH VLSID, DEPT OF E.C.E, KAKINADA INSTITUTE OF ENGINEERING AND TECHNOLOGY,KORANGI,ANDHRAPRADESH,INDIA,533461

[2]ASSOSIATE PROFESSOR, KAKINADA INSTITUTE OF ENGINEERING AND TECHNOLOGY,KORANGI,ANDHRAPRADESH,INDIA,533461

Abstract: Adders are the key 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.Due to its high modularity and carry-free addition, a redundant binary (RB) representation can be used when designing high performance multipliers. The conventional RB multiplier re -quires an additional RB partial product (RBPP) row, because an error-correcting word (ECW) is generated by both the radix-4 Modified Booth encoding (MBE) and the RB encoding. This incurs in an additional RBPP accumulation stage for the MBE multiplier. In this paper, a new RB modified partial product generator (RBMPPG) is proposed; it removes the extra ECW and hence, it saves one RBPP accumulation stage. Therefore, the proposed RBMPPG generates fewer partial product rows than a conventional RB MBE multiplier. Simulation results show that the proposed RBMPPG based designs significantly improve the area and power consumption when the word length of each operand in the multiplier is at least 32 bits.

Key words: Modified Booth encoding (MBE), Redundant binary (RB), RB modified partial product generator (RBMPPG), quality configurable.

I.INTRODUCTION

The digital multiplier is a ubiquitous arithmetic unit in microprocessors, digital signal processors, and emerging media processors. It is also a kernel operator in applicationspecific data path of video and audio codes, digital filters, computer graphics, and embedded systems. Compared with many other arithmetic operations, multiplication is timeconsuming and power hungry. The critical paths dominated by digital multipliers often impose a speed limit on the

entire design. Hence, VLSI design of high-speed multipliers, with low energy dissipation, is still a popular research subject. Redundant binary (RB) representation is one of the signed digit representations first introduced by Avizienis [9] in 1961 for fast parallel arithmetic.Many algorithms and architectures have been proposed to design high-speed and low-power multipliers [1-13]. A normal binary (NB) multiplication by digital circuits includes three steps. In the first step, partial products are generated; 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. Two methods have been us ed 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 [5-6]. Both methods allow the partial product reduction treeto be reduced at a rate of 2:1. The RB addition is carry-free, making it a promising substitute for two's complement multi-operand addition in a tree-structured multiplier. Similar to a normal binary (NB) multiplier, an RB multiplier is anatomized into three stages and consists of four modules: the Booth encoder, RB partial product generator (also known as 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 convention-al 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:

$$NRBPPAS = \log (N /4 + 1)$$

$$= n - 1, \text{ if } N = 2^n$$

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 RBPProws 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.

2. 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. [6] noticed that some hard multiples can be obtained by the differences of two sim-ple power-of-two multiplies. A new radix-16 Booth en-coding (RBBE-4) technique without ECW has been pro-posed in [4]; 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 [10] when requiring the same number of partial products.

3. RELATED STUDY

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 engaging RB multiplier designs; the designs are synthesized using the Nan Gate 45nm Open Cell Library. when the word length of each of the operands is at least 32 bits. This paper concentrates on the RBPP generator for outlining a 2 - bit RB multiplier with less incomplete item pushes by taking out the additional ECW. Another RB adjusted fractional item

generator in light of MBE (RBMPPG-2) is proposed. In the proposed RBMPPG-2, the ECW of each column is moved to its next neighbor push. Moreover, the additional ECW created by the last halfway item column is joined with both the two most huge bits (MSBs) of the main fractional item push and the two minimum huge bits (LSBs) of the last fractional item push by rationale rearrangements. In this manner, the proposed strategy decreases the quantity of RBPP lines from /4 + 1 to /4, i.e., a RBPP gathering stage is spared. The proposed technique is connected to 8×8-piece, 16×16-piece, 32×32-piece, and 64×64-piece RB multiplier plans; the outlines are combined utilizing the NanGate 45nm Open Cell Library. The proposed plans accomplish noteworthy decreases in territory what's more, power utilization contrasted and existing multipliers at the point when the word length of each of the operands is at minimum 32 bits. While a humble increment in postponement is experienced (around 5%), the power-postpone item (PDP) at word lengths of no less than 32 bits affirms that the proposed plans are the best likewise by this figure of legitimacy. This paper is composed as takes after. Area 2 presents radix-4 Booth encoding. The plan of the regular RBPP generator is additionally explored. Area 3 presents the proposed RBMPPG. This area likewise illustrates the reception of the proposed RBMPPG into different word-length RB multipliers. Segment 4 gives the assessment aftereffects of the new RB multipliers utilizing the proposed RBMPPG for various word lengths and looks at them to past best outlines found in the specialized writing. The conclusion is given in Section 5.

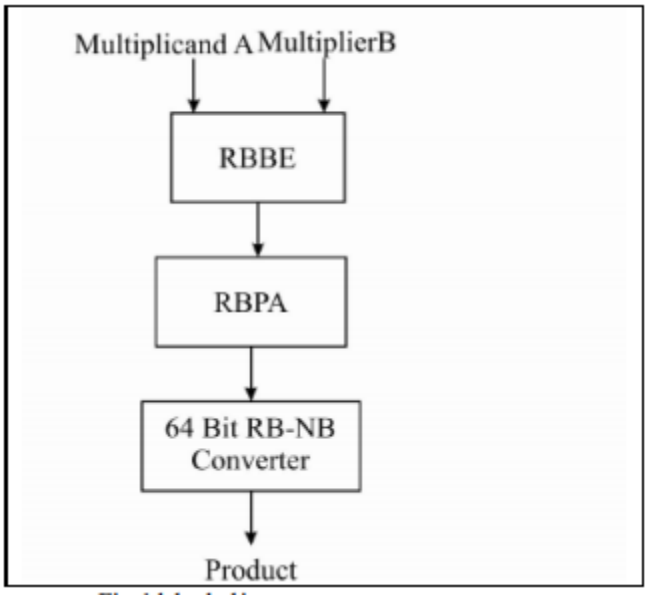
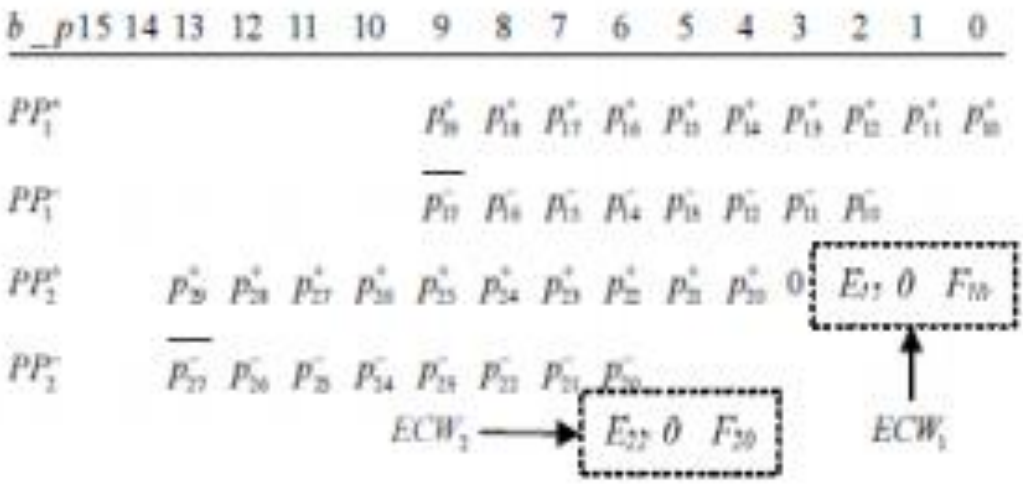


Fig.3.1. Block diagram.

IV.PROPOSED SYSTEM

Booth encoding has been proposed to facilitate the multiplication of two's complement binary numbers [7]. It was revised as modified Booth encoding (MBE) or radix-4 Booth encoding [8].The MBE scheme explained in the table, where A = aN-1 a N-2 …..a 2 a 1 a 0stands for the multiplicand, and B =b N-1 b N-2 ….b 2 b 1 b 0 stands the multiplier bits .The multiplier bits are grouped in set of three adjacent bits. The two side bits are overlapped with neighboring groups except the first multiplier bits group in which it is {b1, b0, 0}. Each group is decoded by selecting the partial product shown in Table I, where 2Aindicates twice the multiplicand, which can be obtained by left shifting. Negation operation is achieved by inverting each bit of A and adding '1' (defined as correction bit) to the LSB. Booth encoding has been proposed to facilitate the multiplication of two's complement binary numbers. It was revised as modified Booth encoding (MBE) or radix-4 Booth encoding.The MBE scheme explained in the table, where A = aN-1 a N-2 …..a 2 a 1 a 0stands for the multiplicand, and B =b N-1 b N-2 ….b 2 b 1 b 0 stands the multiplier bits. The multiplier bits are grouped in set of three adjacent bits. The two side bits are overlapped with neighboring groups except the first multiplier bits group in which it is {b1, b0, 0}. Each group is decoded by selecting the partial product shown in Table I, where 2Aindicates twice the multiplicand, which can be obtained by left shifting. Negation operation is achieved by inverting each bit of A and adding _1' (defined as correction bit) to the LSB [10-13]. Both MBE and RB coding schemes introduce errors and two correction terms are required: 1) when the NB number is converted to a RB format, -1 must be added to the LSB of the RB number; 2) when the multiplicand is multiplied by -1 or - 2 during the Booth encoding, the number is inverted and +1 must be added to the LSB of the partial product. A single ECW can compensate errors from both the RB encoding and the radix-4 Booth recoding.



In the second stage, a 4-stage RBA summing tree is used to sum 16 RB partial products. Each RBA block contains 64 RB full adder (RBFA) cells and a varying number of RB half adder (RBHA) cells depending on where it is located. The proposed RBMPPG-2 can be applied to any bit RB multipliers with a reduction of a RBPP accumulation stage compared with conventional designs. Although the delay of RMPPG-2 increases by 1-stage of TG delay, the delay of one RBPP accumulation stage is significantly larger than a 1- stage TG delay. Therefore, the delay of the entire multiplier is reduced. The improved complexity, delay and power consumption are very attractive for the proposed design. The multiplier consists of the proposed RBMPPG-2, three RBPP accumulation stages, and one RB-NB converter. Eight RBBE-2 blocks generate the RBPP they are summed up by the RBPP reduction tree that has three RBPP accumulation stages. Each RBPP accumulation block contains RB full adders (RBFAs) and half adders (RBHAs).

V. SIMULATION RESULTS

This paper presents a high performance 64x64 bit Redundant binary multiplier with modified redundant binary partial product generator, RB summing tree, and 128 RB to NB Converter. Multiplier designs are synthesized using Xilinx ISE 13.1 targeting a Xilinx Virtex-5 FPGA device XC5VLX110t (package 2ff1136) using randomly generated input patterns. Top View module, RTL Schematic View are mentioned in Fig. 2 & Fig. 3 and Design Summary report of designed multiplier are given in Fig 4. Improvement in area, delay, and power are verified by comparing both conventional and designed RB multipliers. with the application of partial product generation technique as proposed in [12], one less RB partial product accumulation stage is needed, which reduces the total number of RBAs required associated with addition process of

this RB partial product accumulation stage. However this technique [12] strikes with the modest increase in delay, which is compensated by RB-NB converter

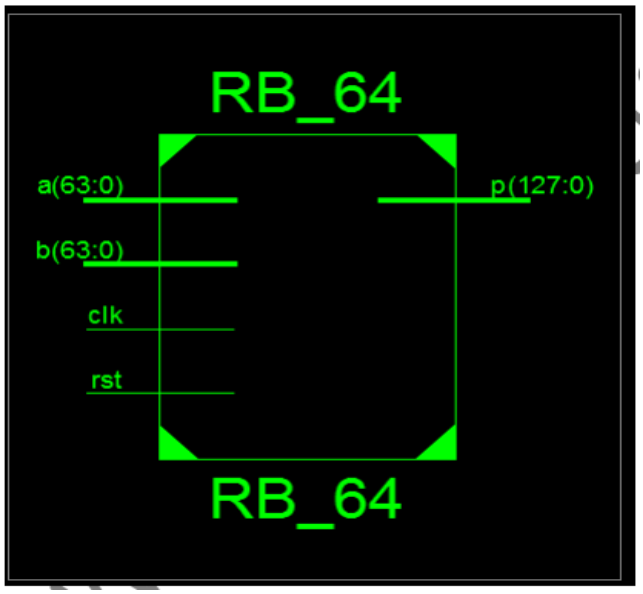


Fig.5.1. model diagram.

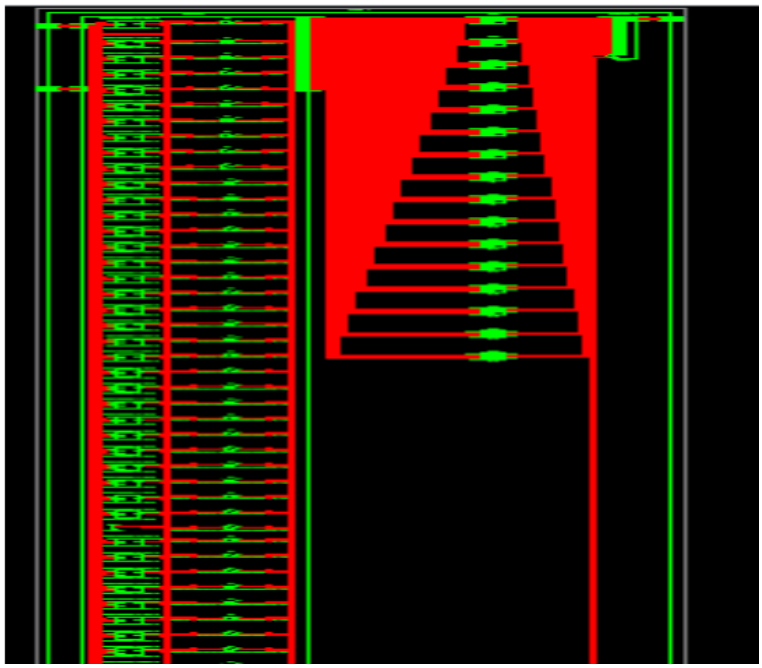


Fig.5.2. Rtl diagram.

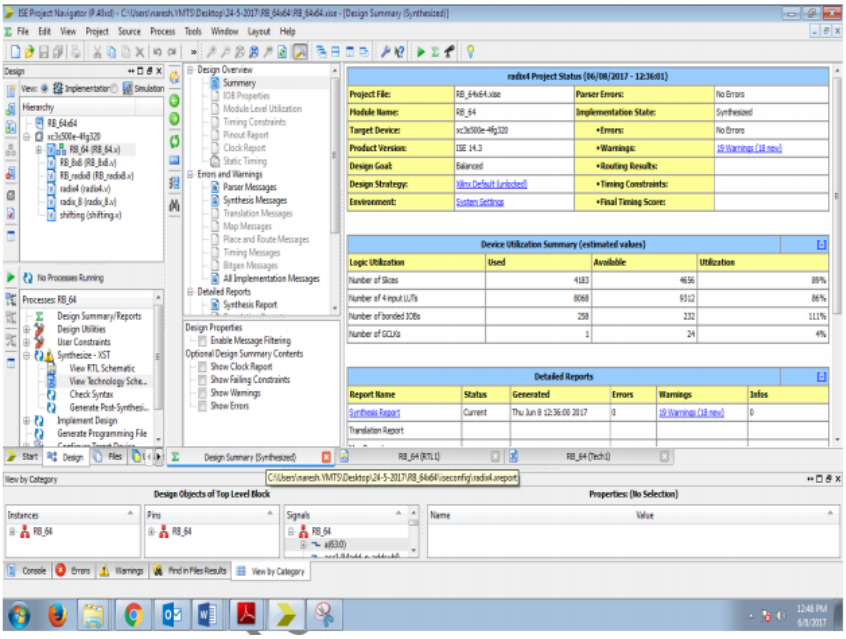


Fig.5.3. data summary.

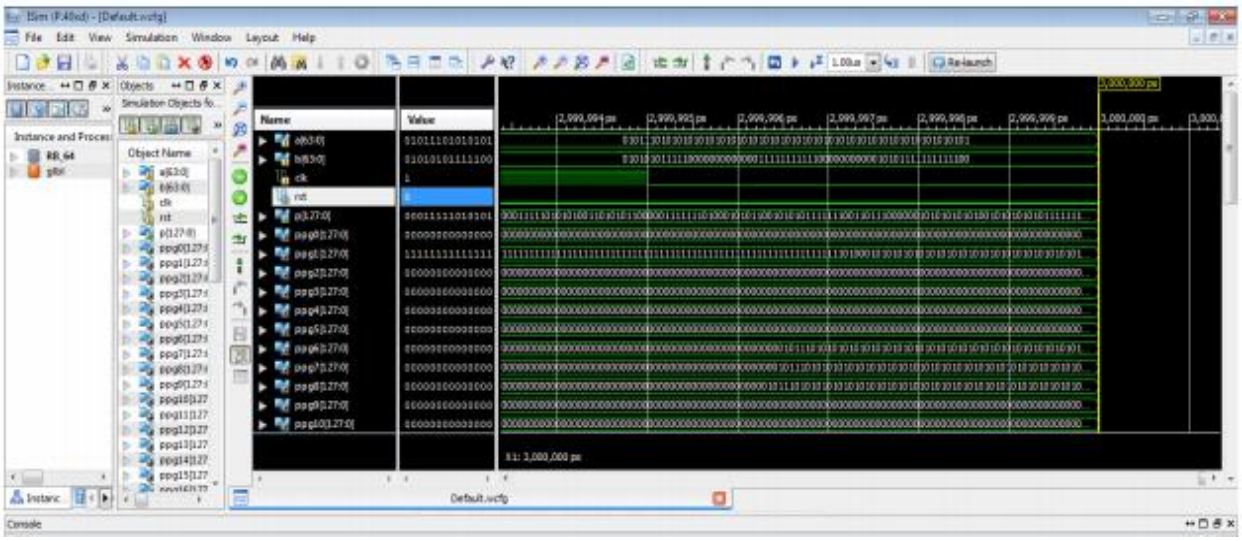


Fig.5.4. Simulation results.

Conclusion The high performance 64x64 bit RB multiplier architecture has been designed in this paper, by considering a tradeoff between area, power and delay. This design eliminates ECW that is introduced by the previous design. Therefore RBPP accumulation stage is saved. The new RB partial product generation technique can be applied to proposed design to reduce the number of RBPP rows from [N/4 +1] to [N/4]. An area efficient Modified RB partial product generator introduces modest delay in our architecture which has been minimized by means of delay

efficient RB-NB converter to some extent. The designed approach for 64x64 bit RBMBE multiplier shows improved performance over CRBMBE multiplier in terms of area and delay.

REFERENCES

[1] IEEE standard for floating–point arithmetic. IEEE Standards Committee, Oct. 2006.

[2] F. Y. Busaba, T. Slegel, S. Carlough, C. Krygowski, and J. G. Rell. The design of the fixed point unit for the z990 microprocessor. InProc. ACM Great Lakes14th Symposium on VLSI, pages 364– 367, Apr. 2004.

[3] M. F. Cowlishaw. Decimal floating-point: Algorism for computers. InProc. IEEE16th Symposium on Computer Arithmetic, pages 104– 111, July 2003.

[4] M. A. Erle and M. J. Schulte. Decimal multiplication via carry-save addition. InProc. IEEE Int'l Conference on Application-Specific Systems, Architectures, and Processors, pages 348–358, June 2003.

[5] M. A. Erle, E. M. Schwarz, and M. J. Schulte. Decimal multiplication with efficient partial product generation. In Proc. IEEE17th Symposium on Computer Arithmetic, pages 21–28, June 2005.

[6] R. D. Kenney and M. J. Schulte. High-speed multioperand decimal adders.IEEE Trans. on Computers, 54(8):953–963, Aug. 2005.

[7] R. D. Kenney, M. J. Schulte, and M. A. Erle. High-frequency decimal multiplier. InProc. IEEE Int'l Conference on Computer Design: VLSI in Computers and Processors, pages 26–29, Oct. 2004.

[8] T. Lang and A. Nannarelli. A radix-10 combinational multiplier. InProc.40th Asilomar Conference on Signals, Systems, and Computers, pages 313–317, Oct. 2006.

[9] R. H. Larson. High-speed multiply using four input carrysave adder. IBM Tech. Disclosure Bulletin, 16(7):2053–2054, Dec. 1973.

[10] N. Ohkubo and M. Suzuki. A 4.4 ns CMOS 54x54–bit multiplier using pass-transistor multiplexer. IEEE Journal of Solid State Circuits, 30(3):251–256, Mar. 1995.