iJRASET
International Journal For Research in
Applied Science and Engineering Technology

Volume: 5 Issue: VIII Month of publication: August 2017

DOI: http://doi.org/10.22214/ijraset.2017.8026

Modified Partial Product Generator for Redundant Binary Multiplier with High Modularity and Carry-Free Addition

Thoka. Babu Rao[1], G. Kishore Kumar[2]
[1], M. Tech in VLSI & ES, Student at Velagapudi Ramakrishna Siddhartha Engineering College, Jawaharlal Nehru Technological University -Kakinada, Kanuru, Vijayawada-7, Andhra Pradesh
[2],Assistant Professor, Electronics and Communication Engineering department, Velagapudi Ramakrishna Siddhartha Engineering College, Jawaharlal Nehru Technological University -Kakinada, Kanuru, Vijayawada-7, Andhra Pradesh

Abstract: Because of its high modularity and carry-free addition, a redundant binary (RB) representation can be used when designing high performance multipliers. The conventional RB multiplier needs for an additional RB partial product (RBPP) row, because an error-correcting word (ECW) is created by both the radix-8 and radix-4 Modified Booth encodings (MBE). This becomes subject in an additional RBPP accumulation stage for the MBE multiplier. A new RB modified partial product generator (RBMPPG) was proposed in this paper; it takes off the extra ECW and hence, it rescues one RBPP accumulation stage. Therefore, than a conventional RB MBE multiplier, the proposed RBMPPG produces fewer partial product rows. Simulation results show that the proposed RBMPPG based designs sufficiently make better the area and power consumption when the word length of each operand in the multiplier is at least 32 bits; these decreases over previous NB multiplier designs need in a small delay increase (approximately 5%). The power-delay product can be making smaller by up to 59% using the proposed RB multipliers when estimated with existing RB multipliers.

I. INTRODUCTION

Digital multipliers are extensively used in arithmetic units of microprocessors, multimedia and digital signal processors. Many algorithms and architectures [1-8] have been nominated to design high-speed and low power multipliers. A simple binary (NB) multiplication by digital circuits comprises three steps. In the first step, partial products are produced; in the second step, all partial products are added by a partial product reduction tree still two partial product rows be left over. In the third step, the two partial product rows are summed up by a fast carry propagation adder. Two methods have been used to carry out the second step for the partial product reduction. A second method uses redundant binary (RB) numbers[8-9], while a first method uses 4-2 compressors . Both methods let have the partial product reduction tree to be reduced at a rate of 2:1.

Avizienis has been making known the redundant binary number representation [1] to carry out signed-digit arithmetic; the RB number has the power to be represented in different ways. For redundant binary addition trees fast multipliers can be designed. The redundant binary representation has also been subjected to a floating-point processor and put into effect in VLSI. High performance RB multipliers have become popular due to the beneficial features, such as high modularity and carry-free addition[3-6].

A RB partial product (RBPP) generator, a RBPP reduction tree and a RB-NB converter forms a RB multiplier. In the partial product generator of parallel multipliers to reduce the number of partial product rows a Radix-4 Booth encoding or a modified Booth encoding (MBE) is generally used. A RBPP row can be secured from two adjacent NB partial product rows by inverting one of the pair rows; an N-bit conventional RB MBE (CRBBE-2) multiplier requires (N/4) RBPP rows. An extra error-correcting word (ECW) is also indispensable by both the RB and the Booth encoding; therefore, the number of RBPP accumulation stages (NRBPPAS) required by a power-of-two word-length (i.e.,2^n-bit) multiplier is given by:

$$\text{NRBPPAS}= [\ \log_2 (N/4+1)],$$
$$=n-1, \text{ if } N= 2^n.$$

If the extra ECW can be eliminated, an RBPP accumulation stage is store up for future use, so resulting in being improvement of complexity and critical path delay for a RB multiplier. For example, a conventional 32-bit RB multiplier has 4 RBPP accumulation stages; if the ECW is eliminated, then the number of RBPP accumulation stages is decreased to 3, i.e., the stage number is decreased by 25%. Note that the problem of extra ECW does not live in standard noteworthy size (i.e., 24×24-bit and 54×54-bit) RB multipliers as used in floating point-arithmetic units.

Another possibility, a high-radix Booth encoding technique can decrease the number of partial products. However, the number of extortionate hard multiples (i.e., a multiple that is not a power of two and the operation cannot be accomplished by simple shifting and/or complementation) increases too. Basil et al. became aware of that some hard multiples can be secured by the differences of two simple power-of-two multiplies. A new radix-8 Booth encoding (RBBE-4) skill without ECW has been proposed in; it keeps away from the matter of hard multiples. A radix-8 RB Booth encoder can be used to defeat the hard multiple problem and keep away from the extra ECW, but at the cost of doubling the number of RBPP rows. For that reason, the number of radix-8 RBPP rows is the identical as in the radix-4 MBE. However, the RBPP generator based on a radix-8 Booth encoding has a complex circuit structure and a lower speed estimated with the MBE partial product generator when necessary for the same number of partial products.

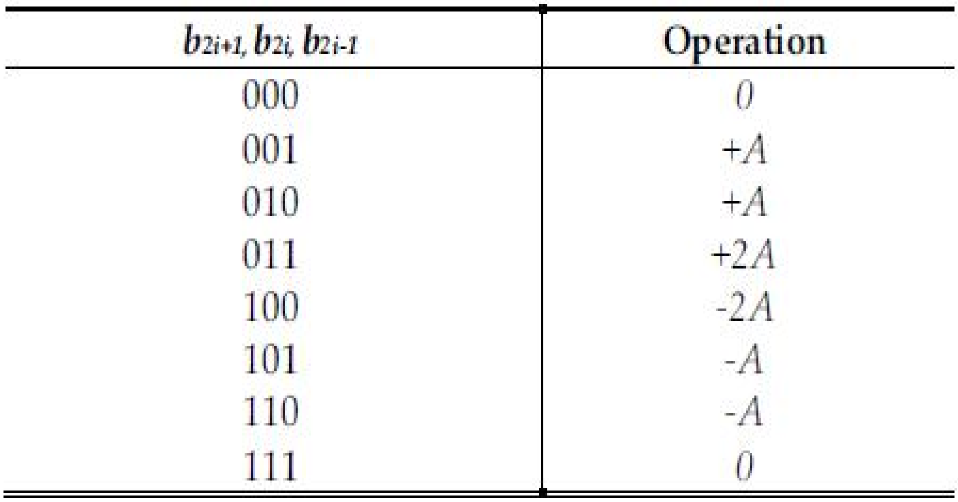
This paper pays particular attention on the RBPP generator for designing a 2^n-bit RB multiplier with fewer partial product rows by removing the extra ECW. A new RB modified partial product generator with reference to MBE (RBMPPG-2) is proposed. In the proposed RBMPPG-2, the ECW of each row is changed to its next neighbor row. In addition, the extra ECW generated by the last partial product row is joined 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. Consequently, the proposed method decreases the number of RBPP rows from (N/4 + 1) to (N/4), i.e., a RBPP accumulation stage is stored up. The proposed method is exerted to 8×8-bit, 16×16-bit, 32×32-bit, and 64×64-bit RB multiplier designs; the designs are synthesized using the Nan Gate 45nm Open Cell Library. The proposed designs successfully reach significant reductions in area and power consumption estimated with existing multipliers when the word length of each of the operands is at least 32 bits. While a modest increase in delay is experienced (approximately 5%), the power-delay product (PDP) at word lengths of at least 32 bits makes formally valid that the proposed designs are the best also by this figure of merit.

II. REVIEW OF BOOTH ENCODING AND RB PARTIAL PRODUCT GENERATOR

A. *Radix-4 Booth Encoding*

To make easy the multiplication of two's complemented binary numbers Booth encoding has been proposed. It was reconsidered as modified Booth encoding (MBE) or radix- 8 Booth encoding. The MBE scheme is summed up in Table I, where $A = a_{N-1}\, a_{N-2}\, a_{N-3}$..........$a_3\, a_2\, a_1\, a_0$ stands for the multiplicand, and $B = b_{N-1}\, b_{N-2}$ ....$b_3\, b_2\, b_1\, b_0$ stands for the multiplier. The multiplier bits are grouped in sets of three adjacent bits. Except the first multiplier bits group in which it is {b1, b0, 0} the two side bits are overlapped with neighboring groups. As shown in Table I each group is decoded by selecting the partial product, where 2A indicates twice the multiplicand, which can be secured by left shifting. By inverting each bit of A and adding '1' negation operation is successfully reached (defined as correction bit) to the LSB [7-8]. Methods have been proposed to solve the problem of correction bits for NB radix-8 Booth encoding (NBBE-2) multipliers. Regardless of how, this problem has not been solved for RB MBE multipliers.

TABLE I: Mbe Shceme



$b_{2i+1}, b_{2i}, b_{2i-1}$	Operation
000	0
001	+A
010	+A
011	+2A
100	-2A
101	-A
110	-A
111	0

B. *Rb Partial Product Generator*

As one RB digit is represented by two bits, then a RBPP is generated with two NB partial products. Using two's complement representation the addition of two N-bit NB partial products X and Y can be expressed as follows :

$$\begin{aligned} X + Y &= X - \overline{Y} - 1 \\ &= \left(-x_N 2^N + \sum_{i=0}^{N-1} x_i 2^i\right) - \left(-\overline{y_N} 2^N + \sum_{i=0}^{N-1} \overline{y_i} 2^i\right) - 1 \\ &= -(x_N - \overline{y_N}) 2^N + \sum_{i=0}^{N-1} (x_i - \overline{y_i})\, 2^i - 1 \\ &= (X, \overline{Y}) - 1 \qquad (2) \end{aligned}$$

Errors and two correction terms are introduced by Both MBE and RB coding schemes required: 1) when the NB number is changed the form 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. The conventional partial product architecture of an 8-bit MBE multiplier. Where b_{-p} represents the bit position, p^+_{ij} or p^-_{ij} and is generated by using an encoder and decoder. An N-bit CRBBE-2 multiplier includes N/4 RBPP rows and one ECW; the ECW takes the form as follows:

$$ECW=E_{(N/4)2}0\ F_{(N/4)0}.......\ 0\ E_{i2}\ 0\ F_{i0}\ ....\ 0\ E_{12}\ 0\ F_{10} \qquad (3)$$

One extra RBPP accumulation stage is required for a 2^n-bit CRBBE-2 multiplier due to the ECW. There are 5 RBPP accumulation stages for a 64- bit RB multiplier; therefore, the number of RBPP accumulation stages can be decreased by 20% when get rid of the ECW in a 64-bit RB multiplier, which becomes better both the complexity and the critical path delay.

III. PROPOSED RB PARTIAL PRODUCT GENERATOR

Based on MBE (RBMPPG-2) a new RB modified partial product generator is introduced in this section; in this design, by incorporating ECW into both the two MSBs of the first partial product row (PP^+_1) and the two LSBs of the last partial product row ($PP_{(N/4)}$) it is eliminated.

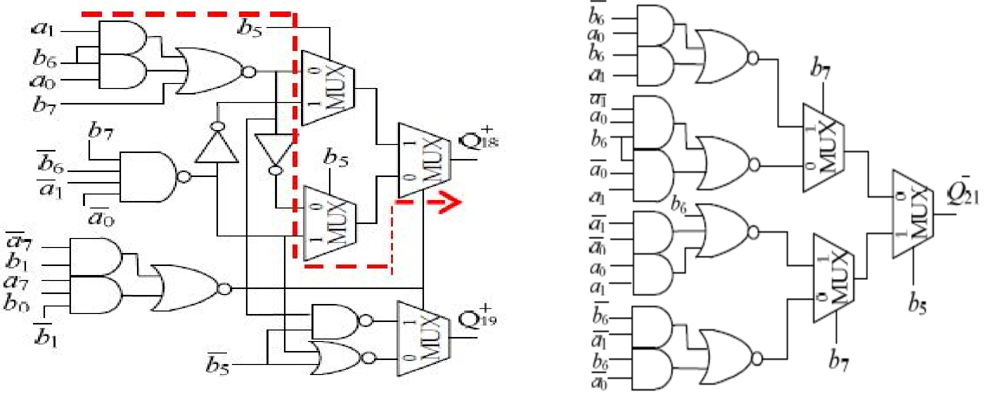
A. Design of RBMPPG-2-based High-Speed RB Multipliers

Compared with conventional designs the proposed RBMPPG-2 can be applied to any 2^n bit RB multipliers with a reduction of a RBPP accumulation stage. In spite of the delay of RMPPG-2 increases by 1-stage of TG delay, the delay of 1-stage TG delay is significantly lesser than one RBPP accumulation stage. Therefore, the delay is reduced in the entire multiplier. The achieved complexity, delay and power consumption are very attractive for the proposed design.

The proposed RBPP generator a 32-bit RB MBE multiplier using is shown in Fig. 1. The multiplier contains the proposed RBMPPG-2, three RBPP accumulation stages, one RB-NB converter and eight RBBE-2 blocks generate the RBPP (p^+_i, p^-_i); they are added up by the

RBPP reduction tree which contains three RBPP accumulation stages. Each RBPP accumulation block has RB full adders (RBFAs) and half adders (RBHAs) [4]. The 64-bit RB-NB converter which uses a hybrid parallel prefix/ carry select adder, converts the final accumulation results into the NB representation, [11] (as one of the most efficient fast parallel adder designs).

In a conventional 32-bit RB MBE multiplier architecture there are 4 stages; however, by using the proposed RBMPPG-2, the number of RBPP accumulation stages is decreased from 4 to 3. As well as power consumption, there are significant savings in delay and area. The improvements in delay, area and power consumption are further indicated in the next section by simulation.



(a) (b)

Fig1. The circuit diagram of the modified partial product variables: a) Q^+_{18} and Q^+_{19}, b) $Q^+_{20.}$

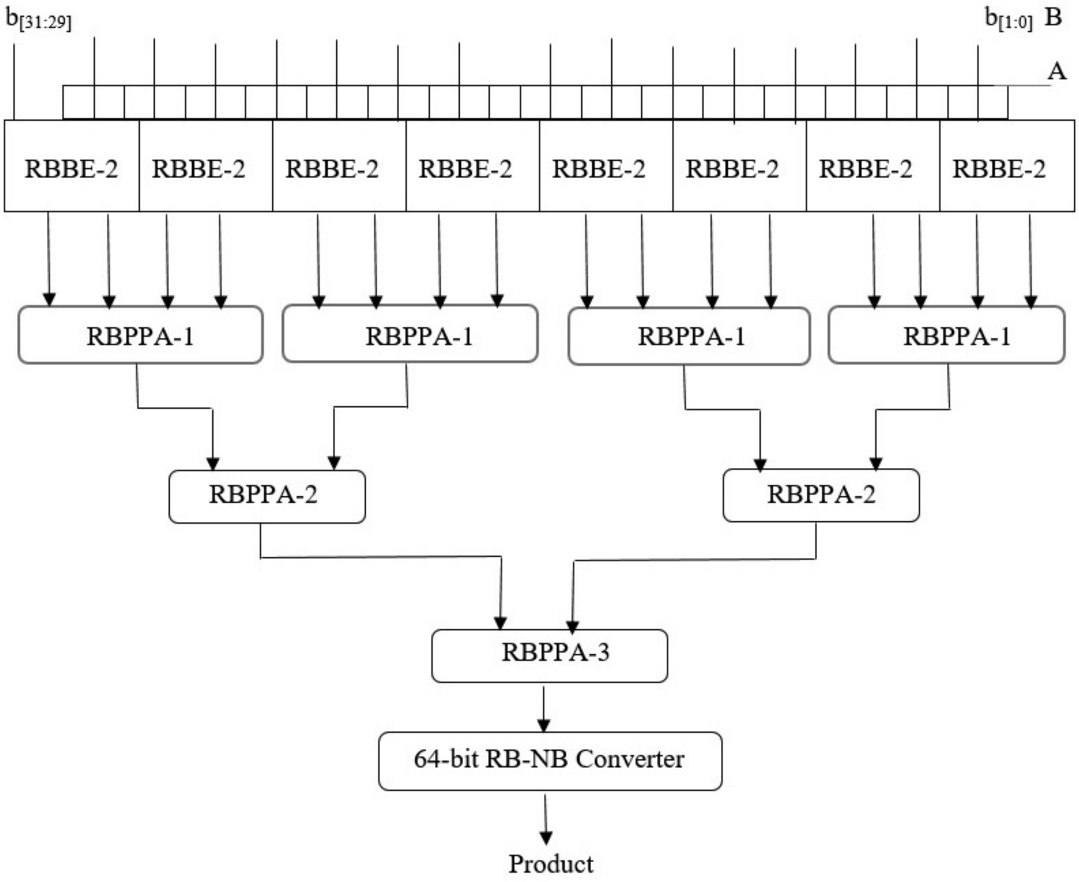


Fig2. The block diagram of a 32-bit RB multiplier using the proposed RBMPPG-2.

The number of RBPP accumulation stages in different 2^n -bit RB multipliers, i.e., 8×8-bit, 16×16-bit, 32×32-bit, 64×64-bit multipliers were compared in below table.

TABLE II

Rbpp Accumulation Stages Comparison In Rbpp Reduction Tree

Methods	64×64	32×32	16×16	8×8
CRBBE-2	5	4	3	2
RBBE-4	4	3	2	1
Proposed	4	3	2	1

The proposed design has 4 RBPP accumulation stages for a 64-bit multiplier; compared with CRBBE- 2 multipliers, it reduces the partial product accumulation delay time by 20%. However both the proposed design and RBBE-4 have the same number of RBPP accumulation stages, RBBE-4 is more intricate, because it uses radix-16 Booth encoding.

B. A High-Speed Multiplier Using a Redundant Binary Adder Tree

An array multiplier and a multiplier using a Wallace tree are well-known for their high-speed multiplication. The former has a good repeatability of unit cells and is favorable for VLSI implementation. Therefore many multipliers have been fabricated using this method. However, the n-bit multiplication time is proportional to n, and this method requires a long computation time for larger n‘s. On the other hand, in the latter, the n-bit multiplication time is proportional to $\log_2 n$. Thus the multiplication time increases slowly for larger n‘s. However, the physical design is rather difficult because of the complex interconnection in an LSI chip. So there are only a few multipliers based on this method.

The advantages of high-speed multiplication and easy physical design are presented by the multiplier using the new algorithm. On the other hand, the number of signal lines in this multiplier is about twice as many as that of conventional multipliers. Therefore we paid special interest to signal wiring during the physical design of the multiplier.

IV. ALGORITHM OF THE MULTIPLIER

A. The Multiplication Time based on the New a algorithm

A redundant binary representation which is one of the signed digit representations proposed by Avizienis is internally used by this multiplier. In the n-bit multiplier based on the new algorithm as shown in Fig, n partial products are added up pair wise in a $\log_2$ n level binary adder tree.

Addition of two n-digit redundant binary numbers is performed in parallel in a constant time. In a time proportional to $\log_2$ n the summation of the partial products is performed. As conversion of the number into a two's complement binary number is required so that the sum is represented in a redundant binary number. A carry look ahead adder with a time proportional to $\log_2$n is used for this conversion. As a result, n-bit multiplication is carried out in a time proportional to $\log_2$ n.

B. Redundant Binary Representation Additions

Addition of two numbers in a binary number system requires a computation time not less than proportional to the logarithm of the word length of the operands, by reason of carry propagation. Regardless of how, in a redundant binary number system, addition of two numbers can be performed in a constant time independent of the word length, since carry propagation can be completely removed. This is one of the features of our multiplier.

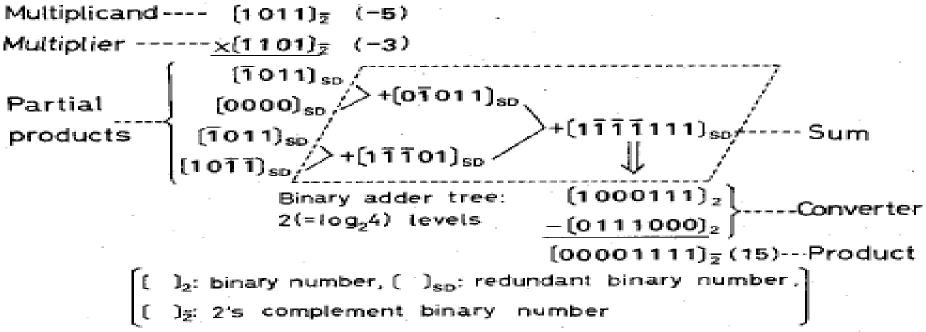

Multiplicand---- [1011]₂̄ (-5)
Multiplier ------x[1101]₂̄ (-3)
Partial products
[1̄011]SD
[0000]SD
[1̄011]SD
[101̄1̄]SD
+[01̄011]SD
+[11̄1̄01]SD
+[11̄1̄1̄111]SD----- Sum
Binary adder tree: 2(=log₂4) levels
[1000111]₂
-[0111000]₂
-----Converter
[00001111]₂̄ (15)---Product
[ ]₂: binary number, [ ]SD: redundant binary number,
[ ]₂̄: 2's complement binary number

Fig3. Example of multiplication by use of a redundant binary adder tree.

The redundant binary representation has a digit set {0, 1, 1} which is used in our multiplier, where l denotes –l. In a constant time utilizing the redundancy, addition of two n-digit redundant binary numbers is performed. This addition is carried out in the following two steps.

In the first step, an augends and an addend are added by two digits in the same figure positions. In addition there are six types of combinations of two digits as shown in Table I. The results of addition are two signals: intermediate carry digit C_i and intermediate sum digit Si defined as listed in this table.

TABLE III

Rules for the addition in first step without carry propagation

Type no	Combination	Next lower order position	Carry C_i	Sum S_i
1	{T,1}	-------------------	1	0
2	{1,0}	Neither are negative	1	T
		Otherwise	0	1
3	{1,T}	----------------------	0	0
4	{0,0}	----------------------	0	0
5	{0,T}	Neither are negative	0	T
		Otherwise	T	1
6	{T,T}	----------------------	T	0

In The second step the sum digit Z_i is obtained at each position by adding S_i and C_i from the next lower order position. After the first step, at any position in the second step there is no carry generation.

Fig shows an example of addition in accordance with the above rule. Therefore addition of two n-digit redundant binary numbers is performed in a constant time.

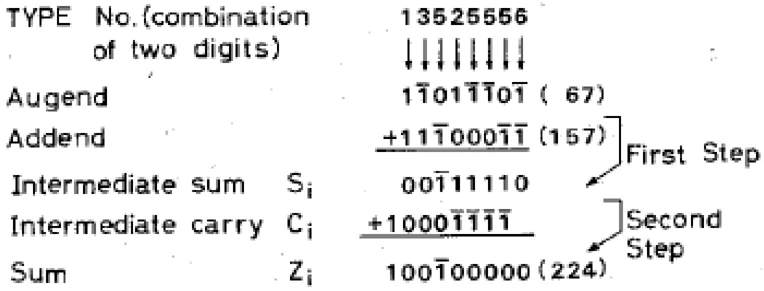

TYPE No.(combination of two digits)
13525556
Augend
$1\bar{1}01\bar{1}\bar{1}0\bar{1}$ (67)
Addend
$+11\bar{1}000\bar{1}\bar{1}$ (157)
First Step
Intermediate sum S_i
$00\bar{1}11110$
Intermediate carry C_i
$+1000\bar{1}\bar{1}\bar{1}\bar{1}$
Second Step
Sum Z_i
$100\bar{1}00000$ (224)

Fig4. Example of addition without carry propagation.

C. A Redundant Binary Number to a Two's Complement Binary Number conversion

Because in the external representation of a multiplier two's complement numbers are used, the conversion number must be done between a redundant binary number and a two's complement binary. This conversion can be done easily as follows. The conversion performed by changing the most significant digit 1 to $\bar{1}$. As an n-digit redundant binary number

$$A\left(=\sum_{i=0}^{n-1} a_i*2^i, \qquad a_i\in\{0,1,\bar{1}\}\right)$$

(4) is equal to

$$A^+\left(=\sum_{a_i=1} a_i*2^i\right)-A^-\left(=\sum_{a_i=\bar{1}}(-a_i)*2^i\right)$$

(5)

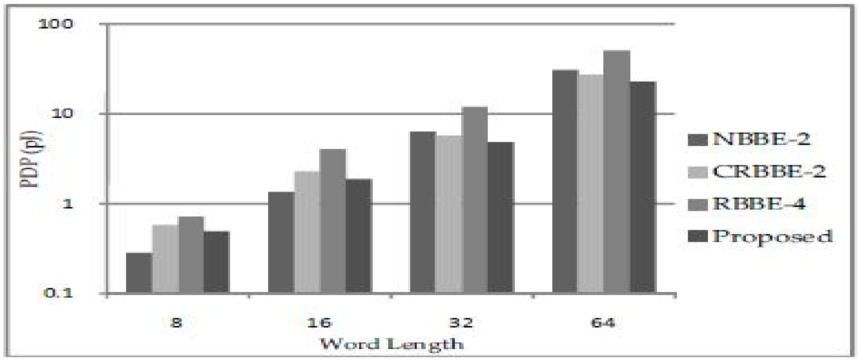


Fig4. PDP comparison of the NB and RB MBE multipliers at different word-lengths.

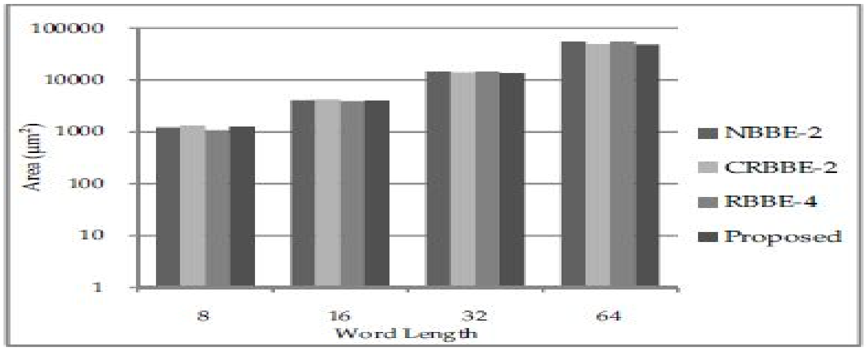


Fig5. Area comparison of the NB and RB MBE multipliers at different word-lengths.

V. CONCLUSIONS

In this paper a new modified RBPP generator has been proposed; this design removes the additional ECW that is inserted by previous designs. Consequently, due to the elimination of ECW a RBPP accumulation stage is saved. The new RB partial product generation technique can be applied to any 2^n- bit RB multipliers to reduce the number of RBPP rows from (N/4 + 1) to (N/4). Simulation results have shown that the performance of RB MBE multipliers using the proposed RBMPPG-2 is improved significantly in terms of delay and area. The proposed designs achieve significant reductions in area and power consumption when the word length is at least 32 bits. The PDP can be reduced by up to 59% using the proposed RB multipliers when compared with existing RB multipliers. Hence, the proposed RBPP generation method is a very useful technique when designing area and PDP efficient power-of-two RB MBE multipliers.

REFERENCES

[1] Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa, and N. Takagi, “A high speed multiplier using a redundant binary adder tree,” IEEE J. Solid-State Circuits, vol. SC-22, pp.28-34, 1987.

[2] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, “A 33 MFLOPS floating point processor using redundant binary representation,” in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), pp. 152–153, 1988.]

[3] H. Makino, Y. Nakase, and H. Shinohara, “A 8.8-ns 54x54-bit multiplier using new redundant binary architecture,” in Proc. Int. Conf. Comput. Design (ICCD), pp. 202-205, 1993.

[4] Y. Kim, B. Song, J. Grosspietsch, and S. Gillig, “A carry-free 54b×54b multiplier using equivalent bit conversion algorithm,” IEEE J. Solid-State Circuits, vol. 36, pp. 1538–1545, 2001.

[5] Y. He and C. Chang, “A power-delay efficient hybrid carrylookahead carry-select based redundant binary to two’s complementconverter,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, pp. 336–346, 2008

[6] G. Wang and M. Tull, “A new redundant binary number to 2'scomplement number converter,” in Proc. Region 5 Conference: Annual Technical and Leadership Workshop, pp. 141-143, 2004

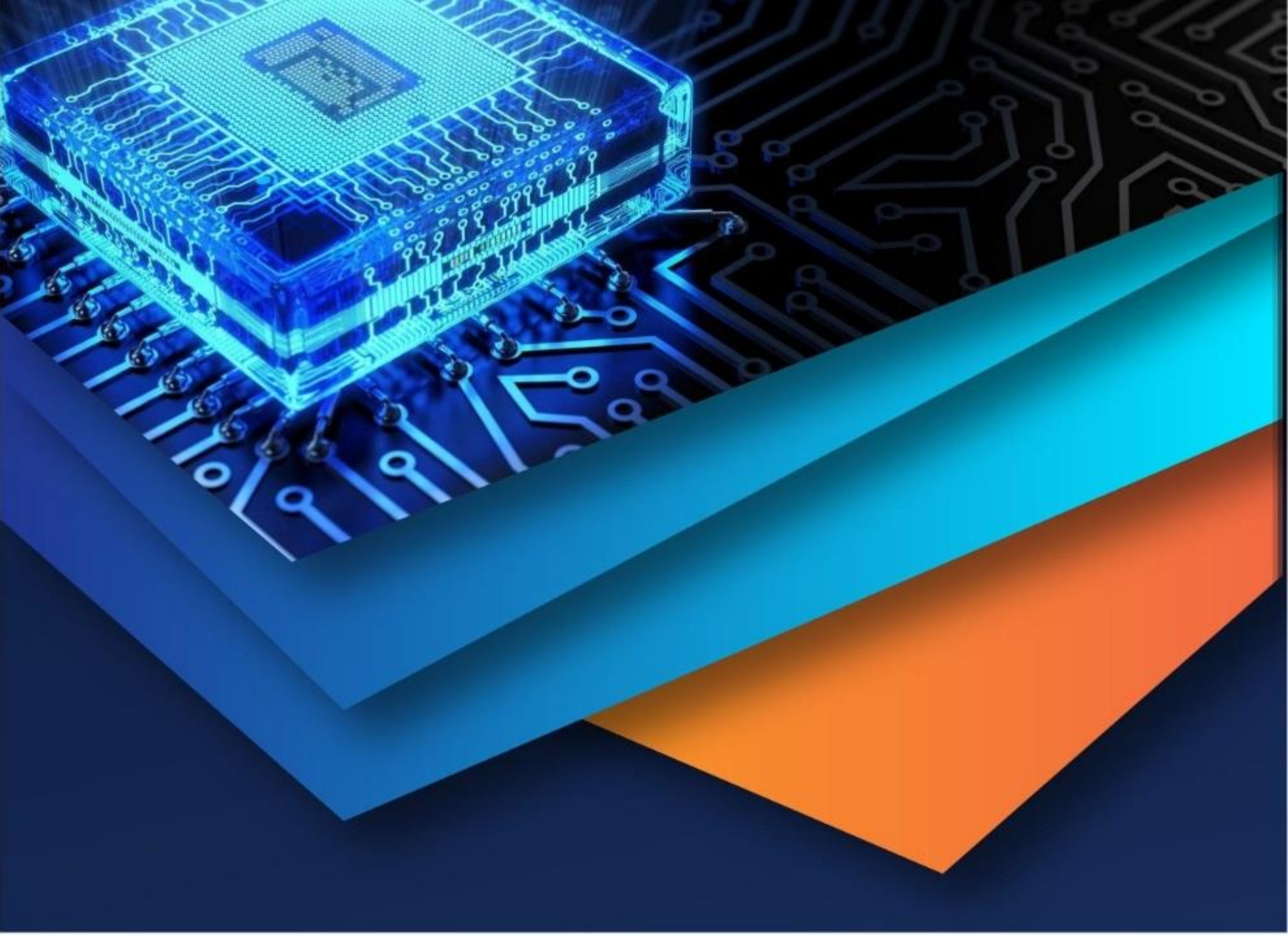
[7] W. Yeh and C. Jen, “High-speed Booth encoded parallel multiplier design,” IEEE Trans. Computers, vol. 49, pp. 692-701, 2000.

[8] J. Kang and J. Gaudiot,“A simple high-speed multiplier design,” IEEE Trans. Computers, vol. 55, pp.1253-1258, 2006.

[9] Y. He and C. Chang, “A new redundant binary Booth encoding for fast 2^n-bit multiplier design,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, pp. 1192–1199, 2009.

[10] O. MacSorley, “High-speed arithmetic in binary computers,” IRE Proc., vol. 49, pp. 67–91, 1961.

[11] G. Dimitrakopoulos and D. Nikolos, “High-speed parallelprefix VLSI Ling adders,” IEEE Trans. Computers, vol. 54, pp. 225-231, 2005.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY