

# Verilog Implementation of 64-bit Redundant Binary Product generator using MBE

Santosh Kumar G.B<sup>1</sup>, Mallikarjuna A<sup>2</sup>

M.Tech (D.E), Dept. of ECE, BITM, Ballari, India<sup>1</sup>

Assistant professor, Dept. of ECE, BITM, Ballari, India<sup>2</sup>

## 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 requires 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.

**Keywords**— Modified Booth encoding (MBE), Redundant binary (RB), RB modified partial product generator (RBMPPG), quality configurable.

## 1. 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 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 consuming 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 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 [5-6]. Both methods allow the partial product reduction tree to 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 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.,  $2^n$ -bit) multiplier is given by:  $NRBPPAS = \log_2(N/4 + 1) = n - 1$ , if  $N = 2^n$ .

This paper focuses on the RBPP generator for designing a 2<sup>n</sup>-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  $N/4 + 1$  to  $N/4$ , 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.

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.

This paper concentrates on the RBPP generator for outlining a 2<sup>n</sup>-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.

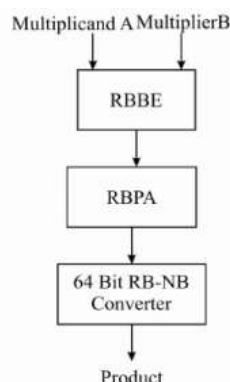
In this manner, the proposed strategy decreases the quantity of RBPP lines from  $N/4 + 1$  to  $N/4$ , i.e., a RBPP gathering stage is spared. The proposed technique is connected to 8×8-piece, 16×16-piece, 32×32-piece, and 64×64-piece RB multiplier plans; the outlines are combined utilizing the NanGate 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.

## 2. METHODOLOGY



**Fig-1:** Generic block diagram of Redundant binary multiplier

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 performed by simple shifting and/or complementation) increases too [14-16]. Besliet al. [16] noticed that some hard multiples can be obtained by the differences of two simple power-of-two multiplies. A new radix-16 Booth encoding (RBBE-4) technique without ECW has been proposed in [14]; 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.

### RADIX -4 BOOTH ENCODING

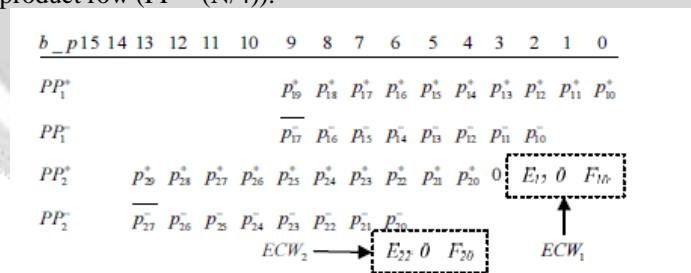
Booth encoding has been proposed to facilitate the multiplication of two's complement binary numbers [17]. It was revised as modified Booth encoding (MBE) or radix-4 Booth encoding [18]. The MBE scheme explained in the table, where  $A = a_{N-1} a_{N-2} \dots a_2 a_1 a_0$  stands for the multiplicand, and  $B = b_{N-1} b_{N-2} \dots b_2 b_1 b_0$  stands for 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  $\{b_1, b_0, 0\}$ . Each group is decoded by selecting the partial product shown in Table I, where  $2A$  indicates 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].

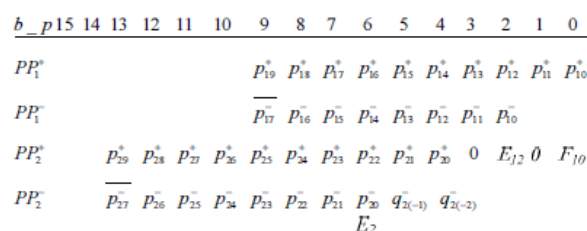
Both MBE and RB coding schemes introduce errors and two correction terms are required: 1) when the NB number is converted to a RB format, -1 must be added to the LSB of the RB number; 2) when the multiplicand is multiplied by -1 or -2 during the Booth encoding, the number is inverted and +1 must be added to the LSB of the partial product. A single ECW can compensate errors from both the RB encoding and the radix-4 Booth recoding.

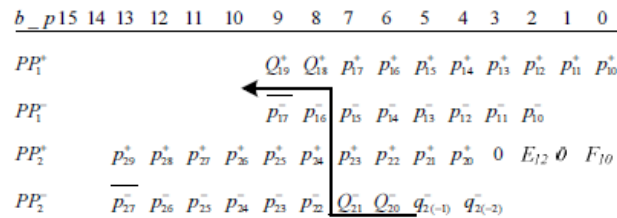
### 3. A PROPOSED RB PARTIAL PRODUCT GENERATOR

A new RB modified partial product generator based on MBE (RBMPPG-2) is presented in this section; in this design, ECW is eliminated by incorporating it into both the two MSBs of the first partial product row ( $PP_1 +$ ) and the two LSBs of the last partial product row ( $PP - (N/4)$ ).

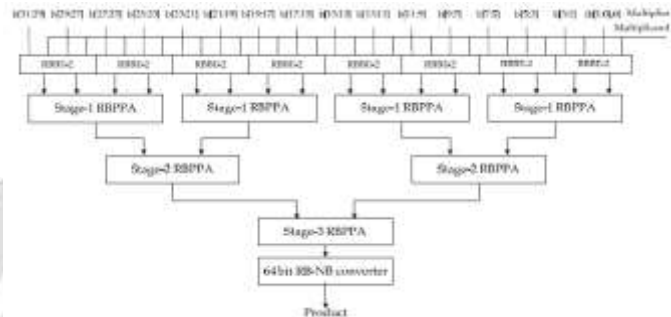
**Fig-2:** The first new RBMPPG-2 architecture for an 8-bit MB multiplier

It differs from conventional type by its error correcting vector. In this type error correcting vectors  $ECW_1$  is generated by  $PP_1$  and  $ECW_2$  is generated by  $PP_2$ .  $ECW_1 = 0$  to eliminate a RBPP accumulation,  $ECW_2$  needs to be incorporated into  $PP_1$  and  $PP_2$ .  $F_{20} b_3 = 000, 001, 010, 011, 111$   $F_{20} = \{0, b_5 b_4 b_3 = 100, 101, 110$

**Fig-3:** Revised RBMPPG by replacing E 22 and F 20



**Fig-4:** Final proposed RBMPPG by totally eliminating ECW



**Fig-5:** The block diagram of a 32-bit RB multiplier using the proposed RBMPPG-2.

The proposed RBMPPG-2 can be applied to any 2-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. A 32-bit RB MBE multiplier using the proposed RBPP generator is shown in Fig. 6. 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) [7].

The 64-bit RB-NB converter converts the final accumulation results into the NB representation, which uses a hybrid parallel prefix/ carry select adder [25] (as one of the most efficient fast parallel adder designs).

There are 4 stages in a conventional 32-bit RB MBE multiplier architecture; however, by using the proposed RBMPPG-2, the number of RBPP accumulation stages is reduced from 4 to 3 (i.e., a 25% reduction). These are significant savings in delay, area as well as power consumption. The improvements in delay, area and power consumption are further demonstrated in the next section by simulation.

Table V compares the number of RBPP accumulation stages in different 2-bit RB multipliers, i.e., 8×8-bit, 16×16-bit, 32×32-bit, 64×64-bit multipliers. For a 64-bit multiplier, the proposed design has 4 RBPP accumulation stages; it reduces the partial product accumulation delay time by 20% compared with CRBBE-2 multipliers.

Although both the proposed design and RBBE-4 have the same number of RBPP accumulation stages, RBBE-4 is more complex, because it uses radix-16 Booth encoding [14].  $Q_{19} + , Q_{18} + , Q_{21} - , Q_{20} -$  are used to represent the modified partial products.

By setting  $PP_{2+}$  to all ones and adding +1 to the LSB of the partial product,  $F_{20}$  can then be determined only by  $b_5$   $F_{20} = \{-1, b_5 = 0 \ F_{20} = \{0, b_5 = 1 \}$  As -1 can be coded as 111 in RB format,  $E_{22}$  and  $F_{20}$  can be represented by  $E_{22}, q_{2(-2)} - , q_{2(-1)} -$  as follows



$$E_2 = \begin{cases} E_{22}, & F_{20} = 0 \\ E_{22} - 1, & F_{20} = -1 \end{cases}$$

$$q_{2(-2)} = q_{2(-1)} = \begin{cases} 0, & F_{20} = 0 \\ 1, & F_{20} = -1 \end{cases}$$

This is further explained by the truth table of E22, F 20 and E2, q 2 - (-2), q2 - (-1).

TRUTH TABLE OF  $E_2, q_{2(-2)}, q_{2(-1)}$  AND  $p_{21}, p_{20}$

$b_7 b_6 b_5$	$E_{22} F_{20}$	$E_2 q_{2(-2)} q_{2(-1)}$	$p_{21}$	$p_{20}$
0 0 0	0 1	1 1 1	0	0
0 0 1	0 0	0 0 0	$a_1$	$a_0$
0 1 0	0 1	1 1 1	$a_1$	$a_0$
0 1 1	0 0	0 0 0	$a_0$	0
1 0 0	1 1	0 1 1	$\overline{a_0}$	1
1 0 1	1 0	1 0 0	$\overline{a_1}$	$\overline{a_0}$
1 1 0	1 1	0 1 1	$\overline{a_1}$	$\overline{a_0}$
1 1 1	0 0	0 0 0	0	0

The relationships between Q 19 + ,Q 18 + ,Q 21 - ,Q20 - and P19 + ,P21 - ,P20 - are summarized in table.

THE TRUTH TABLE OF  $Q_{19}^+, Q_{18}^+, Q_{21}^-, Q_{20}^-$

$p_{19}^+ p_{18}^+ p_{21}^- p_{20}^-$	$Q_{19}^+ Q_{18}^+ Q_{21}^- Q_{20}^-$ ( $E_2=0$ )	$Q_{19}^+ Q_{18}^+ Q_{21}^- Q_{20}^-$ ( $E_2=1$ )	$Q_{19}^+ Q_{18}^+ Q_{21}^- Q_{20}^-$ ( $E_2=-1$ )
0100	0100	0101	0011
0101	0101	0110	0100
0110	0110	0111	0101
0111	0111	1000	0110
1000	1000	1001	0111
1001	1001	1010	1000
1010	1010	1011	1001
1011	1011	1100	1010

Therefore, the extra ECW N/4 is removed by the transformation of 4 partial product variables and one partial product row is saved in RB multipliers with any power-of-two word-length. 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).

#### 4. CONCLUSIONS

A new modified RBPP generator has been proposed in this paper; this design eliminates the additional ECW that is introduced by previous designs. Therefore, a RBPP accumulation stage is saved due to the elimination of ECW. The new RB partial product generation technique can be applied to any 2n -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

#### 5. ACKNOWLEDGMENT

I express my sincere gratitude to my guide, Mallikarjuna A ,Dept. of Electronics and Communication Engineering, BITM, Ballari Institute of Technology, Ballari, Karnataka for the consistence support in successful completion of this paper.

## 6. REFERENCES

- [1] A. Avizienis, "Signed-digit number representations for fastparallel arithmetic," IRE Trans. Electron. Computers, vol. EC-10, pp. 389–400, 1961.
- [2] N.Takagi, H. Yasuura, and S. Yajima, "High-speed VLSI multiplication algorithm with a redundant binary addition tree," IEEE Trans. Computers, vol. C-34, pp. 789-796, 1985.
- [3] Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa, and N.Takagi, "A high speed multiplier using a redundant binaryadder tree," IEEE J. Solid-State Circuits, vol. SC-22, pp. 28-34, 1987.
- [4] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, "A33 MFLOPS floating point processor using redundant binary representation," in Proc. IEEE Int. Solid-State Circuits Conf.(ISSCC), pp. 152–153, 1988.
- [5] H. Makino, Y. Nakase, and H. Shinohara, "A 8.8-ns 54x54-bitmultiplier using new redundant binary architecture," in Proc.Int. Conf. Comput. Design (ICCD), pp. 202-205, 1993.
- [6] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Makino, "An 8.8-ns 54×54-bit multiplier with highspeed redundant binary architecture," IEEE J. Solid-State Circuits, vol. 31, pp. 773-783, 1996.
- [7] Y. Kim, B. Song, J. Grosspietsch, and S. Gillig, "A carry-free54b×54b multiplier using equivalent bit conversion algorithm," IEEE J. Solid-State Circuits, vol. 36, pp. 1538–1545, 2001.
- [8] Y. HeandC.Chang, "A power-delay efficient hybrid carrylookaheadcarry-select based redundant binary to two's complementconverter," IEEE Trans. Circuits Syst. I, Reg. Papers,
- [9] G. Wang and M. Tull, "A new redundant binary number to 2'scomplementnumber converter," in Proc. Region 5 Conference:Annual Technical and Leadership Workshop, pp. 141-143, 2004.
- [10] W. Yeh and C. Jen, "High-speed Booth encoded parallel multiplier design," IEEE Trans. Computers, vol. 49, pp. 692-701, 2000.
- [11] S. Kuang, J. Wang, and C. Guo, "Modified Booth multiplierwith a regular partial product array," IEEE Trans. Circuits Syst.II, vol. 56, pp. 404-408, 2009.
- [12] J. Kang and J. Gaudiot, "A simple high-speed multiplier design," IEEE Trans. Computers, vol. 55, pp.1253-1258, 2006.

