ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 5, May 2017

Implementation of Radix-8 Partial Product Generator for Redundant Binary Multipliers

S.Priya, M.Sasikala

***Abstract*— Redundant Binary Partial Product Generator technique are used to reduce by one row the maximum height of the partial product array generated by a radix-4 Modified Booth Encoded multiplier, without any raise in the delay of the partial product creation Block. This RBPPG technique is of particular interest in all multiplier designs, but especially in Arithmetic multipliers for high-performance ALU Designs and Processors. The proposed method is general and can be extended to higher radix encodings, as well as to any size square and m x n Array multipliers. We evaluate the proposed approach by comparison with Normal Booth Multiplier; the results based on a rough theoretical analysis and on logic synthesis showed its efficiency in terms of area, delay and power. Simulation results show that the proposed Multiplier based designs significantly improve the area, delay and power consumption when the word length of each operand in the multiplier is 8 & 32 bits.**

***Index Terms*— Redundant binary, modified Booth encoding, RB partial product generator, RB multiplier..**

I. INTRODUCTION

Digital multipliers are widely used in arithmetic units of microprocessors, multimedia and digital signal processors. 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 redundant binary number representation has been introduced by Avizienis [1] to perform signed-digit arithmetic; the RB number has the capability to be represented in different ways.

Fast multipliers can be designed using redundant binary addition trees [2-3]. The redundant binary representation has also been applied to a floating-point processor and implemented in VLSI [4]. High performance RB multipliers have become popular due to the advantageous features, such as high modularity and carry-free addition [5-9].A RB multiplier consists of a RB partial product (RBPP)generator, a RBPP reduction tree and a RB-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 __/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_(_/4 + 1)_,
= n − 1, if _ = 2_ (1)

If the additional ECW can be removed, an RBPP accumulation stage is saved, so resulting in improvements 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 removed, then the number of RBPP accumulation stages is reduced to 3, i.e., the stage count is decreased by 25%. Note that the problem of extra ECW does not exist in standard significand size (i.e., 24×24-bit and 54×54-bit) RB multipliers as used in floating point-arithmetic units [5-6]. Alternatively, 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]. Besli et 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. 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 __/4 + 1_ to __/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. While a modest increase in delay is encountered (approximately 5%), the power-delay product (PDP) at word lengths of at least 32 bits confirms that the proposed designs are the best also by this figure of merit. This paper is organized as follows. Section 2 introduces radix-4 Booth encoding. The design of the conventional RBPP generator is also reviewed. Section 3 presents the proposed RBMPPG. This section also demonstrates the adoption of the proposed RBMPPG into various word-length RB multipliers. Section 4 provides the evaluation results of the new RB multipliers using the proposed RBMPPG for different word lengths and compares them to previous best designs found in the technical literature. The conclusion is provided in Section5.

II EXISTING METHOD

A.Radix-4 Booth Encoding and Decoding

Radix-4 Booth encoded modulo 2n +1 multiplier using a proprietary number representation that is suited only for the International Data Encryption Algorithm was proposed. Booth encoded modulo 2n +1 multipliers using diminished-1 and weighted-binary representations were described. However, the multiplexers (MUXs) employed to generate the modulo-reduced partial products and the correction factor increase the circuit area and power dissipation.

The proposed approach not only minimizes the number of Booth Encoder (BE) and Booth Decoder (BD) blocks for partial product generation, but also eliminates the overhead of correction factor generation. The proposed modulo 2n +1 multiplier saves area substantially and dissipates less power in comparison with the non-encoded multiplier and the Booth encoded multiplier. The Energy-Delay Product (EDP) analysis shows favorable results for the proposed multiplier.

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 is summarized, where $A=a_{N-1}\ a_{N-2}\ \ldots\ a_2\ a_1\ a_0$ stands for the multiplicand, and $B=b_{N-1}\ b_{N-2}\ \ldots\ b_2\ b_1\ b_0$ stands for the multiplier. The multiplier bits are grouped in sets 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}.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. Methods have been proposed to solve the problem of correction bits for NB radix-4 Booth encoding (NBBE-2) multipliers. However, this problem has not been solved for RB MBE multipliers. we designed a Booth Encoder based on MBE Scheme as per the bellow table.

TABLE 1

MBE SCHEME

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.THE MODIFIED PARTIAL PRODUCT VARIABLES Q^+_{18} ,Q^+_{19} AND Q^+_{20}

The critical path delay (the dash line) consists of a 1-stage AND-OR-Inverter gate, a 1-stage inverter, and 2-stage TGs. Therefore, RBMPPG-2 just increases the TG delay by 1-stage compared with the MBE partial product of the above technique can be applied to design any 2 -bit RB multipliers. It eliminates the extra ECWN/4 and saves one RBPP accumulation stage, i.e., three XOR gate delays, while only slightly increasing the delay of the partial product generation stage. In general, an N-bit RB multiplier has N/4 RBPP rows using the proposed RBMPPG-2. The partial product variables $P^+1_{(N+1)}$, P^+_{1N}, $P^-_{(N/4)1)}$ and $P^-_{(N/4)0}$ can be replaced $Q^+1_{(N+1)}$,Q^+_{1N},$Q^-_{(N/4)1)}$ and $Q^-_{(N/4)0}$. The radix-4 Booth decoding of a PPR (PP /) needs additional 3-input OR gates (Fig. 4). Therefore, the extra ECWN/4 is removed by the transformation of 4 partial product variables

$Q^{+}1_{(N+1)}, Q^{+}_{1N}, Q^{-}_{(N/4)1}$ and $Q^{-}_{(N/4)0}$ .one partial product row is saved in RB multipliers with any power-of-two word-length.

III PROPOSED SYSTEM

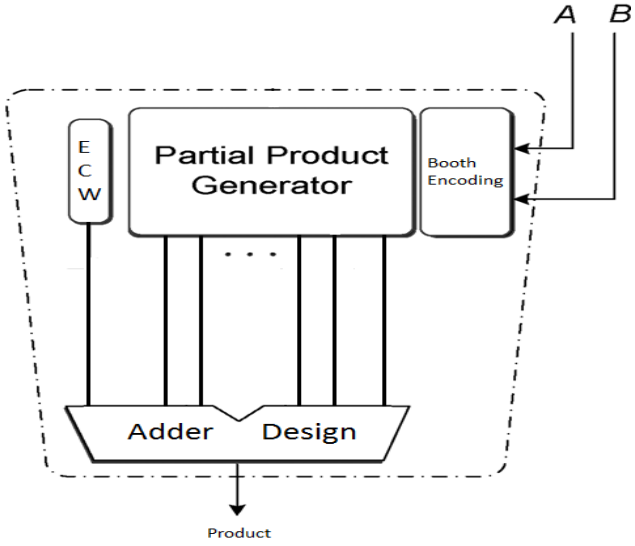


Fig.1 Block diagram

The proposed RBMPPG-2 can be applied to any 2n-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 (Pi+ , Pi-); 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.

A.RADIX-8 BOOTH ENCODING

Radix-8 Booth encoded modulo 2n +1 multiplier using a proprietary number representation that is suited only for the International Data Encryption Algorithm was proposed. Booth encoded modulo 2n +1 multipliers using diminished-1 and weighted-binary representations were described. However, the multiplexers (MUXs) employed to generate the modulo-reduced partial products and the correction factor increase the circuit area and power dissipation. The proposed approach not only minimizes the number of Booth Encoder (BE) and Booth Decoder (BD) blocks for partial product generation, but also eliminates the overhead of correction factor generation. The proposed modulo 2n +1 multiplier saves area substantially and dissipates less power in comparison with the non-encoded multiplier and the Booth encoded multiplier . The Energy-Delay Product (EDP) analysis shows favorable results for the proposed multiplier.

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 is summarized, where $A=a_{N-1}\ a_{N-2}\ \ldots\ a_2\ a_1\ a_0$ stands for the multiplicand, and $B=b_{N-1}\ b_{N-2}\ \ldots\ b_2\ b_1\ b_0$ stands for the multiplier. The multiplier bits are grouped in sets 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}. 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. Methods have been proposed to solve the problem of correction bits for NB Radix-8 Booth encoding (NBBE-2) multipliers. However, this problem has not been solved for RB MBE multipliers. we designed a Booth Encoder based on MBE Scheme as per the bellow table.

A multiplexer (or mux) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2^n inputs has n select lines; Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth. A multiplexer is also called a data selector. Multiplexers can also be used to implement Boolean functions of multiple variables.An electronic multiplexer makes it possible for several signals to share one device or resource, for example one A/D converter or one communication line, instead of having one device per input signal. Alternatively, a high-radix Booth encoding technique can reduce the number of partial products.

TABLE 2
MBE SCHEME

Radix-8 Booth Encoding				
Inputs(bits of m-bit multiplier				Partial Product
X(2)	X(1)	X(0)	X(-1)	PPRi
0	0	0	0	0
0	0	0	1	Y
0	0	1	0	Y
0	0	1	1	2Y
0	1	0	0	2Y
0	1	0	1	3Y
0	1	1	0	3Y
0	1	1	1	4Y
1	0	0	0	-4Y
1	0	0	1	-3Y
1	0	1	0	-3Y
1	0	1	1	-2Y
1	1	0	0	-2Y
1	1	0	1	-Y
1	1	1	0	-Y
1	1	1	1	0

B.MBE SCHEME ENCODER

Both MBE and ECB coding schemes introduce errors and two correction terms are required. A single ECW can compensate errors from both the MBE encoding and the Radix-8 Booth recoding. ECW Code words will assigned like bellow Fig. The Last row Does not Contains any ECW words.

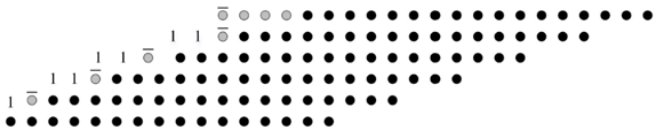


Fig.2 MBE Scheme encoder

C.ADDER

The Decoder Part is designed by adder, is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic units. A ripple carry adder is a logic circuit in which the carry-out of each adder is the carry in of the arry adder because each carry bit gets rippled into the next stage. Based on this design we use multiple inputs into one adder block.

D.MODIFIED BOOTH ENCODING

This modified booth multiplier is used to perform high-speed multiplications using modified booth algorithm. This modified booth multiplier's computation time and the logarithm of the word length of operands are proportional to each other. We can reduce half the number of partial product. Radix-4 booth algorithm used here increases the speed of multiplier and reduces the area of multiplier circuit. In this algorithm, every second column is taken and multiplied by 0 or +1 or +2 or -1 or -2 instead of multiplying with 0 or 1 after shifting and adding of every column of the booth multiplier. Thus, half of the partial product can be reduced using this booth algorithm. Based on the multiplier bits, the process of encoding the multiplicand is performed by radix-4 booth encoder.

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 is summarized in Table I, where A= $a_{N-1}\, a_{N-2}\, \ldots\, a_2\, a_1\, a_0$ stands for the multiplicand, and B= $b_{N-1}\, b_{N-2}\, \ldots\, b_2\, b_1\, b_0$ stands for the multiplier. The multiplier bits are grouped in sets 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}.

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. Methods have been proposed to solve the problem of correction bits for NB radix-4 Booth encoding (NBBE-2) multipliers. However, this problem has not been solved for RB MBE multipliers

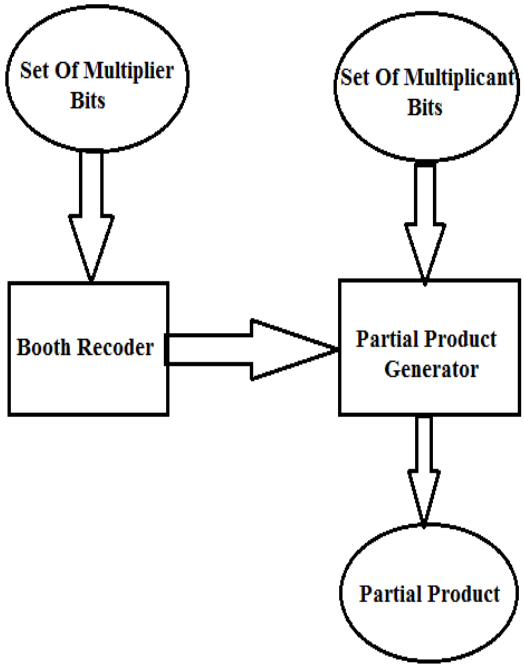
Booth multiplication algorithm or Booth algorithm can be defined as an algorithm or method of multiplying binary numbers in two's complement notation. It is a simple method to multiply binary numbers in which multiplication is performed with repeated addition operations by following the booth algorithm. Again this booth algorithm for multiplication operation is further modified and hence, named as modified booth algorithm.

E.PARTIAL PRODUCT GENERATOR

The main objective is to reduce the area and energy of the system. This can be achieved by minimizing the redundancy of the circuit. A partial product generator generates a product formed by multiplying the multiplicand by one digit of the multiplier when the multiplier has more than one digit. Partial products are used as intermediate steps in calculating larger products. For a partial product to be generated a constant multiplication and addition of the multiplicands has to be performed. The multiplication is performed in an AND logic and addition in OR logic. The input is given in the form of array to minimize the delay of the system. Note that all partial product outputs are generated in a parallel manner with a constant time.

In hardware multipliers, the generation of partial products is a necessary step in the process known to the art for efficient production of a final product. A way to increase the speed of hardware multipliers is through the use of the Booth algorithm. The alternate Booth partial product generation for hardware multipliers of the present invention is directed to a method and apparatus for eliminating the encoding of the bits of the multiplier prior to entering the partial product generating cell of the present invention which may result in less hardware and increased speed.

. 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.



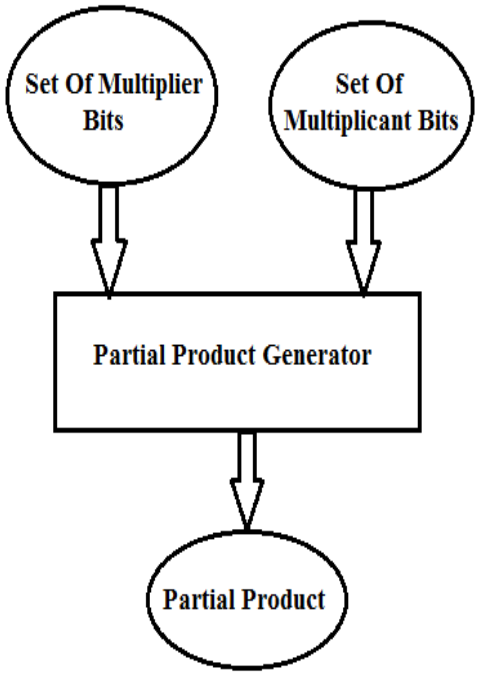


Fig.3 Partial Product Generator

IV EXPERIMENTAL RESULTS

PROPOSED-RBMPPG-2 8 BIT ARCHITECTURE

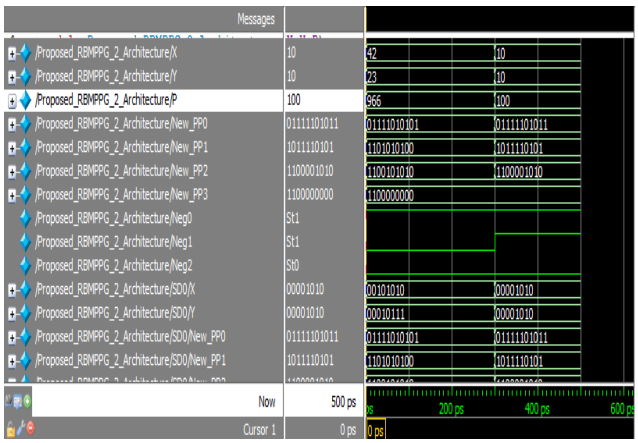


Fig.4 Proposed-RBMPPG-2 8 Bit Architecture

PROPOSED-RBMPPG-2 32 BIT ARCHITECTURE

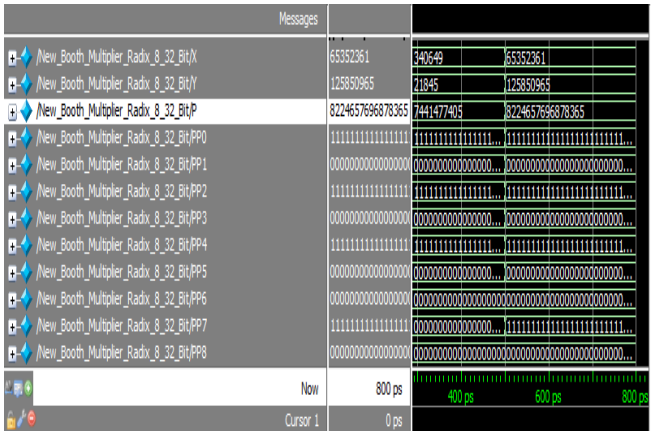


Fig.5 Proposed-RBMPPG-2 32 Bit Architecture

V COMPARISION RESULTS

POWER CALCULATION

METHOD NAME	POWER	
	Voltage	Current
Normal booth radix-4 32 bit multiplier	72437.50 mW	28975.0mA
Proposed booth radix-4 32 bit multiplier	60325.0 mW	24130.0mA
Normal booth radix-8 32 bit multiplier	6485633 mW	3603129mA
Proposed booth radix-8 32 bit multiplier	1629930 mW	905516mA

DELAY CALCULATION

METHOD NAME	DELAY		
	Over all delay	Gate delay	Path delay
Normal booth radix-4 32-bit multiplier	211.77ns	75.85ns 35.8%	135.92ns 64.2%
Proposed radix-4 32-bit multiplier	201.17ns	73.43ns 33.7%	127.74ns 63.5%
Normal booth radix-8 32-bit multiplier	190.41ns	64.03ns 33.6%	126.38ns 66.4%
Proposed radix-8 32-bit multiplier	132.06ns	43.04ns 32.6%	89.02ns 67.4%

VI CONCULATION

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 2-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] A.Avizienis, "Signed-digit number representations for fast parallel 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 binary adder tree," IEEE J. Solid-State Circuits, vol. SC-22, pp. 28-34, 1987.

[4] 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.

[5] 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.

[6] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Makino, "An 8.8-ns 54×54-bit multiplier with high speed 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-free 54b×54b multiplier using equivalent bit conversion algorithm," IEEE J. Solid-State Circuits, vol. 36, pp. 1538–1545, 2001.

[8] Y. He and C. Chang, "A power-delay efficient hybrid carrylookahead carry-select based redundant binary to two's complement converter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, pp. 336–346, 2008.

[9] 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.

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