International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER SSRN

IJIEMR Transactions, online available on 7th Jun 2019. Link :http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-06

Title: **DESIGN OF ADVANCED REDUNDANT BINARY MULTIPLIER USING PARTIAL PRODUCT GENERATOR**

Volume 08, Issue 06, Pages: 293–300.

Paper Authors

GUTTULA VIJAY, RAJESHBABU CHITIKENA
Godavari Institute of Engineering and technologyRajahmundry, Andhrapradesh 533296

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DESIGN OF ADVANCED REDUNDANT BINARY MULTIPLIER USING PARTIAL PRODUCT GENERATOR

[1]GUTTULA VIJAY, [2]RAJESHBABU CHITIKENA

[1]ECE Department, Godavari Institute of Engineering and technology, Rajahmundry, Andhrapradesh 533296

[2]Assistant Professor,ECE Department,Godavari Institute of Engineering and technologyRajahmundry, Andhrapradesh 533296

[1]gvijay236.236@gmail.com, [2]rajeshchitikena@gmail.com

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 implication 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 hybrid adder stage for the MBE multiplier. In this paper, a new RB modified partial product generator (RBMPPG) is proposed. And existed design partial product addition can be done by using RBPPA. In modified design partial products addition is done by hybrid adder. Propagation delay path between input and output is less in modified design than existed design. Simulation results show that the modified hybrid RBMPPG based designs significantly improve the memory and speed when the word length of each operand in the multiplier is at least 32 bits.

KEYWORDS—Redundant binary, modified Booth encoding, RB partial product generator, RB multiplier, hybrid adder.

I. INTRODUCTION

Multiplication plays important roles in various digital systems such as computers, process controllers, signal processors, and so on. Designing fast multipliers has long been a great theoretical and practical interest for computer scientists and engineers. Various multiplication algorithms have been proposed and practically used [1]. Especially, with recent advances of technology of integrated circuits, many researchers have tried to develop high-speed multiplication algorithms which are suitable for VLSI implementation (e.g., [1]). In this paper, they propose a high speed multiplication algorithm internally using redundant binary representation [1].Multiplier can be implemented as either combinational or pipelined multiplier in terms of hardware. Combinational multiplier implementation increases both hardware resources and critical path in two stages. At stage1, N*N number of inputs produced N number of PPRs. At stage2, number of adders increases for the design so combinational method is not efficient for large multiplier design. Pipelined multiplier overcomes

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

the above disadvantages with recoding structure. Targeting the optimized design of pipelined multiplier unit, partial product reduction techniques and addition approaches are discussed in section2. PPRs generation and addition processes involved to design different multipliers with constraints are discussed below.In n bit binary integer multiplication, n partial products are first generated and then added up pair wise by means of a binary tree of redundant binary adders. Since parallel addition of two n-digit redundant binary numbers can be performed in a constant time independent of n without carry propagation, n bit multiplication can be performed in a time proportional to log2 n. The computation time is almost the same as that by a multiplier with a Wallace tree, in which three partial products will be converted into two, in contrast to our two-to-one conversion, and is much shorter than that by an array multiplier for longer operands. The number of computation elements of an n bit multiplier based on the algorithm is proportional to n2. It is almost the same as those of conventional ones. Furthermore, since the multiplier has a regular cellular array structure similar to an array multiplier, it is suitable for VLSI implementation. Thus, the multiplier is excellent in both computation speed and regularity in layout. It can be implemented on a VLSI chip with an area proportional to n2 log2 n. The algorithm can be directly applied to both unsigned and 2's complement binary integer multiplication.

Paper [3] describes a 16-bit X 16-bit multiplier for 2 two's complement binary numbers based on a new algorithms. This multiplier has been fabricated on an LSI chip using a standard n-E/D MOS process technology with a 2.7-pm design rule. This multiplier is characterized by use of a binary tree of redundant binary adders. In the new algorithm, n-bit multiplication is performed in a time proportional to Iogz n and the physical design of tree multiplier is constructed of a regular cellular array. This new algorithm has been proposed by 'rakagi et a~ The 16-bit x 16-bit multiplier chip size is 5.8x 6.3 mm using the new layout for a binary adder tree. The chip contains about 10 600 transistors, and the longest logic path includes 46 gates. The multiplication time was measured as 120ns. We estimate that a 32-bit X 32-bit multiplication time is about 140 ns. Redundant binary representation [4] was adopted to remove carry propagation. Each of the redundant binary digits, [-1, 0, **11,** has been encoded into two binary bits. This has made it possible to perform multiplication faster than the combination of Booth algorithm and Wallace tree, and division faster than SRT.A new redundant binary (RB) architecture for the high speed multiplier is presented in [5]. In this architecture, a pair of partial products is converted to one RB number by inverting one of the pair without additional circuit and latency. Generated RB partial products are added by the Wallace tree of improved RB adders (RBAs) which have a latency of 0.9ns, and converted to a normal binary (NB) number by a simply structured RB-to-NB converter in which the carry propagation circuit is constructed only with simple selector circuits.The existing RBMPPG [1] 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; these reductions over previous NB multiplier designs incur in a modest delay increase (approximately 5%). The power-delay product can be reduced by up to 59% using the proposed RB multipliers when compared with existing RB multipliers. In modified design partial products generation is same as existed one but addition is done by using hybrid adder which enhances the delay and memory.

II. MODIFIED DESIGNARCHITECTURE

Two important steps needed for modified multiplier design.

1. First step to generate partial product rows (PPRs).
2. Second step to add PPRs.

Figure 1 shows the algorithm for modified design.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

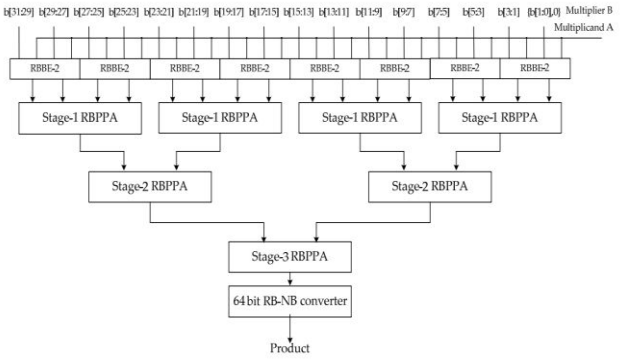

Figure 1: Existing multiplier block diagram

The above figure 1 shows the architecture of existed system. The multiplier consists of 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). The 64-bit RB-NB converter converts the final accumulation results into the NB representation, which uses a hybrid parallel prefix/carry select adder (as one of the most efficient fast parallel adder designs). But this existed system does not produce effective results so a new system which is discussed in below section.

The aim of this study is implementation of modified partial product generator for RB multipliers.

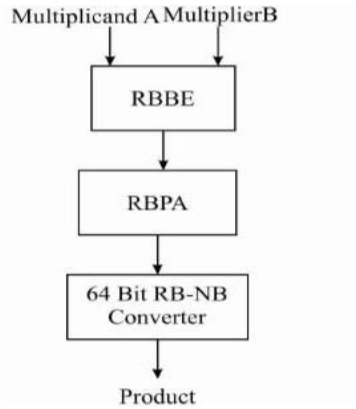

Figure 2: Multiplier (existing &modified) flow chart

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 [1,2]. A RBPP row can be obtained from two adjacent NB partial product rows by inverting one of the pair rows.

RADIX -4 BOOTH ENCODING

Booth encoding has been proposed to facilitate the multiplication of two's complement binary numbers [18]. It was revised as modified Booth encoding (MBE) or radix-4 Booth encoding [19].The MBE scheme explained in the table, where A = a_{N-1} a_{N-2} …..a_2 a_1 a_0 stands 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 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 [11-14].

TABLE I

TABLE I

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*

PRODUCT GENERATOR

As two bits are used to represent one RB dig it, then a RBPP is generated from t wo NB partial products [1-6]. The addition of two N-bit NB partial products

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

X and Y using two's complement representation can be expressed as follows

$$X + Y = X - Y - 1 = (X, Y^{-}) - 1$$

Where Y^{-} is the inverse of Y.The RBPP is generated by inverting one of the two NB partial products and adding -1 to the LSB.. Each RB dig it X_i belongs to the set{-1,0,1};this is coded by two bits (X_i^{-}, X_i^{+}) .RB numbers can be coded in several ways.Table 11 shows one specific RB encoding

TABLE II

RB ENCODING USED IN THIS WORK [6]

X_i^+	X_i^-	RB digit (X_i)
0	0	0
0	1	$\overline{1}$
1	0	1
1	1	0

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.

MODIFIED RB PARTIAL PRODUCT GENERTOR

A new RB modified partial product generator based on MBE (RBM PPG-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^{+}) and the two LSBs of the last partial product row (PP^{-})

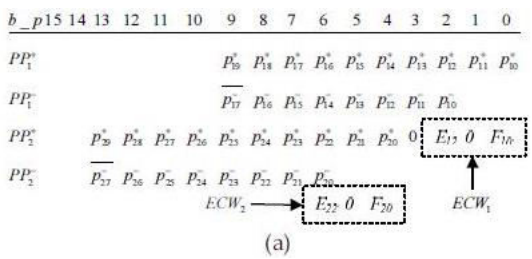

Fig.2(a)The first new RBM PPG-2 architecture for an 8-bit MB multiplier.

It is differ from conventional type by its error correcting vector. In this type error correcting vectors ECW1 is generated by PP1 and ECW 2 is generated by PP2.

$$ECW1 = 0\ E_{12}\ 0\ F_{10}$$

$$ECW2 = 0\ E_{22}\ 0\ F_{20}$$

To eliminate a RBPP accumu lation, ECW_2 needs to be incorporated into PP1 and PP2.

F_{20}={-1 , b_5 b_4 b_3=000,001,010,011,111} F_{20}={0 , $b_5 b_4 b_3$=100,101,110}

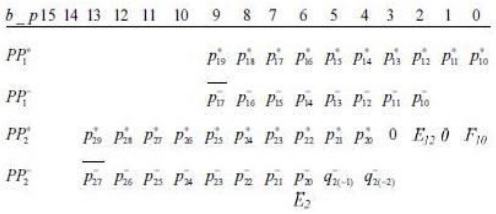

Fig 2(b) Revised RBM PPG by replacing E_{22} AND F_{20}

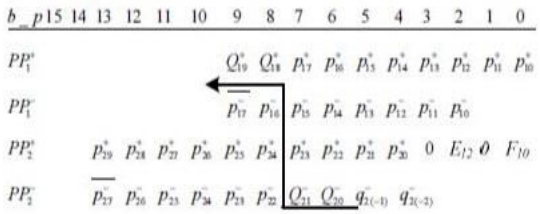

Fig2(c) Final proposed RBMPPG by totally eliminating ECW_2

Q_{19}^{+}, Q_{18}^{+}, Q_{21}^{-}, Q_{20}^{-} are used to represent the modified part ial products .By setting $PP2^{+}$ 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_2 ,$q_{2(-2^{-})}$,$q_{2(-1)^{-}}$ as follows

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

$$E_2 = \begin{cases} E_{22}, & F_{20} = 0 \\ E_{22} - 1, & F_{20} = -1 \end{cases}$$
$$q^-_{2(-2)} = q^-_{2(-1)} = \begin{cases} 0, & F_{20} = 0 \\ 1, & F_{20} = -1 \end{cases}$$

This is further explained by the truth table of E_{22} ,F_{20} and E_2, $q_{2(-2)}$, $q^{-}_{2(-1)}$.

TABLE III

TRUTH TABLE OF E_2, $q^-_{2(-2)}$, $q^-_{2(-1)}$ AND p^-_{21},p^-_{20}.

$b_7b_6b_5$	$E_{22}F_{20}$	$E_2q^-_{2(-2)}q^-_{2(-1)}$	p^-_{21}	p^-_{20}
0 0 0	0 $\bar{1}$	$\bar{1}$ 1 1	0	0
0 0 1	0 0	0 0 0	a_1	a_0
0 1 0	0 $\bar{1}$	$\bar{1}$ 1 1	a_1	a_0
0 1 1	0 0	0 0 0	a_0	0
1 0 0	1 $\bar{1}$	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 $\bar{1}$	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}^{-}, Q_{20}^{-} and P_{19}^{+}, P_{21}^{-}, P_{20}^{-} 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

Logic functions of Q_{19}^{+}, Q_{18}^{+}, $Q_{21}^{-}$and Q_{20}^{-} can be expressed as follows

$$Q^+_{19} = (b_7 \oplus b_5 + b_7b_6b_5) \cdot p^+_{19} + \overline{b_7}\,\overline{b_5} \cdot (p^+_{18} + p^-_{21} + p^-_{20} + p^+_{19}) + b_7\overline{b_6}b_5 \cdot (p^+_{18}p^-_{21}p^-_{20} \oplus p^+_{19})$$

$$Q^+_{18} = (b_7 \oplus b_5 + b_7b_6b_5) \cdot p^+_{18} + \overline{b_7}\,\overline{b_5} \cdot (\overline{p^-_{21}+p^-_{20}} \oplus p^+_{19}) + b_7\overline{b_6}b_5 \cdot (p^-_{21}p^-_{20} \oplus p^+_{18})$$

$$Q^-_{21} = (b_7 \oplus b_5 + b_7b_6b_5) \cdot p^-_{21} + \overline{b_7}\,\overline{b_5} \cdot \overline{p^-_{21} \oplus p^-_{20}} + b_7\overline{b_6}b_5 \cdot p^-_{21} \oplus p^-_{20}$$

$$Q^-_{20} = (b_7 \oplus b_5 + b_7b_6b_5) \cdot p^-_{20} + \overline{b_7}\,\overline{b_5} \cdot \overline{p^-_{20}} + b_7\overline{b_6}b_5 \cdot \overline{p^-_{20}}$$

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.

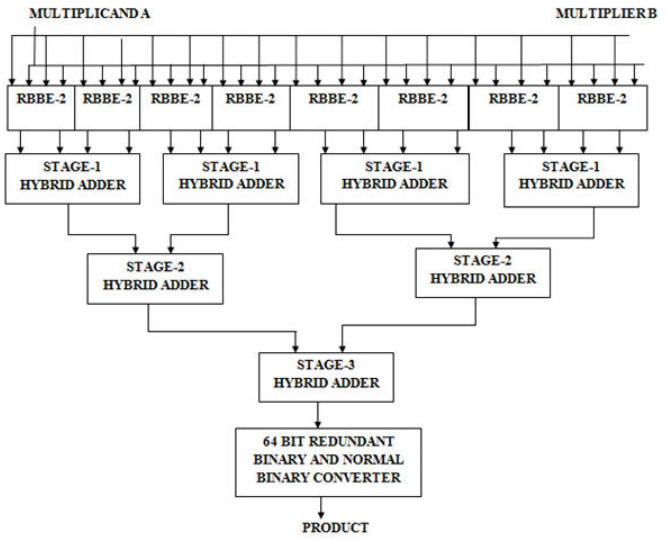

Figure 3: Modified design block diagram

From above figure (2) we can observe the block diagram of proposed system. A proposed multiplier is a combinational logic circuit used in digital systems to perform the multiplication of two binary numbers. These are most commonly used in various applications especially in the field of digital signal processing to perform the various algorithms. A redundant binary representation which is one of the signed digit representations proposed by Avizienis is internally used by this multiplier. n-bit multiplier based on the new algorithm as shown in Fig, 4.1. N partial products are added up pair wise in a log2 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 log2 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 log2n is used for this conversion. As a result, n-bit multiplication is carried out in a time proportional to log2 n.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

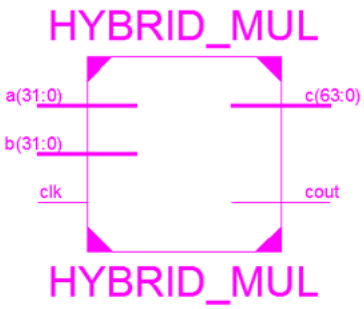

Figure 4: Modified RTL schematic

RBPA

In the second stage, a 4-stage RBA summing tree is used to sum 16 RB partial products. Each RBA b lock 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 b locks 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).

RB –NB CONVERTER

The 64-bit RB-NB converter converts the final accumulation results into the NB representation, which uses a hybrid parallel prefix/carry select adder.

III. PERFORMANCE ANALYSIS AND FUNCTIONAL VERIFICATION

Finally we verify the results for 2 different types of multiplier designs which are having same partial product generation method but partial product rows are added by using RBPPA in existing design and hybrid adder used in modified design.

1 RBBE based multiplier with hybrid adder

2 RBBE based multiplier with RBPPA

Above modified design is coded in verilog HDL synthesized, simulated and implemented using XILINX 14.7 software with FPGA technology SPARTAN 3E. Modified design RTL schematic, functional verification and FPGA design testing output with the help of Isim simulator.

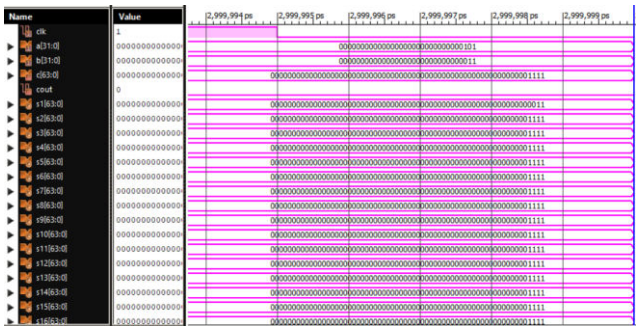

Figure 5: Modified design stimulus

Power consumption analysis modified design is compared with the existed design [11]. Table 2 shows the comparison of results with help of equation shown in below as

$$P_d = \propto_t f_{clk} C_{load} V_{dd}^2 \quad (2)$$

Where $\propto$t=switching levels of design,

f_{clk} = clock frequency,

C_{load} = load capacitance of the design,

V_{dd}^2 = supply voltage.

Comparison of delay and memory results of 2 designs shown in below figures and tables.

TABLE IV: DELAY AND MEMORY ANALYSIS

	Total delay	**Logic delay**	**Route delay**	**Memory**
Existing system	127.363 nsec	60.608 nsec	49.755 nsec	4667688KB
Proposed system	43.834 nsec	3.971 nsec	39.863 nsec	4619840KB

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

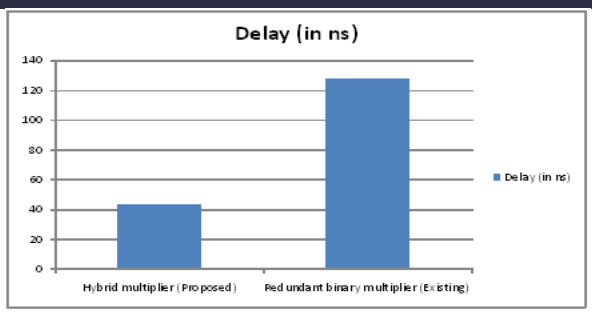

Figure 6: Delay report for 2 designs

IV. CONCLUSION

In this paper, performance of 32-bit RBBE based multiplier with hybrid adder is presented. The design is implemented with the help of SPARTAN 3E FPGA technology. Different algorithms are studied for reduce the partial product rows like RADIX-4 modified booth encoding, RBBE booth encoding etc. Finally modified design using RBBE MBE to generate PPRs. Addition between the PPRs is done with the help of hybrid adder instead of different adders like Carry Save Adder, Carry Skip Adder and etc. Modified design has 66% less delay compared with existed design.

ACKNOWLEDGMENT

We would like to thank all the authors in the references for providing great knowledge and helpful advices when ever required.

REFERENCES

[1] SmrutiBokade, PravinDakhole, "CLA based 32-Bit Signed Pipelined Multiplier," International Conference on Communication and Signal Pro- cessing, April 6-8, 2016, India.

[2] Avizienis, "Signed-digit number representations for fastparallel arith met ic," IRE Trans. Electron. Co mputers,vol. EC-10,pp. 389– 400, 1961.

[3] Harata, Y. Nakamu ra, H. Nagase, M. Takigawa, and N.Takagi, "A high speed mult iplier using a redundant binaryadder tree," IEEE J. Solid-State Circuits, vol. SC-22, pp. 28-34,1987.

[4] Shiann-RongKuang, Jiun-Ping Wang, Cang-Yuan Guo, "Modified Booth Multipliers With a Regular Partial Product Array" IEEE Transactions on Circuits and Systems, DOI 10.1109/TCSII.2009.2019334, Vol56, Issue 5, May2009.

[5] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S.Kuninobu, "A33 MFLOPS floating point processor using redundant binary representation," in Proc. IEEE Int. Solid -State Circu its Conf.(ISSCC), pp. 152–153, 1988.

[6] H. Makino, Y. Nakase, and H. Shinohara, "A 8.8-ns54x54-bit mu ltiplier using new redundant binary architecture," in Proc.Int. Conf. Co mput. Design (ICCD), pp. 202-205, 1993.

[7] Y. Kim, B. Song, J. Grosspietsch, and S. Gillig, "A carry -free54b×54b mult iplier using equivalent bit conversion algorith m,"IEEE J. Solid-State Circu its, vol. 36, pp. 1538–1545,2001.

[8] HeandC.Chang, "A power-delay efficient hybrid carrylookaheadcarry-select based redundant binary to two'scomplementconverter," IEEE Trans. Circu its 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. Co mputers, vol. 49, pp. 692-701,2000.

[11]Aihongyao, Ling Li and Mengzhe sun "Design of Pipeline Multiplier Based on Modified Booth's Algorithm and Wallace Tree" ECWAC 2011, part 1, CCIS 143,pp 67-73,2011.

[12] S. Kuang, J. Wang, and C. Guo, "Modified Booth mu ltip lierwith a regular part ial product

array," IEEE Trans. Circuits Syst.II, vol. 56, pp. 404-408, 2009.

[13] J. Kang and J. Gaudiot,"A simple high -speed mu ltiplier design,"IEEE Trans. Co mputers, vol. 55, pp.1253-1258, 2006.

[14]F. Lamberti, N. Andrikos, E. Antelo, and P. Montuschi, "Reducingthe computation time in (short bit -width) two's complementmu ltipliers," IEEE Trans. Co mputers, vol. 60, pp. 148-156, 2011.

[15] Y. He and C. Chang, "A new redundant binary Booth encoding for fast 2 -bit multiplier design," IEEETrans. Circuits Syst. I,Reg. Papers, vol. 56, pp. 1192–1199, 2009.

[16] Y. He, C. Chang, J. Gu , and H. Fah my, "A novel covalent redundant binary Booth encoder," in Pro c. IEEE Int. Symp. CircuitsSyst. (ISCAS), vol. 1, pp. 69– 72, 2005