A Novel VLSI Architecture of Multiplier on Radix – 4 using Redundant Binary Technique

L.Sriharish
Department Of Electronics And Communication
Gudlavalleru Engineering College
Gudlavalleru

M. Kamaraju, PhD
Department Of Electronics And Communication
Gudlavalleru Engineering College
Gudlavalleru

ABSTRACT
The work mainly deals with in improving multiplication process by using Redundant Binary Technique. By implementing the existing method of Multiplication and Accumulation structure in Real time applications, occurs some difficulties like some hard multiples, and getting partial products in multiplication stage, it was not useful for higher radix values. The covalent redundant binary booth encoding algorithm overcomes the hard multiple generation problem and it reduces the partial products. The proposed algorithm dumped into the Booth encoding partial product generation stage. In this stage first step is to change the normal binary to redundant binary to make simple to avoid hard multiples. The partial products also reduce in the same stage. The method is implemented in the Fast Fourier Transform, Digital signal processors, and in Arithmetic logic unit. Finally the output results acquired for this method is number of gates are reduced and partial products are reduced up to 32 for 128 bit processor.

Keywords
RBR technique, BEPPG, CRBBE, RBPPG, RBA summing tree stage, RB to NB stage.

1. INTRODUCTION
The nature of the work is introduced to improve the speed of the multiplier and to remove the hard multiples and to reduce the partial products than previous work. By improving the multipliers in the ALU processors from past to present they are giving better results comparing to previous improvements. So, now introducing the Redundant Binary Representation technique in digital multipliers to overcome drawbacks in previous techniques. Redundant Binary (RB) representations first introduced by Avizienis in 1961 for fast parallel arithmetic. This new Arithmetic was applied for fast multiplication by takagi and implemented by Edamatsu.

1.1. Existing multiplication method
In Multiplier and Accumulation structure the multiplier can be divided into four operation steps[1]. First step- booth algorithm, second step – partial product summation, third step – final addition, fourth step – accumulation as shown in Figure 1.

Step 1: Multiplication process done between n bits Multiplicand (X) and m bits Multiplier (Y).

Step 2: n bits Partial products (P_0~P_j) will be generated after multiplying n bits and m bits.

Step 3: Final addition between Partial products Summation (S) bits and Carry(C) bits.

Step 4: Accumulation results takes place between multiplication results (X*Y) and finally will get Accumulation Result (Z).

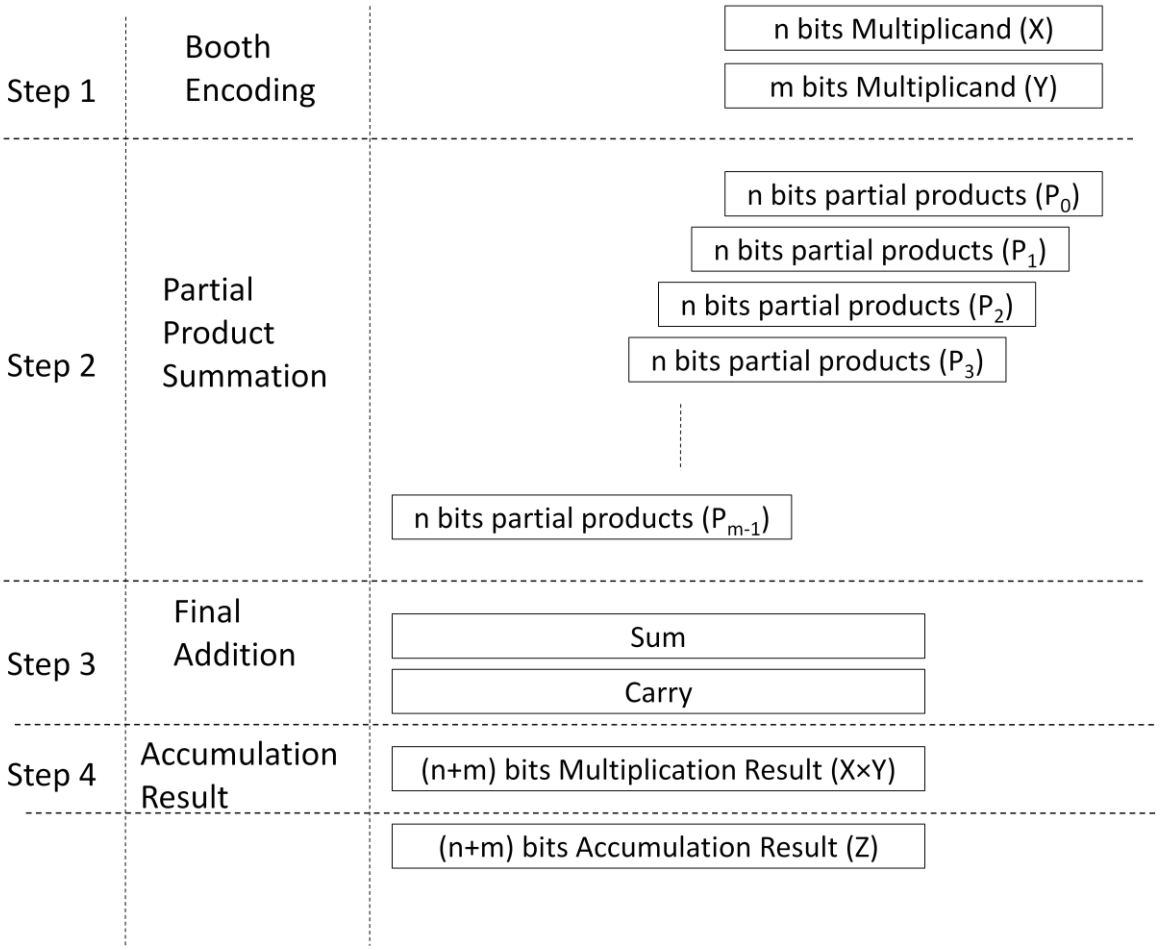

Figure 1: Basic Arithmetic steps of Multiplication and Accumulation

1.2. Proposed Multiplication method using Redundant Binary Representation (RBR) technique

The Redundant Binary Representation Technique will eliminate the hard multiples generation problem by increasing the radix values. A formula will deal in this as follows:

2. HARD MULTIPLES PROBLEM

In radix-**r** Booth-k encoding ($r=2^k$), a signed digit c_i is generated from adjacent binary bits $y_{k(i+1)-1}y_{k(i+1)-2}\ldots y_{ki+1}y_{ki}$ and a borrow bit y_{ki-1} as follows:

$$C_i=-2^{k-1}\times y_{k(i+1)-1}+\sum_{j=2}^{k} 1\times 2^{k-j}\times y_{k(i+1)-j}+y_{ki-1},$$

$$for\ i=0,1,2,\ .....[\frac{N}{k}]-1$$

where **k** is an integer, N is the word length of the multiplier Y and $y_{-1}=0$.As the radix number **r** of Booth-**k** encoder increases, the number of Booth encoded digits and hence the number of partial products decreases to approximately $1/_k$of the original number. However, as the number of multiples increases with the radix to $\mathbf{2^k+1}$, the number of hard multiples also increases simultaneously. Each group is encoded in parallel to generate a select signal from the set {±4M, ±3M, ±2M, ±M, 0}.c_iM refers to the select signal for the partial product c_iX, where X is the multiplicand. The partial product 3X is a hard multiple, which can only be obtained by adding X and 2X and by a carry propagation adder (CPA)[9].

The normal binary booth encoding table will represent the multiplier generation bits in positive sign and negative sign. While operating the processor then priority will be given to the multiple generation bits. According to table multiplier bits will be generated, this process will be done in the internal Architecture of the processor[5].

3. COVALENT REDUNDANT BINARY BOOTH ENCODING (CRBBE) ALGORITHM

Covalent redundant binary booth encoding (CRBBE) algorithm[6] is used to simplify the generation of hard multiples and reduce the number of RB partial products without introducing any form of correction vector. Block diagram of CRBBE as shown in below Figure 2.

Table 1: Radix-8 Normal Binary Booth Encoding (NBBE-3)

Multiplier bits	c_iM	Multiplier bits	c_iM
Y3i+2y3i+1y3i(y3i-1)		Y3i+2y3i+1y3i(y3i-1)	
000(0)	+0	100(0)	-4M
000(1)	+M	100(1)	-3M
001(0)	+M	101(0)	-3M
001(1)	+2M	101(1)	-2M
010(0)	+2M	110(0)	-2M
010(1)	+3M	110(1)	-M
011(0)	+3M	111(0)	-M
011(1)	+4M	111(1)	-0

3.1. Design of CRBBE-4- based RB multiplier

The block diagram of 64*64 consists of 3 stages:

(1) Booth encoder and partial product generator stage (BEPPG stage)
(2) Redundant binary adder summing tree stage (RBA summing stage)
(3) Redundant binary to NB conversion stage (RB-to-NB stage)

Step 1

Booth Encoder and Partial Product Generator stage (BEPPG stage):

In this stage the input [Y_{63}~Y_0] bits given to the CRBBE at top. After getting multiplication through [X_{63}~X_0] from the 5M multiple generation blocks it generates 16 partial products connected to the summing tree stage.

Step 2

Redundant Binary Adder summing tree stage (RBA summing stage):

Here all the 16 partial products generate 68 bits. These bits are added by the Redundant Binary Adders (RBA)[2] denoted as1, 2, 3, 4. RBA_1 had divided into 8 blocks with 68 bits. These 8 blocks (RBA_1) split into 4 blocks (RBA_2) with 72 bits.4 blocks (RBA_2) split into 2 blocks (RBA_3) with 80 bits. Finally, 2blocks (RBA_3) into one block (RBA_4) with 96 bits[7].

Step 3

Redundant Binary to Normal Binary conversion stage (RB-to-NB stage):

Finally the Output (Z_{127}~Z_0) gets 128 bits by adding 32 bits to (RBA_4) 96 bits. Here Redundant Binary bits changes to Normal Binary bits[9].

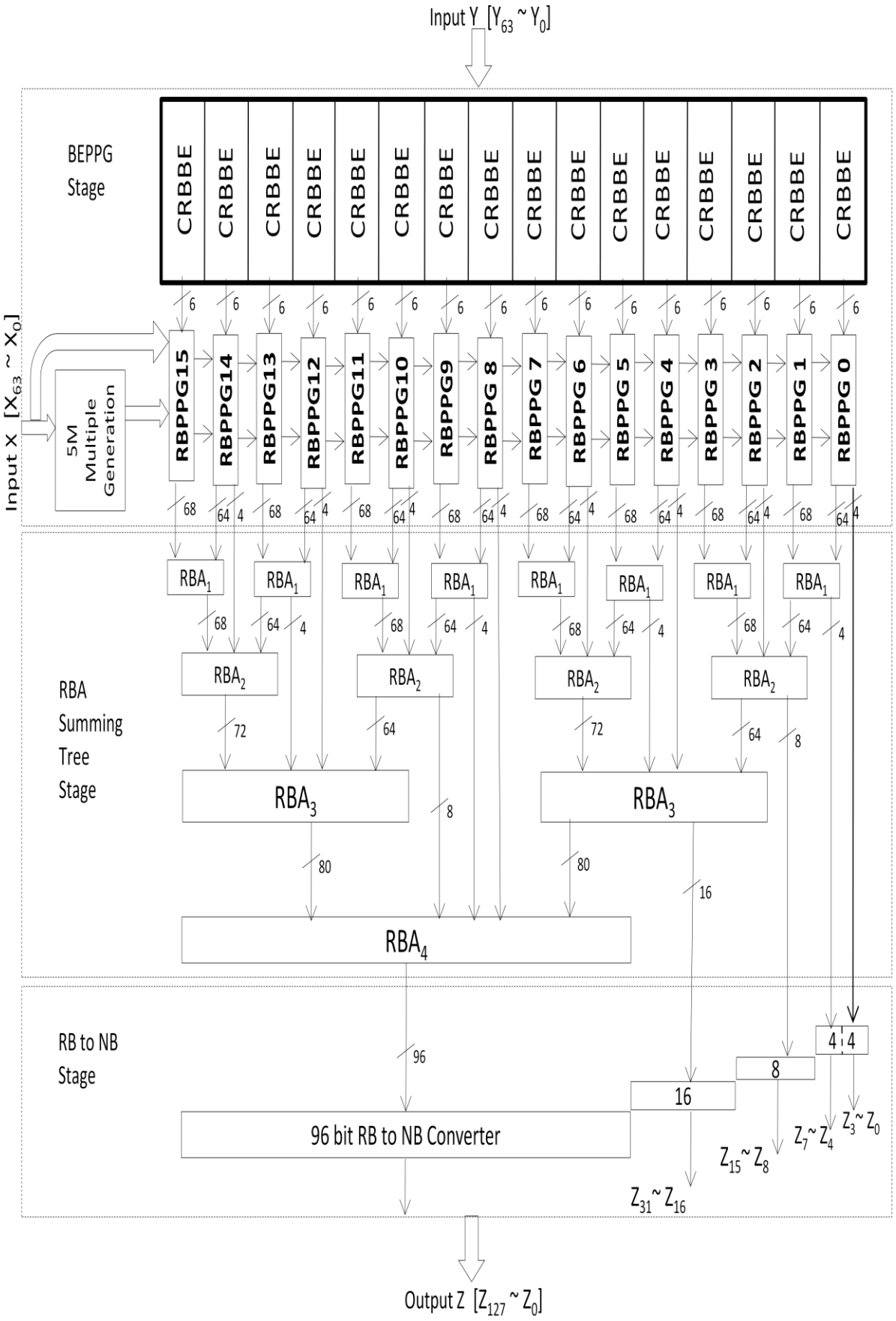

Figure 2: Block diagram of 64*64 RB Multiplier

4. SIMULATION RESULTS

4.1. Existing Method of Simulation Reports: (Modified booth algorithm)

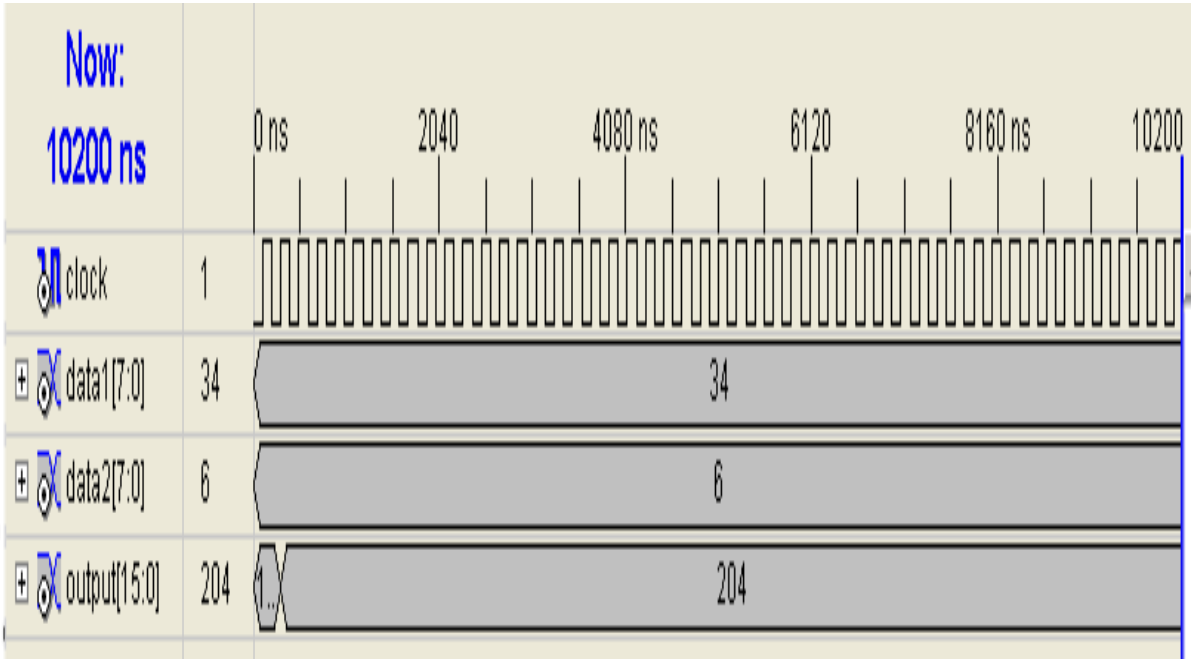

Figure 3: Graph of Existing Simulation Result

The clock signal sets at 1 by giving the 2 data inputs to get output result by dumping the Modified booth algorithm.

Input in Decimal: 34*6
Output in Decimal: 204

Input in Binary : 100010*0110
Output in Binary : 11001100

4.2. Proposed Method of Simulation Reports: (RBR Technique)

By dumping the Covalent Redundant Binary Booth Encoding Algorithm[8] the output result will give less partial products and more speed.

Input in Decimal : 45 *32
Output in Decimal : 1440

Input in Binary : 101101 *100000
Output in Binary : 10110100000

Table 2: Results comparison table

Parameters	Existing	Proposed
No. of gates used	1416	66
Power(mw)	459	224

While comparing with existing method gates and power is reduced in proposed method. In existing method multiplication process is slow due to its Multiplier and Accumulator structure, so it requires more power and obviously more number of gates. In proposed method multiplication process is fast[4] because RBR Technique provides more number of bits, so no need of much more power[3] and no need of more number of gates.

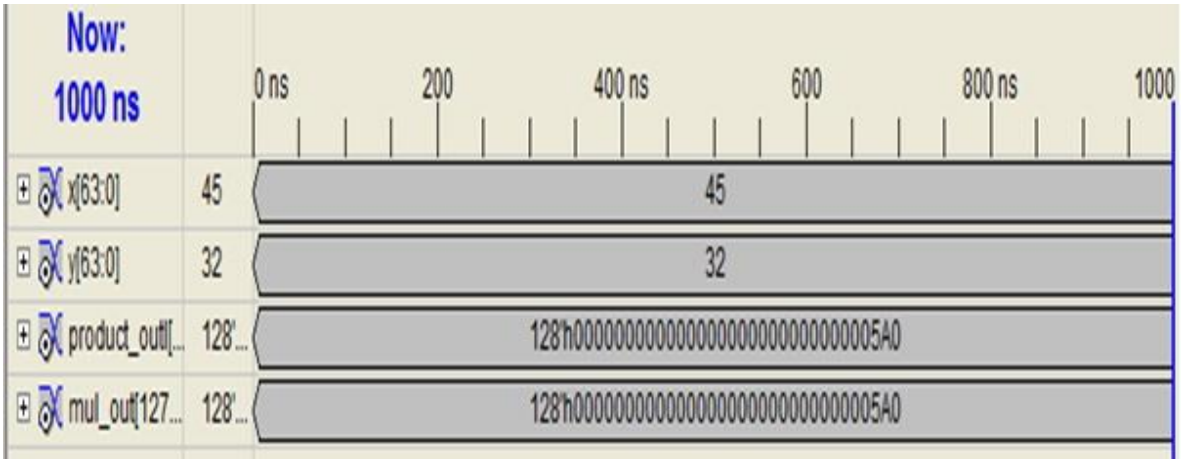

Figure 4: Graph of Proposed Simulation Result

5. CONCLUSION

The RBR technique removes hard multiples and reduces the partial products. By removing hard multiples in the Digital multiplier of the processor, Digital multiplier will perform much faster than existing method. Hence speed of the processor has been increased and it is used for higher multiplications by proposed method. In future RBR technique is used for Digital Signal Processing, Fast Fourier Transformations and multimedia applications.

6. REFERENCES

[1] Young-HO Seo, Dong-Wook Kim, "A VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorithm", IEEE, VLSI Systems, Vol.18, No.2, FEB 2010.

[2] H. Chang, "A power-delay efficient hybrid carry-look ahead/ array-select based redundant binary to two's complement converter, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 2, pp.336–346, Feb. 2008.

[3] Kuan-Hung Chen and Yuan-Sun Chu, "A Low-Power Multiplier With the Spurious Power Suppression Technique" IEEE VLSI SYSTEMS, Vol. 15.NO.7, July 2007.

[4] J.-Y. Kang and J.-L. Gaudiot, "A simple high-speed multiplier design," IEEE Trans. Comput., vol. 55, no. 10, pp. 1253–1258, Oct. 2006

[5] C. Shi, W.Wang, L. Zhou, L. Gao, P. Liu, and Q. Yao, "32B RISC/DSP media processor: MediaDSP3201," SPIE Embedded Processors for Multimedia and Communications II, vol. 5683, pp. 43–52, Mar. 2005.

[6] Y. He, C. H. Chang, J. Gu, and H. A. H. Fahmy, "A novel covalent redundant binary Booth encoder," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS'2005), Kobe, Japan, May 2005, vol. 1, pp. 69–72.

[7] Y. Kim, B.-S. Song, J. Grosspietsch, and S. F. Gillig, "A carry-free 54b b multiplier using equivalent bit conversion algorithm" IEEE J. Solid-State Circuits, vol. 36, no. 10, pp. 1538–1545, Oct. 2001.Y. He and C.

[8] B. Parhami, Computer Arithmetic Algorithms and Hardware Designs. New York: Oxford Univ. Press, 2000.

[9] V. Kantabutra, "A recursive carry-lookahead/carry-select hybrid adder," IEEE Trans. Comput., vol. 42, no. 12, pp. 1495–1499, Dec. 1993. B. Parhami, Computer Arithmetic Algorithms and Hardware Designs New York: Oxford Univ. Press, 2000.