International Journal of Electronics Signals and Systems

Volume 4 | Issue 2 Article 8

October 2014

CRBBE ALGORITHM FOR LOW POWER AND HIGH SPEED MULTIPLIER DESIGN

K. SANJEEVARAO
Dept. of E.C.E, Sir C.R.REDDY College of Engineering, Eluru, ksanjeevarao@gmail.com

A. RAMKUMAR
Dept. of E.C.E, Sir C.R.REDDY College of Engineering, Eluru, a_ramkumar11@yahoo.com

Follow this and additional works at: https://www.interscience.in/ijess

 Part of the Electrical and Electronics Commons

Recommended Citation

SANJEEVARAO, K. and RAMKUMAR, A. (2014) "CRBBE ALGORITHM FOR LOW POWER AND HIGH SPEED MULTIPLIER DESIGN," *International Journal of Electronics Signals and Systems*: Vol. 4 : Iss. 2 , Article 8.
Available at: https://www.interscience.in/ijess/vol4/iss2/8

This Article is brought to you for free and open access by Interscience Research Network. It has been accepted for inclusion in International Journal of Electronics Signals and Systems by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

CRBBE ALGORITHM FOR LOW POWER AND HIGH SPEED MULTIPLIER DESIGN

K.SANJEEVARAO[1], A.RAMKUMAR

[1]M. Tech VLSI Design, [2]M. Tech, Assistant Professor, Dept. of E.C.E, Sir C.R.REDDY College of Engineering, Eluru

Abstract- With the advent of the VLSI technology, designers could design simple chips with the more number of transistors. multipliers have large area, long latency and consume considerable power. Reduction of power consumption makes a device reliable. and The use of redundant binary (RB) arithmetic in the design of high-speed digital multipliers is beneficial due to its high modularity and carry-free addition, a high-radix-modified booth encoding algorithm is desired. However its use is hampered by the complexity of generating the hard multiples and the overheads resulting from negative multiples and normal binary(NB) to RB conversion. This paper proposes new RB booth encoding scheme to circumvent these problems. The idea is to polarize two adjacent booth encoded digits to directly from an RB partial product to avoid the hard multiple of high-radix booth encoding without incurring any correction vector, and the algorithm achieved high speed compared to existing multiplication algorithms for a gamut of power –of-to word lengths up to 64 b.

Index terms- *Arithmetic circuit, Booth encoding algorithm digital multiplier, redundant binary adder (RBA)*

I. INTRODUCTION

The digital multiplier is a obligatory and critical arithmetic unit in microprocessors, digital signal processors, multimedia application accelerators and emerging media processors.

It is also a kernel operator in application- specific data path of video and audio codec's, digital filters, computer graphics, and embedded systems. Two operations are essential in fast multiplier design namely the partial product generation and their accumulation. Compared with many other arithmetic operations, time-consuming and power hungry. The critical paths dominated by digital multipliers often speed limit on the entire design hence vlsi design of high-speed multiplication with low power dissipation is still popular research subject. Algorithms for speeding up multiplier work on the basic of reducing either the number of partial products or the time needed to accumulate them [RB) representation is one of the signed digit representations first introduced by A vizien is in 1961 for fast parallel arithmetic. This new arithmetic was applied for fast multiplication by Takagi et al and implemented in VLSI by Edematous.

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. The latter is required mainly for communicating the result to the peripheral devices which are largely designed based on the NB number system. The communications among RB adders across different stages of RB partial product summing tree are simpler than those of the full adders in a carry-save adder tree. In addition, the reduction rate of the redundant binary adder (RBA) summing tree is binary logarithmic to the number of RB partial products, which is particular beneficial to the generic power-of-two word size in computing. Booth encoder and partial product generator affect the efficiency of the partial product generation. The number of partial products that can be saved by this stage impacts the cost, performance, and power consumption of the RB summing tree and the multiplier as a whole. Although the number of partial products can be reduced with a high-radix Booth encoder, the number of hard multiples that are expensive to generate also increases simultaneously.

In order to overcome the overheads of existing Booth encoding algorithms, covalent redundant binary Booth encoding was used. The covalent redundant binary Booth multiplier circuits have been enhanced with a different RB coding and more efficient converters. The proposed method overcomes the hard multiple generation problem of NB Booth encoders without incurring any correction vector.

II. ISSUES OF BOOTH ENCODING ALGORITHM FOR RB MULTIPLICATION

In fast digital multiplier design, modified Booth encoding algorithm is an efficient way to reduce the number of partial products by grouping consecutive bits in one of the two operands to form the signed multiples.

The operand that is Booth encoded is called the multiplier and the other operand is called the multiplicand. Two major issues on using the modified Booth encoding algorithm for RB multiplication are:

A. HARD MULTIPLES PROBLEM

When modified Booth encoding is applied to two's complement number, it is known as normal binary Booth Encoding (NBBE). In radix-r Booth-k encoding ($r=2^k$), a signed digit c_i is generated from adjacent binary bits $y_{k(i+1)-1}y_{k(i+1)-2}\cdots y_{ki+1}y_{ki}$ and a borrow bit y_{ki-1} as follows:

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

$$\text{for } i = 0, 1, 2, \ldots \left\lceil \frac{N}{k} \right\rceil - 1$$

where k is an integer, $\lceil \alpha \rceil$ denotes the smallest integer value larger than or equal to α ,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 2^k+1, the number of hard multiples also increases simultaneously. A hard multiple refers to a multiple that is not a power of two and thus cannot be obtained easily by simple shifting and/or complementation.

The multiplier is partitioned into 4-b groups with an overlapping borrow bit between two adjacent groups. Each group is encoded in parallel to generate a select signal from the set

{±4M, , 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).

To speed up the generation of hard multiples in high-radix Booth encoding, a partially redundant biased Booth encoding (PRBBE) algorithm was used. The following figure depicts the generation and negation of 3X hard multiple.

It is generated in a partially redundant form by using a series of small length adders (4-b). The carry bit of each small length adder is not propagated but brought forward to the partial product summing tree.

However, when the 3X multiple is negated, both the sum and the carry vectors need to be complemented and a "1" is added at the least significant bit (LSB) position.

Therefore, the long strings of zeros between carries become strings of ones in the negative multiple

TABLE 1
RADIX-16 REDUNDENT BINARY BOOTH ENCODING (RBBE-4)

Multiplier Bits	Multiple		Multiplier Bits	Multiple	
	$+M$	$-M$		$+M$	$-M$
C D 0 C (0)	D	0	1 C D 0 (0)	C	$8M$
C D 0 C (1)	M	0	1 C D 0 (1)	M	$8M$
C D 0 1 (0)	M	0	1 C D 1 (0)	M	$8M$
C D 0 1 (1)	$2M$	0	1 C D 1 (1)	$2M$	$8M$
C D 1 C (0)	$2M$	0	1 C 1 0 (0)	$2M$	$8M$
C D 1 C (1)	$4M$	M	1 C 1 0 (1)	C	$5M$ *
C D 1 1 (0)	$4M$	M	1 C 1 1 (0)	C	$5M$ *
C D 1 1 (1)	$4M$	0	1 C 1 1 (1)	C	$4M$
C 1 0 C (0)	$4M$	0	1 1 D 0 (0)	C	$4M$
C 1 0 C (1)	$5M$ *	0	1 1 D 0 (1)	M	$4M$
C 1 0 1 (0)	$5M$ *	0	1 1 D 1 (0)	M	$4M$
C 1 0 1 (1)	$8M$	$2M$	1 1 D 1 (1)	C	$2M$
C 1 1 C (0)	$8M$	$2M$	1 1 1 0 (0)	C	$2M$
C 1 1 C (1)	$8M$	M	1 1 1 0 (1)	C	M
C 1 1 1 (0)	$8M$	M	1 1 1 1 (0)	C	M
C 1 1 1 (1)	$8M$	0	1 1 1 1 (1)	C	0

* Hard multiples.

B. Negative Multiples and NB-to-RB Partial products conversion Problem.

Since Negation in two's complement arithmetic requires carry propagation addition, negative partial product is more efficiently generated by the bit inversion of the multiplicand followed by the insertion of a "1" at its LSB position in the partial product summing tree. Therefore, one additional partial product row is generated to complete the two's complement negation of partial products for the negative multiples.

R=A+B=A-(-B)

Since $-B = \overline{B} + 1$,

$$\begin{aligned} R &= A - (\overline{B}+1) = A - \overline{B} - 1 \\ &= \left(-2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i\right) \\ &\quad - \left(-2^{n-1}\overline{b_{n-1}} + \sum_{i=0}^{n-2} 2^i \overline{b_i}\right) - 1 \\ &= -2^{n-1}(a_{n-1} - \overline{b_{n-1}}) + \sum_{i=0}^{n-2} 2^i (a_i - \overline{b_i}) - 1. \end{aligned}$$

TABLE II
POSITIVE-NEGATIVE COMPLEMENTCODING

Coding (r^+, r^-)	RB Digit r
(0, 0)	$\bar{1}$
(0, 1)	0
(1, 0)	0
(1, 1)	1

As shown in Table, an RB digit r can be encoded with two binary bits r^+ and r^- by

$$r = (r^+, r^-) = r^+ - \overline{r^-}$$

where $r^+, r^- \in \{0,1\}$, and $r \in \{0,1,\bar{1}\}$.

Therefore, according to above equation, the terms $(a_i - \bar{b_i})$ can be encoded as $r_i = (a_i, b_i)$. To eliminate the hardware required for sign extension, the most significant digit term can be simply negated as - (a_{n-1}, b_{n-1}).

$$-(r^+, r^-) = -r^+ + \overline{r^-} = (\overline{r^-}, \overline{r^+}).$$

C TWO"SCOMPLEMENT METHOD (TCM)

TCM is used to resolve the extra correction vector problem associated with the NBBE-2 algorithm. The TCM algorithm uses a divide-and-conquer approach to perform the two's complement conversion so that five signed partial products { 1X, } are originated for selection. In this way, the correction vector due to the negative multiples in two's complement arithmetic can be eliminated. If TCM is used for the design of RB multiplier, the RB coding induced compensation constants can also be similarly circumvented. With TCM algorithm, the RB multiplier achieves exactly $\lceil N/4 \rceil$ RB partial products as opposed to $\lceil N/4 \rceil + 1$ in NBBE-2 multiplier. Besides, the multiplier is modular and more regularly structured. However, the worst case delay of the TCM algorithm is logarithmically proportional to the operand lengths (OlogN).Comparing with the constant delay time of conventional Booth encoding algorithms, the dependency of speed on word length of TCM algorithm is a limiting factor for large integer multiplication.

D Redundant binary booth encoding

To overcome the problem of generating hard multiples in high-radix Booth encoding noticed that some hard multiples can be obtained by the differences of two simple (power of two) multiples the partial products so generated conform to the format of positive negative RB coding Covalent Redundant binary booth encoding (crbbe) Algorithm.

DESIGN OF CRBB-4 BAED RB MULTIPLIER

CRBBE-4 is composed of two adjacent radix-4 Booth encoders. Its gate-level implementation is represented, where the sign and magnitude of the radix-4 Booth encoded digit d_i are represented with three binary bits, sgn_i , $m_i^{(2)}$, $m_i^{(1)}$, and as follows:

$d_i = (-1)^{sgn(i)}(m_i^{(2)} + m_i^{(1)})$

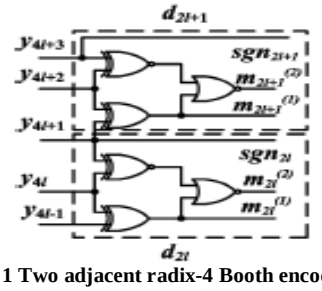


Fig 1 Two adjacent radix-4 Booth encoder

The above figure shows the 'l' th slice of a radix-16 CRBBE-4 circuit for the generation of the control signals c_lM_l. The indexes 'i' and 'l' are related by i=2l. The lower encoder takes three consecutive bits $y_{2i+1}y_{2i}y_{2i-1} = y_{4l+1}y_{4l}y_{4l-1}$ from the multiplier to generate the magnitude bits $m_{2l}^{(2)}$ and $m_{2l}^{(1)}$ of d_i Its sign bit $sgn_i = y_{4l+1}$. The upper encoder takes the binary bits $y_{2i+3}y_{2i+2}y_{2i+1} = y_{4l+3}y_{4l+2}y_{4l+1}$ and generates the magnitude bit $m_{2l+1}^{(2)}$ and $m_{2l+1}^{(1)}$ of d_{i+1}. Its sign bit $sgn_{i+1} = y_{4l+3}$. All of these output signals are mapped to the polarization circuit. The control signals c_lM_l, generated are used to select the RB partial products correspond to the multiples c_lX. [7]

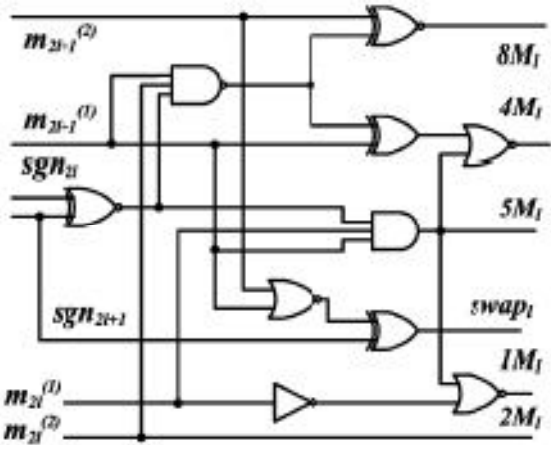


Fig 2 Polarization circuit

The polarization circuit performs the mapping (d_{i+1}, d_i) (p_l^+, p_l^-) .

The control signals $1M_l$, $2M_l$, $4M_l$ and $8M_l$ are computed as follows:

$$1M_l = m_{2l}^{(1)} \cdot \overline{5M_l}$$
$$2M_l = m_{2l}^{(2)}$$
$$4M_l = \overline{\left(m_{2l+1}^{(1)} \cdot m_{2l}^{(2)} \cdot (sgn_{2l+1} \odot sgn_{2l}) \odot m_{2l+1}^{(1)}\right)} \cdot \overline{5M_l}$$
$$8M_l = \overline{m_{2l+1}^{(1)} \cdot m_{2l}^{(2)} \cdot (sgn_{2l+1} \odot sgn_{2l}) \odot m_{2l-1}^{(2)}}.$$

The 5M multiple is generated as

$$5M_l = (sgn_{2l+1} \odot sgn_{2l}) \cdot m_{2l+1}^{(1)} \cdot m_{2l}^{(1)}.$$

The control flag, swap is used to exchange p_l^+ and p_l^- in the partial product generator to negate the selected RB partial product. When d_{i+1} is 0, the sign bit of d_{i+1} is complemented before it is used as an active high swap flag to the RBPPG. Otherwise, the original sign of d_{i+1} is used as the swap flag. Therefore, the swap signal can be generated by:

$$swap_l = \overline{\left(m_{2l+1}^{(1)} + m_{2l+1}^{(2)}\right)} \oplus sgn_{2l+1}$$

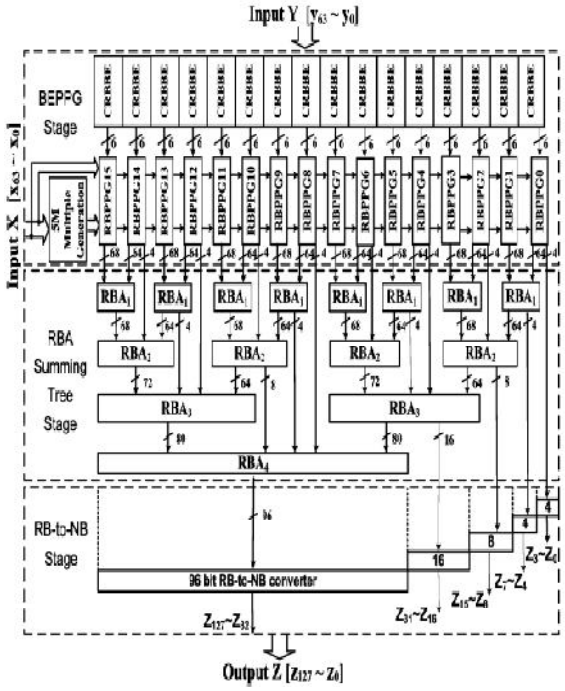


Fig 3 Block diagram of 6*64 bit RB multiplier

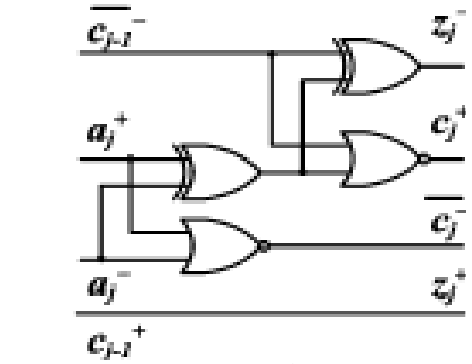


Fig 4 RB half adder

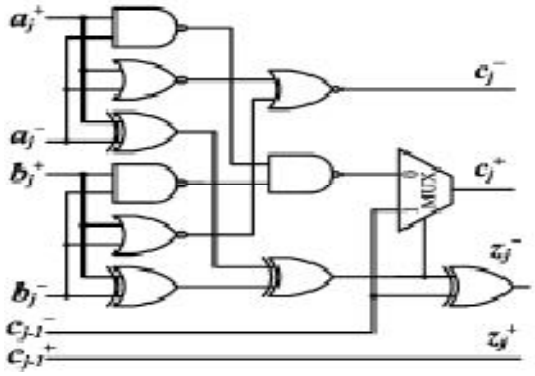


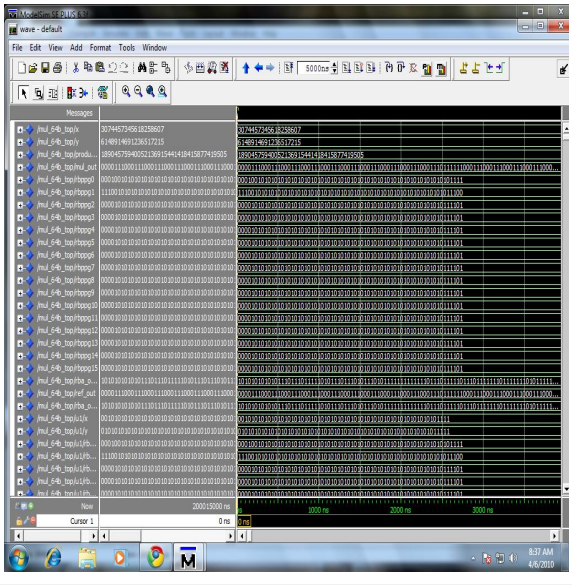
Fig 5 RB full adder

Architecture of CRBBE-4 Based RB Multiplier. This section exemplifies the use of CRBBE-4 for the design of a 64*64 bit CRBBE-4 Based multiplier. Which consists of three stages, booth encoder and RBPPG, RBA summing tree and R- to-NB converter? In the first stage Booth encoder and partial product generator affect the efficiency of the partial product generation. The number of partial products that can be saved by this stage impacts the cost, performance, and power consumption of the RB summing tree and the multiplier as a whole. In the first stage, 16 CRBBE-4 slices are used to generate the control signals from the multiplier. The hard multiple 5X is generated. The multiplicand bits are shifted and selected into 16 rows of RB partial products in 16 slices of RBPPG.

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 RBA block in the i-th level, designated RBA_i (i=1 to 4) contains 2^{i+1} RBHA cells in its most significant digit positions. Due to the positive-negative-complement coding, the second binary bit $pp_{1,j}^-$ of the RB partial product generated from CRBBE-4 and RBPPG circuit should be inverted before it is input to the RBA. A preprocessing circuit is needed for each RB digit to avoid the inconsistent representations of "0" prior to the RBA summing tree stage. An important benefit of the coding format adopted in this design is that these preprocessing circuits can be completely eliminated due to its symmetry.

An RB-to-NB converter converts the final accumulation result to NB representation. Due to the unequal delay profile of the final RB result bits, the conversion can be carried out in uneven groups of consecutive digits according to their arrival time. Groups of 4, 4, 8, 16 and 96 digits from the least significant digit position are evaluated concurrently. The first three groups of 4, 4, and 8 digits can be independently converted with ripple-carry adders to reduce the circuit complexity. The carry generation of the next group of 16 digits can be evaluated with a carry-look ahead adder as they do not depend on the final summation results in the RBA tree stage.

SIMULATION REPORTS:



CONCLUSION

Hence, a high-speed and energy-efficient RB multiplier is designed based on new covalent RB Booth encoding algorithm. The idea is to polarize two adjacent Booth-encoded digits into a differential pair to restore the effective RB partial product reduction rate without the NB-to-RB conversion overhead. This method fully exploits the characteristics of the positive–negative complement coding of RB number to directly generate an RB partial product from two adjacent Booth-encoded digits. Consequently, it shares the same advantages of RB Booth encoder for the ease of generating hard multiples and avoidance of error compensation vector, the two problems that are confronted by RB multiplier with normal binary Booth encoding.

REFERENCES:

[1] 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.

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

[3] G. W. Bewick, "Fast multiplication: Algorithms and implementation," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1994.

[4] O. L. MacSorley, "High-speed arithmetic in binary computers," IRE Proc., vol. 49, pp. 67–91, Jan. 1961.

[5] P.Kornerup, "Digit-set conversions: Generalizations and applications," IEEE Trans. Comput., vol. 43, no. 5, pp. 622–629, May 1994.

[6] 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.

[7] 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.

[8] 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.

A.RAM KUMAR Mtech(VLSISD) M. Tech. degree in swarnandhra college of engineering and technology narasapur, Andhra Pradesh, India . Presently he is working as assistant professor in Department of Computer Science & Engineering, in Sir C R Reddy College of Engineering Eluru, Andhra Pradesh, India.

K. Sanjeevarao received the B.Tech Degree in Electronics and Communication Engineering from ANU in 2009. Pursuing M.Tech in SIR C.R.R COLLEGE.

◈◈◈