A NOVEL COVALENT REDUNDANT BINARY BOOTH ENCODER

Yajuan He, Chip-Hong Chang, Jiangmin Gu

Centre for High Performance Embedded Systems,
Nanyang Technological University,
50 Nanyang Drive, Singapore 637553

Hossam A. H. Fahmy

Electronics and Communications
Department, Faculty of Engineering,
Cairo University, Giza 12613, Egypt

Abstract— **The benefit of high radix Booth encoders in reducing the number of partial products in fast multipliers has been hampered by the complexity of generating the hard multiples. The use of redundant binary (RB) Booth encoder can overcome this problem and avoid the error compensation vector but at the cost of doubling the number of RB partial products. This paper presents a novel covalent RB Booth encoder to generate a compound RB partial product from two adjacent Booth encoded digits. The new encoder fully exploits the characteristics of Booth encoded numbers to restore the effective partial product reduction rate of RB Booth encoder while maintaining the simplicity of hard multiple generators and eliminating the constant correction vector. A legitimate comparison on an 8×8-bit RB multiplier prototype shows that the multiplier constructed with our proposed Booth encoder consumes lower power and computes faster than those with the normal binary and redundant binary Booth encoders.**

I.INTRODUCTION

Digital multiplier is an obligatory and critical arithmetic unit in microprocessors, digital signal processors and multimedia application accelerators. Two operations are essential in fast multiplier design, namely the partial product generation and their accumulation. Algorithms for speeding up multiplier work on the basis of reducing either the number of partial products or the time needed to accumulate them. Attempts have been made to employ Redundant Binary (RB) number [1][2] as an internal format for partial product accumulation. The carry-free addition allows the partial products to be reduced at a rate of 2:1 using the RB adders as oppose to the reduction rate of 3:2 with binary carry-save adders. Moreover, the regular structure of the RB summing tree makes RB multipliers amendable to VLSI layout. Although the RB summing tree is in general more costly to implement in the entire RB multiplier design, the Booth encoder and the RB Partial Product Generator (PPG) determine how efficient these RB partial products are generated, and contribute indirectly to the performance and cost of the multiplier.

Traditionally, Booth-2 encoding, also known as the modified Booth algorithm, is employed to reduce the number of additions in the summation network by half. Although the number of partial products can be reduced aggressively with high radix Booth encoder, the number of hard multiples and the difficulty in their generation increases commensurately with the radix. As Booth encoding adds delay in the partial product generation and reduces the delay in their summation, there is a bound on the radix of Booth encoder to be beneficial. To extend this limit, a redundant binary signed-digit Booth encoder was proposed by N. Besli [3] to simplify the generation of hard multiples and the negation of the multiples. It turns out that the scheme actually devastates the partial product reduction rate. As the circuit for each digit of the RB partial product will be duplicated in a large number, considerably more hardware is incurred.

In this paper, we present a novel Covalent Redundant Binary Booth Encoder (CRBBE) whereby a compound RB partial product is generated from two adjacent Booth encoded digits. In this encoder, the signs of the encoded digits are ignored at the outset and only the absolute values of the coefficients need to be produced. Hard multiples are generated with similar simplicity as the RB Booth encoder of [3]. Therefore, the hardware resource used to build this circuit is greatly reduced. Compared with the normal binary Booth encoder, the proposed CRBBE gets rid of the correction vector [2] while having the same partial product reduction rate for the same radix of Booth encoding.

II.PRELIMINARIES

A. Redundant Binary Number Representations

A Redundant Binary (RB) number is a special subset of the generalized signed digit number representation [4]. It consists of digits from the set {−1, 0, 1}. Symbolically, a RB digit can be represented in a positive-negative coding format with the notation (X^+, X^-). For binary X^+ and X^-, the RB digit, R can be calculated as:

$$R=X^+-X^- \tag{1}$$

which yields the coding style defined in TABLE I.

The addition of two n-bit 2's complement numbers, A and B can be expressed as

$$A+B=A-(-B)=A-(\bar{B}+1)=(A,\bar{B})+(0,1). \tag{2}$$

0-7803-8834-8/05/$20.00 ©2005 IEEE.

TABLE I. CODING FORMAT FOR RB NUMBER

X^+	X^-	RB digit, R
0	0	0
1	0	1
0	1	$\bar{1}$
1	1	0

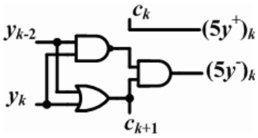


Figure 1. RB adder for generating the 5*M* hard multiple [3]

The composite number $(A,\overline{B})$ can be interpreted as a RB number. Let a_i and b_i be the i^{th} bits of A and B, respectively. Equation (2) implies that the addition can be evaluated by first inver-ting all bits of B and then grouping each inverted bit with the corresponding bit of A to form a RB number before the constant digit $\bar{1}$, i.e., (0, 1) is added to its least significant digit (LSD). It should be noted that the weight of the most significant digit (MSD) of a RB number is 2^{n-1}, while that of a 2's complement number is $-(2)^{n-1}$. Therefore, the MSD of the RB number can be rewritten as ($\overline{b_{n-1}}$, a_{n-1}) to avoid the sign extension problem in the partial product accumulation of redundant binary multiplication.

B. Booth Encoding

1) Normal binary Booth encoding

The Booth algorithm [5] is an efficient way to reduce the number of partial products by grouping the bits of the operand called multiplier to form signed numerals. As the radix value, 2^n (n=1, 2, 3...) of the Booth-n encoding increases, the number of encoded Booth digits for the multiplier decreases by a factor to $1/n$, so does the number of partial products. It might appear that choosing the highest possible radix Booth algorithm according to the bit-width of the multiplier results in a fastest multiplier. However, a close examination revealed that the number of multiples also increased commensurately with the radix to $2n+1$, so did the number of hard multiples [6], which referred to a multiple that is not a power of two and thus cannot be obtained easily by simple shifting and/or complementation. As a result, some carry propagation adders are needed to generate the hard multiples. Since the generation of the partial products is not accomplished until all hard multiples are produced, the latency of the multiplier increases. Therefore, there is hardly any advantage for Booth-3 encoding and above due to the difficulty of generating the hard multiples and the complex decoding logic.

On the other hand, these limitations can be lifted in RB system up to Booth-4 encoding by exploiting the redundancy of RB number. To differentiate between the two types of Booth encoding, we call the former normal binary Booth encoding (NBBE) and the latter, which generates the redundant binary signed digit, the redundant binary Booth encoding (RBBE).

TABLE II. REDUNDANT BINARY BOOTH-4 ENCODING

Normal Binary	Multiple		Normal Binary	Multiple	
b_{i+3} b_{i+2} b_{i+1} b_i $b_{i\text{-}1}$	$+M$	$-M$	b_{i+3} b_{i+2} b_{i+1} b_i $b_{i\text{-}1}$	$+M$	$-M$
0 0 0 0 (0)	0	0	1 0 0 0 (0)	0	8*M*
0 0 0 0 (1)	*M*	0	1 0 0 0 (1)	*M*	8*M*
0 0 0 1 (0)	*M*	0	1 0 0 1 (0)	*M*	8*M*
0 0 0 1 (1)	2*M*	0	1 0 0 1 (1)	2*M*	8*M*
0 0 1 0 (0)	2*M*	0	1 0 1 0 (0)	2*M*	8*M*
0 0 1 0 (1)	4*M*	*M*	1 0 1 0 (1)	0	5*M* *
0 0 1 1 (0)	4*M*	*M*	1 0 1 1 (0)	0	5*M* *
0 0 1 1 (1)	4*M*	0	1 0 1 1 (1)	0	4*M*
0 1 0 0 (0)	4*M*	0	1 1 0 0 (0)	0	4*M*
0 1 0 0 (1)	5*M* *	0	1 1 0 0 (1)	*M*	4*M*
0 1 0 1 (0)	5*M* *	0	1 1 0 1 (0)	*M*	4*M*
0 1 0 1 (1)	8*M*	2*M*	1 1 0 1 (1)	0	2*M*
0 1 1 0 (0)	8*M*	2*M*	1 1 1 0 (0)	0	2*M*
0 1 1 0 (1)	8*M*	*M*	1 1 1 0 (1)	0	*M*
0 1 1 1 (0)	8*M*	*M*	1 1 1 1 (0)	0	*M*
0 1 1 1 (1)	8*M*	0	1 1 1 1 (1)	0	0

2) Redundant binary Booth encoding

To overcome the problem of generating hard multiples in high-radix Booth encoding, N. Besli et al. noticed that some hard multiples can be obtained by the differences of two simple (power-of-two) multiples [3]. The partial products so generated conform to the format of positive-negative RB coding. This distinguishing Booth encoding logic is RB Booth encoding. TABLE II illustrates the RB Booth-4 encoding, where the original hard multiples of ±3*M*, ±6*M* and ±7*M* are replaced by ±(4*M*−*M*), ±(8*M*−2*M*) and ±(8*M*−*M*), respectively. The only exception is the hard multiple 5*M*, which cannot be readily produced in this manner. Therefore, additional hardware is necessary to generate this 5*M* multiple. Fig. 1 shows a simple RB adder to add 4*M* and 1*M*. It turns out that this RB adder is carry-free and does not lie in the critical path of the RB Booth-4 encoder and PPG circuit. Compared to NBBE, the ease of generating the hard multiples in RBBE has been offset, to certain extend, by its complex circuitry involving the use of high fan-in gates. In addition, the cost of high fan-in gates and their associated detriments are aggravated by the duplication of each digit in the RB partial products [3].

III. DESIGN OF COVALENT RB BOOTH ENCODER

Using RBBE, a pair of coefficients, (p_i^+, p_i^-) is generated from the bits of the multiplier. It is used to form the i^{th} RB partial product of the multiplicand [3]. The hard multiple problem is circumvented at the expense of doubling the number of RB partial products. This is because only one Booth encoder is used to generate one RB partial product as opposed to the use of NBBE where two binary partial products can be grouped to form a single RB partial product. Half of the binary bits representing the RB partial product generated from the simple multiple in the RBBE encoding are filled with "0"s, which is very inefficient. In this section, a novel covalent redundant binary Booth encoding (CRBBE) algorithm is proposed. It utilizes the characteristics of Booth encoded numbers to generate a reduced number of RB partial products.

TABLE III. PERMISSIBLE (d_{i+1} d_i) IN BOOTH-1 ENCODING

d_{i+1}=1	d_{i+1}=0	d_{i+1}=$\bar{0}$	d_{i+1}=$\bar{1}$
1$\bar{0}$	0 1	$\bar{0}\bar{0}$	$\bar{1}$1
1$\bar{1}$	0 0	$\bar{0}\bar{1}$	$\bar{1}$0

TABLE IV. MODIFIED DUPLET (d_{i+1} d_i) IN BOOTH-1 ENCODING

d_{i+1}=1	d_{i+1}=0	d_{i+1}=$\bar{0}$	d_{i+1}=$\bar{1}$
1$\bar{0}$ = (2, 0)	01=$\bar{0}$1=(1, 0)	$\bar{0}\bar{0}$=0$\bar{0}$=(0, 0)	$\bar{1}$1 = (1, 2)
1$\bar{1}$ = (2, 1)	00=$\bar{0}$0=(0, 0)	$\bar{0}\bar{1}$=0$\bar{1}$=(0, 1)	$\bar{1}$0 = (0, 2)

The idea of CRBBE is to produce a RB partial product equivalent to two NB partial products generated from a pair of adjacent Booth encoders. This is possible provided that the two adjacent Booth encoders always generate signed digit coefficients of opposite signs so that their correspondding partial products can be combined to form a single positive-negative coded RB partial product. A special RB Booth encoder can be designed to generate a compound RB partial product from two Booth encoded digits. We call it covalent RBBE for its analogy to the way a covalent compound is formed from charge sharing. To introduce the idea, we start by Booth-1 encoding, where the multiplier is encoded to exhibit the following properties:

Property 1: No two consecutive non-zero digits are of the same sign, i.e, $d_{i+1}d_i = -1 \; \forall \; i \in [0,n]$, where d_{i+1} and d_i are two nearest non-zero digits and n is the width of the RB partial product. It ensures that the signs of the nonzero digits alternate in the encoded multiplier. For example, $0100\bar{1}0$ is legal while 010010 and $0\bar{1}00\bar{1}0$ are illegal.

Property 2: Any zero between a leading 1 and a trailing $\bar{1}$ is substituted by a negative zero $\bar{0}$. For example, $01\bar{0}\bar{0}\bar{0}\bar{1}001\bar{0}\bar{1}0$ is a legal format.

According to the above properties, not all combinations of contiguous digit pairs are permissible in the encoded numbers. TABLE III depicts all possible combinations, grouped into four categories based on the left digit d_{i+1}. From the definition of Booth encoding, the coefficient of the multiple, $p_i^+ - p_i^- = 2d_{i+1}+d_i$. Since the numeral 0 is neutral and equivalent to $\bar{0}$, we can reformat the duplets in TABLE III such that one digit of the pair is positive and the other is negative, as shown in TABLE IV. In TABLE IV, the numerals in bracket represent the coefficients, (p_i^+, p_i^-). The shaded cells represent the positive-negative pairs while the other cells represent the negative-positive pairs.

The above CRBBE is based on Booth-1 encoding, where the two contiguous digits, d_{i+1} and d_i, are mapped from two contiguous bits $b_{i+1}b_i$ and b_ib_{i-1} of the multiplier, respectively. Thus, the pair of coefficients, (p^+_i, p^-_i) is a compound RB coefficient. Similar to the simple RB coefficient, negative compound coefficient can be obtained from its positive counterpart by simply swapping the values of p^+_i and p_i^-. The sign of the MSD d_{i+1} is used to determine if the swapping of p^+_i and p_i^- is necessary. The criterion for the ordering of the coefficients is given by (3).

TABLE V. PROPOSED CRBBE-2 DUPLETS ($d_{i+1}d_i$)

d_{i+1}=2	d_{i+1}=1	d_{i+1}=0	d_{i+1}=$\bar{0}$	d_{i+1}=$\bar{1}$	d_{i+1}=$\bar{2}$
2$\bar{0}$	12=2$\bar{2}$	02=$\bar{0}$2	$\bar{0}\bar{0}$=0$\bar{0}$	$\bar{1}$2	$\bar{2}$2
2$\bar{1}$	11*	01=$\bar{0}$1	$\bar{0}\bar{1}$=0$\bar{1}$	$\bar{1}$1	$\bar{2}$1
2$\bar{2}$	10=1$\bar{0}$	00=$\bar{0}$0	$\bar{0}\bar{2}$=0$\bar{2}$	$\bar{1}$0	$\bar{2}$0
	1$\bar{0}$			$\bar{1}\bar{0}$=$\bar{1}$0	
	1$\bar{1}$			$\bar{1}\bar{1}$*	
	1$\bar{2}$			$\bar{1}\bar{2}$=$\bar{2}$2	

$$\left(p_i^+, p_i^-\right) = \begin{cases} (2\cdot|d_{i+1}|, |d_i|) & \text{if } d_{i+1} > 0 \quad \text{(pos-neg pair)} \\ (|d_i|, 2\cdot|d_{i+1}|) & \text{if } d_{i+1} < 0 \quad \text{(neg-pos pair)} \end{cases} \tag{3}$$

With some restriction on the legitimacy of the encoded digits, two contiguous digits, $d_{i+1}d_i$, of Booth-2 encoding can also be mapped from three contiguous bits $b_{2i+3}b_{2i+2}$ b_{2i+1} and $b_{2i+1}b_{2i}b_{2i-1}$ of the multiplier to generate the compound coefficients for a RB partial product. The contiguous digits, $d_{i+1}d_i$, can be similarly rewritten as positive-negative or negative-positive pairs except for the cases of 11 and $\bar{1}\,\bar{1}$, which correspond to the hard multiples ±5M as shown in TABLE V. From the shaded columns of TABLE V, the coefficient reordering criterion is given by (4).

$$(p_i^+, p_i^-) = \begin{cases} (4\cdot|d_{i+1}|, |d_i|) & \text{if } d_{i+1} > 0\text{, except } 11 \\ (|d_i|, 4\cdot|d_{i+1}|) & \text{if } d_{i+1} < 0\text{, except } \bar{1}\bar{1} \end{cases} \tag{4}$$

Fig. 2(a) shows the CRBBE-2 circuit which is composed of two adjacent Booth-2 encoders. The lower encoder is fed from the binary bits $b_{2i+1}b_{2i}b_{2i-1}$ of the multiplier and generates the control signals, $1m_i$ and $2m_i$, and a sign bit, sgn_i taken directly from the MSB, b_{2i+1}. The upper encoder is fed from the bits $b_{2i+3}b_{2i+2}b_{2i+1}$, and generates the signals, $1m_{i+1}$, $2m_{i+1}$, and a sign bit, $sgn_{i+1} = b_{2i+3}$. All these output signals will be reformatted according to TABLE V before they are passed to the RB PPGs. The reformatting circuit is shown in Fig. 2(b). From (4), the sign of d_{i+1} decides the value of p_i^+ and p_i^-. When d_{i+1} is zero, its sign bit is complemented before it is used as an active high swap flag. Otherwise, the original sign is used as the swap flag. An active low swap flag is also generated for the swapping circuit of the RB PPG. Therefore, we have (5).

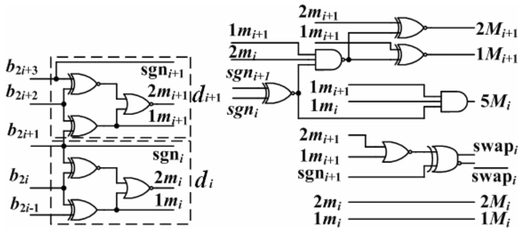
To convert the duplets, 12 to $2\bar{2}$ or $\bar{1}\bar{2}$ to $\bar{2}2$, the output signals of $2m_{i+1}$ and $1m_{i+1}$ from the upper encoder are to be complemented in order to convert |1| to |2| when the contiguous digits are of the same sign and when both signals $1m_{i+1}$ and $2m_i$ are active. For all other duplets, the output signals $2m_{i+1}$ and $1m_{i+1}$ retain their original values. The converted signals, $2M_{i+1}$ and $1M_{i+1}$ of the upper encoder are given by (6) and (7). The control signal for the duplets 11 and $\bar{1}\,\bar{1}$ corresponding to the special ±5M multiples marked * in TABLE V can be generated by (8).

$$\text{swap}_i = (1m_{i+1} + 2m_{i+1}) \odot \text{sgn}_{i+1} \tag{5}$$

$$2M_{i+1} = \overline{\left(1m_{i+1}\cdot 2m_i\cdot(\text{sgn}_i \odot \text{sgn}_{i+1})\right)} \odot 2m_{i+1} \tag{6}$$

$$1M_{i+1} = \overline{\left(1m_{i+1}\cdot 2m_i\cdot(\text{sgn}_i \odot \text{sgn}_{i+1})\right)} \odot 1m_{i+1} \tag{7}$$

$$5M = (\text{sgn}_{i+1} \odot \text{sgn}_{i+1}) \cdot 1m_{i+1} \cdot 1m_i \tag{8}$$



(a) covalent Booth-2 encoder (b) reformatting circuit

Figure 2. CRBBE-2 encoder

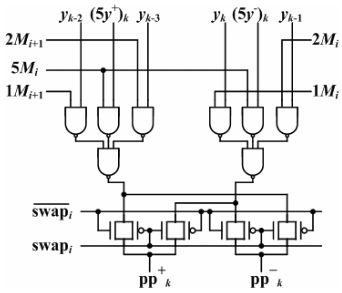


Figure 3. RB Partial Product Generator (PPG) for CRBBE-2

Fig. 3 shows a slice of the PPG circuit for generating one RB digit in CRBBE-2. The logic gates in the input stage are realized with complementary CMOS logic style to reduce their loading to the CRBBE-2 circuits. Since a large number of such input stages are connected in parallel, strong driving capability is required from the outputs of the encoders. The output stage of the RB PPG circuit is implemented in transmission gate style with unity fan-out. It drives the CMOS input stage of the redundant binary adder (RBA) in the partial product summing tree.

IV.SIMULATION RESULTS

An 8×8-bit RB multiplier is used as a vehicle to compare the performances of our proposed CRBBE and other Booth encoders. For a legitimate comparison, the radices of different Booth encoders are chosen based on the same number of RB partial products generated. Therefore, we compare the simulated worst case delay, power dissipation and transistor count of three RB multipliers employing CRBBE-2, NBBE-2 and RBBE-4, respectively. The same RBA summing tree and RB-to-NB converter circuits of [2] are used for these multipliers.

The simulation was carried out using HSIM in TSMC 0.18μm technology at 1.8V. 4096 randomly generated data with the input rate of 100MHz is applied to the circuits. The simulation results are listed in TABLE VI. From TABLE VI, it is evident that our proposed CRBBE multiplier outperforms the others. The RBBE one consumes an excessive number of transistors due to its inefficient composition of PPG. The high transistor count also degrades its power dissipation. Although the transistor count of our multiplier is slightly higher than that of NBBE multiplier, almost all the surplus transistors are located in the input stage of the PPG circuit. This part of the circuit consumes only a small fraction of power of the entire multiplier.

TABLE VI SIMULATION RESULTS OF 8×8-bit RB MULTPLIERS

8×8-bit Multiplier	Transistor Count	Power(mW) @ 100MHz	Delay(ns)	PDP(pJ)
CRBBE-2	2628	0.45	2.25	1.01
NBBE-2[2]	2592	0.48	2.72	1.31
RBBE-4[3]	4172	0.66	2.43	1.60

It is worth noting that the results vary as the width of the operands change. For an n×n-bit multiplier, the number of RB partial products generated from NBBE-2, RBBE-4 and CRBBE-2 are $[n/4]+1$, $[n/4]$ and $[n/4]$, respectively. In NBBE multiplier, a correction vector must be generated to compensate for the aggregate errors resulting from both RB coding and the Booth encoding [2]. Since most DSPs work on power-of-two operands, this correction vector will increase the number of stages of the summing tree and incur additional hardware for its accumulation. Consequently, the power dissipation and worst case delay are degraded. In this simulation, the RBA tree of the CRBBE multiplier has one row lesser than that of the NBBE multiplier, resulting in some saving in power and delay. If the bit length of the multiplier is exactly 2^k-1, the NBBE multiplier circuit may become comparable or even surpass since the correction vector will have little effect on the depth of the RBA summing tree.

V.CONCLUSION

In this paper, a novel covalent redundant binary Booth encoder is presented. The design of this new encoder fully exploits the characteristics of Booth encoded numbers to generate a compound RB partial product from two adjacent Booth encoded digits. Consequently, it provides the same advantages of RB Booth encoder for the ease of generating hard multiples and avoidance of error compensation vector, the two problems that generally confronted by NB Booth encoding multiplier. Additionally, the proposed CRBBE transcend RBBE in its ability to reduce the total number of RB partial products by half for the same radix.

REFERENCES

[1] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, "A 33 MFLOPS floating point processor using redundant binary representation," in *Proc. Int. Solid-State Circuits Conf.,* pp. 152-153, 342-343, Feb. 1988.

[2] H. Makino, et al., "An 8.8-ns 54×54-bit multiplier with high speed redundant binary architecture," *IEEE J. Solid-State Circuits,* vol. 31, no. 6, pp. 773-783, Jun. 1996.

[3] N. Besli, and R.G. Deshmukh, "A novel redundant binary signed-digit (RBSD) Booth's encoding," in *Proc. IEEE Southeast Conf.,* pp. 427-431, Apr. 2002.

[4] B. Parhami, "Generalized signed-digit number systems: A unifying framework for redundant number representations," *IEEE Trans. Computers,* vol 39, pp. 89–98, Jan. 1990.

[5] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Computers,* Sep. 1961.

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