

Complex Number Representation in RCBNS Form for Arithmetic Operations and Conversion of the Result into Standard Binary Form

Hatim Zaini and R. G. Deshmukh
 Florida Institute of Technology
rgd@ee.fit.edu

ABSTRACT

This paper introduces a novel method for complex number representation. The proposed Redundant Complex Binary Number System (RCBNS) is developed by combining a Redundant Binary Number and a complex number in base $(-1+j)$. Donald [1] and Walter Penny [2,3] represented complex numbers using base $-j$ and $(-1+j)$ in the classified algorithmic models. A Redundant Complex Binary Number System consists of both real and imaginary-radix number systems that form a redundant integer digit set. This system is formed by using complex radix of $(-1+j)$ and a digit set of $\alpha = 3$, where α assumes a value of $-3, -2, -1, 0, 1, 2, 3$. The arithmetic operations of complex numbers with this system treat the real and imaginary parts as one unit. The carry-free addition has the advantage of Redundancy in number representation in the arithmetic operations. Results of the arithmetic operations are in the RCBNS form. The two methods for conversion from the RCBNS form to the standard binary number form have been presented. In this paper the RCBNS reduces the number of steps required to perform complex number arithmetic operations, thus enhancing the speed.

Keywords: Complex Binary number, Redundant Binary Number, Redundant Complex Binary Number System, Addition and Subtraction.

1. INTRODUCTION

Algorithms in complex orthogonal transformations, correlations, and filtrations are involved in arithmetic computations (e.g. geometric analysis in graphics or image processing). These applications require efficient representations and treatment of complex numbers. In today's computers involving complex numbers, the complex operations use the real and imaginary parts separately and then accumulate their individual results to obtain the final result. For example, the natural way of multiplication of two complex numbers requires four real multiplications and one real addition and one real subtraction. To convert complex number in the RCBNS form, the real part and the imaginary part of the given

number must be treated as one unit. This will increase the speed of arithmetic operations due to a decrease in the number of steps involved in these operations.

The complex Binary Number System (CBNS) method uses a base of $(-1+j)$ instead of a base 2 normally used in the Standard Binary Number (SBN). There is a unique representation for each complex number in the new base of $(-1+j)$. The value of an n -bit binary number in the CBNS form with the base $(-1+j)$ can be written in the form of a power series as shown below.

$$\dots\dots+a_2(-1+j)^2+a_1(-1+j)^1+a_0(-1+j)^0 \quad (1)$$

The coefficients 'a' assume a value of 0 or 1. The powers of $(-1+j)$ are grouped in a row of 4-bit positions as shown in Table 1. Each row labeled as q_i is generated by multiplying the current row by -4 . In other words, the following expression can be used to generate a row of 4 bits: $(q_{0,j}=q_{0,3} \ q_{0,2} \ q_{0,1} \ q_{0,0} = (2+j2) \ (-j2) \ (-1+j) \ 1)$, where $q_{i+1,j} = q_{i,j} * (-4)$. For example, multiplying row 0 by -4 generates the 1st row and multiplying the 1st row by -4 generates the 2nd row and so on.

In 1988, T. T. Dao attempted to define specific arithmetic operations in radices ± 4 by improving an algorithm to generate several bits [4]. The focus is on the CBNS form where $(-1+j)$ is the minimum possible radix in the definition of CBNS. Table 1 shows the weights of $(-1+j)^i$ in a complex plane and this is closely related to the radix -4 Sign Digit (SD) number system. Therefore, -4 is an element of the set of the CBNS of some power and it is efficient to work with the Redundant Binary Sign Digit numbers. The RCBNS is a positional number system that has a complex radix and uses a digit set $\{-\alpha \text{ to } \alpha\}$ that allows for carry free additions. The combination of the two algorithms, the redundant binary numbers system [4, 5] and the binary number system for the complex numbers [1, 2] results in the Redundant Complex Binary Number System (RCBNS).

$(-1+j)^{11}$		$(-1+j)^{10}$		$(-1+j)^9$		$(-1+j)^8$		Row 2 = q_2
R	J	R	J	R	J	R	J	Row 1 * (-4)
32	32	0	-32	-16	16	16	0	
$(-1+j)^7 = q_{1,3}$		$(-1+j)^6 = q_{1,2}$		$(-1+j)^5 = q_{1,1}$		$(-1+j)^4 = q_{1,0}$		Row 1 = q_1
R	J	R	J	R	J	R	J	Row 0 * (-4)
-8	-8	0	-8	4	-4	-4	0	
$(-1+j)^3 = q_{0,3}$		$(-1+j)^2 = q_{0,2}$		$(-1+j)^1 = q_{0,1}$		$(-1+j)^0 = q_{0,0}$		Row 0 = q_0
R	J	R	J	R	J	R	J	
2	2	0	-2	-1	1	1	0	

Table 1. Real and imaginary values with radix $(-1+j)$

In this paper the representation of RCBSN is introduced for the radix $r = (-1+j)$ and a digit set, D_α where $\alpha = 3$ and D_α has the digit set of $(-3, -2, -1, 0, 1, 2, 3)$. Alpha (α) is restricted to $[(r-1)/2] \leq \alpha \leq (r-1)$ where $r = 4$. Redundancy provides carry-free addition in arithmetic operations. The value of a number, $X = (x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ in the RCBNS of base $(-1+j)$ and $\alpha=3$ can be shown by the following expression.

$$X = \sum_{i=0}^{n-1} x_i (-1+j)^i \quad X_i = \{-\alpha, \dots, \alpha\} \quad (2)$$

Figure 1 shows a block diagram for the complex number conversion, followed by the arithmetic operations (additions and subtractions only), and then obtaining the results in the RCBNS form and finally converting the RCBNS result into a regular binary number.

In the next section, the basic properties of the RCBNS Representation are introduced with their basic arithmetic operations.

2. REDUNDANT COMPLEX BINARY NUMBER SYSTEM REPRESENTATION

The redundancy in the RCBNS representation using radix $(-1+j)$ allows the conversion of any complex number without any restrictions. Any complex number is represented by a row of 4 bits in the radix 4. To simplify the conversion from a binary number to a base 4 RCBNS form, a range of $-3-j3$ through $3+j3$ has been used for the representation. A portion of the representation is shown in Table 2.

3. COMPLEX NUMBER CONVERSION IN THE TERM OF RCBNS FORM

Conversions of the complex numbers to the RCBNS form is achieved by converting the complex number into

the binary form for the real and imaginary parts and then by using the following steps:

Step 1: Check the sign of both the real and imaginary parts. If each is positive or negative, then for the positive numbers place a 0 in front of each of 2-bit digit (e.g. for $9 = 10\ 01$, then 010 and 001), and similarly for the negative numbers place 1 in front of each of 2-bit digit (e.g. for $-9 = -10\ 01$, then 110 and 101).

Step 2: Get 3 bits of each part and combine them together with the LSB 3-bit of the imaginary part catenated to the 3 bits of real part and then put them in the order of q_i .

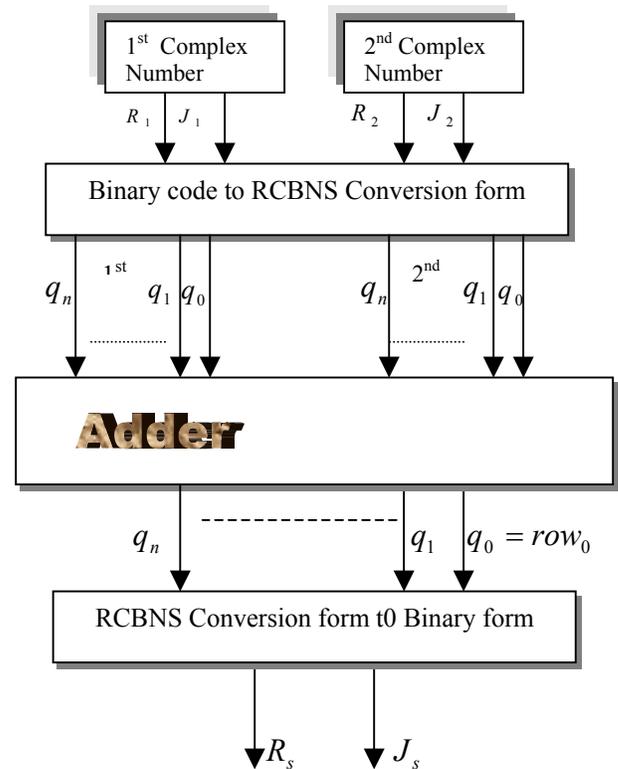


Figure 1. Block diagram for the proposed algorithm

No	$(-1+j)^3$	$(-1+j)^2$	$(-1+j)^1$	$(-1+j)^0$
	$2+j2$	$-j2$	$-1+j$	1
-3-j3	-1	1	1	0
-2-j3	0	1	-1	-3
-1-j3	0	1	-1	-2
0-j3	0	1	-1	-1
1-j3	0	1	-1	0
2-j3	0	1	-1	1
3-j3	0	1	-1	2
-3+j3	0	-1	1	-2
-2+j3	0	-1	1	-1
-1+j3	0	-1	1	0
0+j3	0	-1	1	1
1+j3	0	-1	1	2
2+j3	1	-1	-1	-1
3+j3	1	-1	-1	0
-3-j2	0	1	0	-3
-2-j2	0	1	0	-2
-1-j2	0	1	0	-1
0-j2	0	1	0	0
1-j2	0	1	0	1
2-j2	0	1	0	2
3-j2	0	1	0	3

Table 2. Bit representations for a row in 4 bits

Step 3: Combine the right three bits of the imaginary part to the right three bits of the real part. Repeat the same for the next three bits to form q_i (see example 1).

There are four tables for the combinations of the sign of the real part and the sign of the imaginary part (+/+, +/-, -/+, -/-). Table 3 shows only the positive sign for both the imaginary and real parts.

Step 4: Label a group of 4 bits in the first row as q_0 . To generate the successive rows, multiply the current row by -4 and so on. This results in the change of sign and magnitude from one row to the next. Therefore, even rows have positive and odd rows have negative signs as shown below. ($\dots -q_5, +q_4, -q_3, +q_2, -q_1, +q_0$).

The example below illustrates the conversion of a complex binary number to the RCBNS form. The conversion of the complex number, $(2+j7)$ to the RCBNS form is as follow.

Step 1: Express numbers in the binary form for imaginary and real parts.
 Part 1 (J) $\rightarrow 7 = 0111 \rightarrow 7 = 001011$
 Part 2 (R) $\rightarrow 2 = 0010 \rightarrow 2 = 000010$

Step 2: Get 3 bits of each part and combine them together starting with LSB as 3 bits of J and put them together with 3 bits of R. $q_0 \rightarrow 011010$ & $q_1 \rightarrow 001000$

Step 3: Find the equivalent values for each group of R and J from Table 3.

$$q_0 \rightarrow 011010 \rightarrow 001101101 \rightarrow 1-1-1-1$$

$$q_1 \rightarrow 001000 \rightarrow 000000001 \rightarrow 00111$$

Step 4: Multiply the positive sign to even rows and the negative sign to the odd rows.

$$q_0 \rightarrow 1-1-1-1, \quad q_1 \rightarrow 00-1-1$$

Finally combine them again in the following order.

$\dots, q_5, q_4, q_3, q_2, q_1, q_0$ Use the term $[(-4)^i * q_i]$

for each q_i in the 4-bit row.

3.1 Reduction of Conversion tables

The four tables generated for the four possible combinations of the sign of real and imaginary parts of complex numbers can be reduced to only two tables. The -/- table can be obtained by changing the sign of the 4-bit rows of +/+ table, and vice versa. The same is valid for +/- table to -/+ table and vice versa.

$(R+J) \text{ base}_{10}$		RCBNS $(-1+j), 3$							
J	R	$2+j2$		$-j2$		$-1+j$		1	
		S	0-3	S	0-3	S	0-3	S	0-3
0 0 0	0 0 0	0	0	0	0	0	0	0	0
0 0 0	0 0 1	0	0	0	0	0	0	0	0
0 0 0	0 1 0	0	0	0	0	0	0	0	0
0 0 0	0 1 1	0	0	0	0	0	0	0	0
0 0 1	0 0 0	0	0	0	0	0	0	1	0
0 0 1	0 0 1	0	0	0	0	0	0	1	0
0 0 1	0 1 0	0	0	1	0	0	1	0	1
0 0 1	0 1 1	0	0	1	0	0	1	0	0
0 1 0	0 0 0	0	0	0	1	0	1	0	0
0 1 0	0 0 1	0	0	0	1	0	1	0	0
0 1 0	0 1 0	0	0	0	1	0	1	0	0
0 1 0	0 1 1	0	0	0	1	0	1	0	0
0 1 1	0 0 0	0	0	0	1	0	1	0	0
0 1 1	0 0 1	0	0	0	1	0	1	0	0
0 1 1	0 1 0	0	0	1	1	0	1	1	0
0 1 1	0 1 1	0	0	1	1	0	1	1	0

Table 3. Table of conversion from Binary form to RCBNS form in +/+ case

4. ARITHMETIC OPERATION

Complex numbers expressed in the RCBNS form are used as operands to perform arithmetic operations. The results of the operation will be in the RCBNS form. An example of the addition operation is shown below. In the subtraction operation the sign of real and imaginary parts

of the subtrahend are complemented and then added to the minuend. An example of the addition of two complex numbers, $(9+j11)$ and $(35+j25)$ expressed in the RCBNS form is illustrated below.

First, two complex numbers $(9+j11)$ and $(35+j25)$ are converted into the RCBNS form. Then the addition operation is performed. The complex number $(9+j11)$ is equivalent to $(10-1-1 \ 0100 \ 10-10)$ in the RCBNS form, and similarly the complex number $(35+j25)$ is equivalent to $(0000 \ 010-2 \ 0-112)$.

X	1	0	-1	-1	0	1	0	0	1	0	-1	0
Y	0	0	0	0	0	1	0	-2	0	-1	1	2
S_1	1	0	-1	-1	0	2	0	-2	1	-1	0	2

In some cases the result may have numbers ranging from -6 to 6 ; in such cases normalization is necessary. The process of normalization is as follows:

- Normalize the intermediate result to be in the set of $\{-3, -2, -1, 0, 1, 2, \text{ and } 3\}$.
- Get the final result $S = \text{Normalization} + \text{Carry}$, or $S = S_1$, if S_1 in the range of -3 to 3 .

The above result will be in the RCBNS form. The next section discusses the conversion of the RCBNS result to an equivalent binary number.

5. CONVERSION FROM RCBNS FORM TO BINARY FORM

The result of the arithmetic operation that is in the RCBNS form should be converted back to the standard binary number (SBN) form. Two methods are presented for the conversion. Both methods produce the final result in the 2's complement form.

Method 1: In this method split the real and imaginary parts of each group (one row) and forward it to two dedicated registers. The real number portion is moved into R register and the imaginary number portion is moved into J register as shown in Figure 2(a), (b).

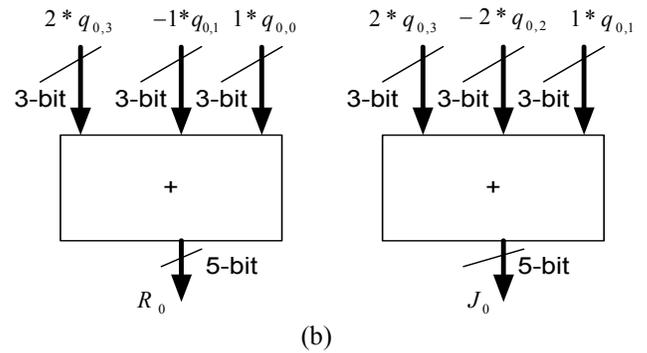
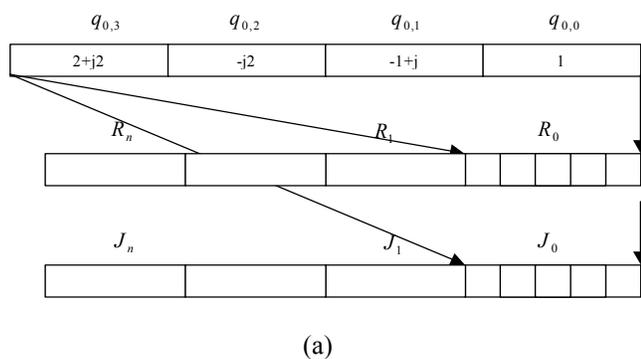


Figure 2. Conversion diagram from RCBNS to SBN

The calculation for a row is giving by the following expressions.

$$R_0 = (2 * q_{0,3}) + (0 * q_{0,2}) + (-1 * q_{0,1}) + (1 * q_{0,0})$$

$$J_0 = (2 * q_{0,3}) + (-2 * q_{0,2}) + (1 * q_{0,1}) + (0 * q_{0,0})$$

The maximum value of the Real part for one row of the RCBNS form is between -12 and 12 ($[3 * (2 + j2)] + [-3 * (-1 + j)] + (3 * 1)$), and the maximum value of the Imaginary part for one row of RCBNS form is between -15 and 15 ($[3 * (2 + j2)] + [-3 * (-j2)] + [3 * (-1 + j)]$). This method requires 5 bits for the output in the adder unit.

For example, in this method, convert the RCBNS number $(10 \bar{1} \bar{1} \ 020\bar{2} \ 1 \bar{1}02)$ ($\bar{1}$ is used for -1 and $\bar{2}$ for -2 for convenience) to SBN form. The result is equal to $row_0 (1 \bar{1}02) - 4 * row_1 (020\bar{2}) + 16 * row_2 (10 \bar{1} \bar{1}) = 44 + j36$ where $row_0 = [2 * 1 + 1 * 0 + 1 * 2] + j[2 * 1 + 2 * \bar{1} + 1 * 0]$ and so on for the rest of the rows.

Method 2: In this method, separate the real and imaginary numbers of each digit in a row to have two parts, one for real and the other for imaginary. This method modifies each row in the form shown $q_{0,3} = R_1, q_{0,2} = J_1, q_{0,1} = J_0, q_{0,0} = R_0$ and the entire real number part goes into register R' . For each row, R'_0 has 5 bits and R'_1 has 3 bits, and the imaginary number part goes into register J' . For each row, J'_0 has 3 bits and J'_1 has 5 bits. The values for the real and the imaginary parts go into the output registers R and J for each row of $R'_0, R'_1, J'_0,$ and J'_1 where:

$$R'_0 = R_0 - J_0, \quad R'_1 = R_1$$

$$J'_0 = J_0, \quad J'_1 = J_1 - R_1$$

Figure 3 shows the block diagram for implementation of Method 2 for conversion from the RCBNS form to the SBN form. An example of this conversion is shown in Figure 4.

6. CONCLUSION

A method to convert complex numbers to their equivalent RCBNS form has been presented. Combining two algorithms, the redundant binary numbers system and the binary number system for complex numbers, results in the Redundant Complex Binary Number System. Operands are received in the RCBNS form as inputs to an adder for the addition (with the sign changed to the subtrahend operation) and the sum results in the RCBNS form. Two methods have been presented for the conversion of the RCBNS result into the SBN form. The entire process will increase the speed of arithmetic operations due to a decrease in the number of steps involved. A circuit diagram implementation of the arithmetic unit based on RCBNS form using FPGA is under progress.

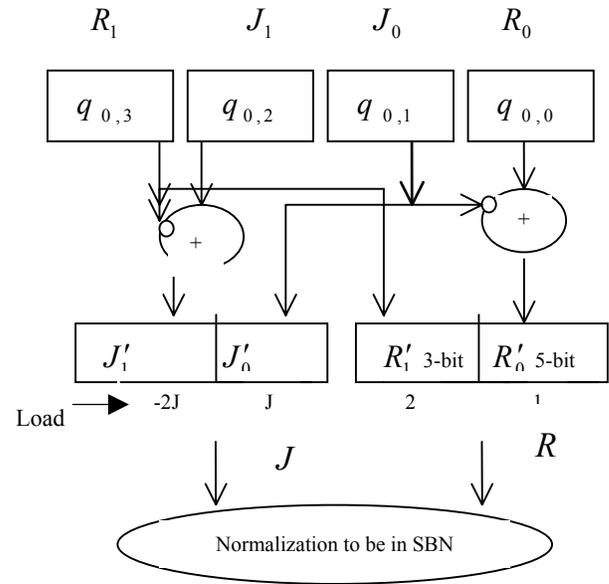
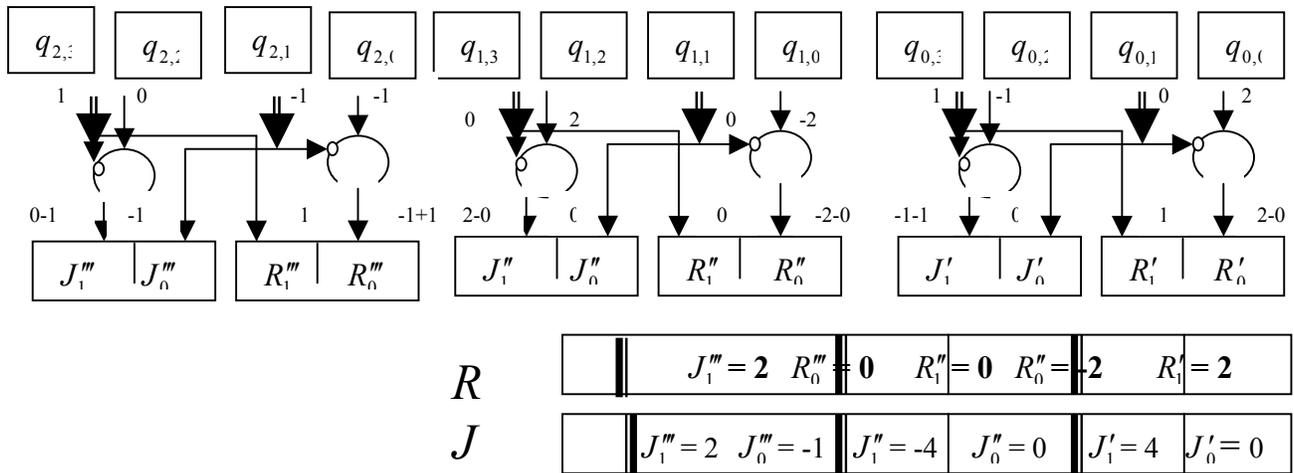


Figure 3. Conversion diagram from RCBNS to SBN



The final result is $= [16*(2)-4*(-2) + 4] + j[16*(1)-4*(-4) + (4)] = 44 + j36$

Figure 4. Conversion of the final result back to SBN

7. REFERENCES

[1] D. E. Knuth, "An Imaginary Number System", Communications of the ACM, Vol. 3, pp. 245-247, 1960.
 [2] W. Penney, "A Number System With a Negative Base," Mathematic Student Journal, Vol. 11, No. 4, pp. 1-2, May 1964.
 [3] W. Penney, "A Binary System For a Complex Numbers," Journal of the ACM, Vol. 12, No. 2, pp. 247-248, April 1965.
 [4] Y. Ohi, T. Aoki, and T. Higuchi, "Redundant Complex Number System," 25th IEEE Int. Symp. on Multiple-Valued Logic, pp. 14-19, Bloomington, IN, USA, May 1995.

[5] A. Avizienis, "Signed Digit Number Representations for Fast Parallel Arithmetic," IRE Trans. Electronic Computers, Vol. EC-10, pp. 389-400, September 1961.
 [6] T. Jamil, N. Holmes, and D. Blest, "Towards Implementation of a Binary Number System for Complex Numbers," Proceeding of the IEEE SouthEastCon, pp. 268-274, 2000.
 [7] T. Jamil, "The Complex Binary Number System," Proceeding of the IEEE potentials on 2002, pp. 39-41.
 [8] T. Aoki, K. Hoshi and T. Higuchi, "Redundant Complex Arithmetic and Its Application to Complex

Multiplier design,” 29th IEEE Int. Symp. on Multiple-Valued Logic, pp. 200-207, Freiburg, 1999.

[9] R. Mellhenny, and M. D. Ercegovic, “On-Line Algorithm for Complex Number Arithmetic,” Proceeding of the IEEE 1998, 32nd Asilomar Conference on Signals, Systems & Computers, part 1 (of 2), Pacific Grove, CA, USA, pp. 172-176, Nov 1-4 1998.

[10] T. T. Dao, “Specific Arithmetic in Radices ± 4 ,” 18th IEEE Int. Symp. on Multiple-Valued Logic, pp. 134 –139, 1988.