Journal of Advanced Computer Science and Technology, 1 (4) (2012) 176-186

www.sciencepubco.com/index.php/JACST

Study of Error Controllability for the New Modulus $\{2^{2n+1}+2^n-1, 2^{2n+1}-1, 2^n-1, 2^{3n}, 2^{3n+1}-1\}$

Samira Modiri[1], Ali Movaghar[2] and Ali Barati[3]

[1]Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran
E-mail: modiri.33@gmail.com
[2]Department of Computer Engineering, Sharif university of Technology, Tehran, Iran
E-mail: movaghar@sharif.edu
[3]Department of Computer Engineering, Dezful Branch, Islamic Azad university, Dezful, Iran
E-mail: iaud.ac.ir@iaud.ac.ir

Abstract

In this paper, a new modulus set $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1\}$ with an efficient residue-to-binary converter using mixed radix conversion algorithm is presented. Moreover, by adding two redundant modulus $\{2^{3n},\ 2^{3n+1}-1\}$, a new modulus set in redundant residue number system is provided that can correct up to $(2n+2)$ error bits. Simulation results of the error control algorithm's functionality with C++ programming language for 10'000 different error bits states show that the average percent of error detection capability using the proposed modulus set by setting n=2 is equal to 77.97%.

Keywords: *Error control, Mixed radix conversion, Redundant residue number systems, Reverse converter*

1 Introduction

Residue number system (RNS) is an unconventional, high speed, and fault tolerance number system that has many applications in today's world such as image processing systems[1], digital signal processing (DSP) [2], RSA algorithm [3], and digital communications [4]. Moreover, RNS is a useful tool for implementation of high speed FIR filters [5]. In residue number systems, instead of sending a number, using modulus in the modulus set, the remainders of the

numbers are sent and all of the operations are performed paralleled. Therefore its operations are very high speed. Moreover, the modulus in the modulus set act as secret keys, because for convert back the received remainders to the original number need to know the modulus set. Therefore, data transmission is secure in residue number systems, and if an adversary can acquire the sent message, he cannot decrypt the packet without the modulus in the modulus set. In addition to these, by adding some redundant modulus, redundant RNS (RRNS) is constructed that has error control ability. Because reliability in data delivery is a critical issue in many applications, fault tolerance is a remarkable feature of redundant residue number systems.

Two critical issues in residue number systems are modulus set selection and implementation of the reverse converter. Reverse converters using a residue to binary algorithm such as Chinese remainder theorem (CRT) [6] or mixed radix conversion (MRC) [7] can convert back the received packet to the original message. Up to now, many modulus set were presented, and efficient reverse converters were designed for them. Famous modulus set are $\{2^n-1,\ 2^n,\ 2^n+1\}$ and many different reverse converters were designed for it [8-10]. Dynamic range (DR) for this modulus set is equal to $(3n)$-bits. DR is a bound that the numbers can be represented uniquely, and is equal to the order of product all the modulus in the modulus set. For applications that need to higher dynamic range, the modulus set $\{2^n-1,\ 2^n,\ 2^n+1\}$ is not suitable. Therefore, other modulus sets with $(4n)$, $(5n)$, and $(6n)$-bits dynamic range were proposed.

In this paper, a new 3-modulus set $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1\}$ with $(5n)$-bits dynamic range is proposed and an efficient reverse converter is designed based on this modulus set. Moreover, by adding two redundant modulus $\{2^{3n},\ 2^{3n+1}-1\}$ to the proposed modulus set, the error control ability is provided. The error control power of the proposed modulus set will be scrutinized in the next sections.

The rest of the paper is organized as follows: An efficient reverse converter for the new proposed 3-main modulus set is designed in section 2. In section 3, performance of the error control capability of the proposed modulus set by adding two redundant modulus is evaluated using simulation of redundant residue number system error control algorithm's functionality for the proposed modulus set using C++ language programming by setting n=2. Finally, the paper is concluded in section 4.

2 Design and Implementation of the Proposed Reverse Converter

A residue number system is defined in terms of relatively prime modulus set $\{P_1, P_2 \ldots P_n\}$ that is $\gcd(P_i, P_j) = 1$ for $(i \neq j)$. A weighted number X can be represented as $X = (x_1, x_2 \ldots x_n)$, where

$$x_i = X \bmod P_i = |X|_{P_i}, \ 0 \leq x_i < P_i \quad (1)$$

Such a representation is unique for any integer X in the range [0, M), where $M=P_1P_2...P_n$ is the dynamic range of the modulus set $\{P_1, P_2... P_n\}$ [11]. The residue to binary conversion can be performed using the MRC as follows:

$$X = V_n \prod_{i=1}^{n} P_i + \cdots + V_3P_2P_1 + V_2P_1 + V_1 \quad (2)$$

The coefficients V_iP can be obtained from residues by

$$V_1 = x_1 \quad (3)$$

$$V_2 = |(x_2 - x_1)\,|P_1^{-1}|_{P_2}|_{P_2} \quad (4)$$

$$V_3 = |((x_3 - x_1)|P_1^{-1}|_{P_3} - V_2)\,|P_2^{-1}|_{P_3}|_{P_3} \quad (5)$$

In the general case, we have

$$V_n = (((x_n - V_1)\,|P_1^{-1}|_{P_n} - V_2)\,|P_2^{-1}|_{P_n} - \ \ldots - V_{n-1})\,|P_{n-1}^{-1}|P_n|_{P_n} \quad (6)$$

where $|P_i^{-1}|_{P_j}$ is multiplicative inverse of P_i modulo P_j. The modular multiplicative inverse of a modulo "m" can be found with the extended Euclidean algorithm. According to these equations, the proposed reverse converter for the new 3-modulus set $\{2^{2n+1}+2^n-1, 2^{2n+1}-1, 2^n-1\}$ can design as follows:

Consider the three-modulus set $\{P_1, P_2, P_3\} = \{2^{2n+1}+2^n-1, 2^{2n+1}-1, 2^n-1\}$ with three corresponding residues (x_1, x_2, x_3). For design of a residue to binary converter, firstly need to prove that the modulus of proposed modulus set $\{2^{2n+1}+2^n-1, 2^{2n+1}-1, 2^n-1\}$ are in fact pair wise relatively prime for the validity of the RNS. Next, should to find the multiplicative inverses, and then the values of the multiplicative inverses and modulus set must substitute in the conversion algorithm formulas. Then, the resulted equations should be simplified by using arithmetic properties. Finally, simplified equations would realize using hardware components such as full adders and logic gates. Based on Euclid's Theorem:

$$\gcd(a, b) = \gcd(b, a \bmod b), a > b \quad (7)$$

Hence,

$$\gcd(2^{2n+1} + 2^n - 1, 2^{2n+1} - 1) = \gcd(2^{2n+1} - 1, 2^n) = \gcd(2^n, -1) = 1 \quad (8)$$

$$\gcd(2^{2n+1}+2^n-1, 2^n-1) = \gcd(2^n-1, 2) = 1 \quad (9)$$

$$\gcd(2^{2n+1}-1, 2^n-1) = \gcd(2^n-1, 1) = 1 \quad (10)$$

Since the greatest common divisors are one, thus the numbers $2^{2n+1}+2^n-1$, $2^{2n+1}-1$, 2^n-1 are relatively prime together. In what follows, by use of three propositions, the closed form expressions for the multiplicative inverses under the mixed radix conversion algorithm are derived that form the basis of our algorithm for the reverse converter.

Proposition 2.1 The multiplicative inverse of $(2^{2n+1}+2^n-1)$ modulo $(2^{2n+1}-1)$ is equal to $k_1= 2^{n+1}$.

Proof:

$$|2^{n+1} \times (2^{2n+1}+2^n-1)|_{2^{2n+1}-1} = 1 \quad (11)$$

Proposition 2.2 The multiplicative inverse of $(2^{2n+1}+2^n-1)$ modulo (2^n-1) is equal to $k_2 = 2^{n-1}$.

Proof:

$$|2^{n-1} \times (2^{2n+1}+2^n-1)|_{2^n-1} = 1 \quad (12)$$

Proposition 2.3 The multiplicative inverse of $(2^{2n+1}-1)$ modulo (2^n-1) is equal to $k_3= 1$.

Proof:

$$|(2^{2n+1}-1)|_{2^n-1} = 1 \quad (13)$$

Therefore, let the values $k_1= 2^{n+1}$, $k_2 = 2^{n-1}$, $k_3=1$, $P_1= 2^{2n+1}+2^n-1$, $P_2 = 2^{2n+1}-1$, $P_3= 2^n-1$ in (2-5) and we have:

$$X = x_1 + P_1 (V_2 + V_3 P_2) = x_1 + (2^{2n+1}+2^n-1)(V_2 + (2^{2n+1}-1)V_3) \quad (14)$$

$$V_1 = x_1 \quad (15)$$

$$V_2 = |(x_2 - x_1)\,|P_1^{-1}|_{P_2}|_{P_2} = |2^{n+1} \times (x_1 - x_2)|_{2^{2n+1}-1} \quad (16)$$

$$\begin{aligned} V_3 &= |((x_3 - x_1)|P_1^{-1}|_{P_3} - V_2)\,|P_2^{-1}|_{P_3}|_{P_3} \\ &= |(2^{n-1} \times (x_3 - x_1) - V_2)\,|_{2^n-1} \end{aligned} \quad (17)$$

According to the following two properties, (14-17) can be simplified to decrease the hardware complexity.

Property 2.1 The residue of a negative residue number $(-\nu)$ in modulo (2^n-1) is the one's complement of ν, where $0 \le \nu < (2^n-1)$ [12].

Property 2.2 The multiplication of a residue number ν by 2^P in modulo $(2^n\text{-}1)$ is carried out by P bit circular left shift, where P is a natural number [12].

For designing an efficient reverse converter, simplify (14, 16, 17) as follow:

$$V_2 = |(x_2 - x_1)|P_1^{-1}|_{P_2}|_{P_2} = |2^{n+1} \times (x_1 - x_2)|_{2^{2n+1}-1} \\ = |2^{n+1} \times x_2|_{2^{2n+1}-1} + |-2^{n+1} \times x_1|_{2^{2n+1}-1} = V_{21} + V_{22} \tag{18}$$

Where,

$$V_{21} = |2^{n+1} \times x_2|_{2^{2n+1}-1} = \underbrace{x_{2,n-1} \dots x_{2,0}}_{n\ bits}\ \underbrace{x_{2,2n} \dots x_{2,n}}_{(n+1)bits} \tag{19}$$

$$V_{22} = |-2^{n+1} \times x_1|_{2^{2n+1}-1} = \left\{ \begin{matrix} \overbrace{\bar{x}_{1,n-1} \dots \bar{x}_{1,1}}^{(n-1)bits}\ \bar{x}_{1,0}\ \overbrace{\bar{x}_{1,2n} \dots \bar{x}_{1,n}}^{(n+1)bits} \\ \underbrace{1 \dots 1 \dots 1}_{(n-1)bits}\ \bar{x}_{1,2n+1}\ \underbrace{1 \dots 1 \dots 1}_{(n+1)bits} \end{matrix} \right\} \tag{20}$$

For realize V_3 based on (17):

$$V_3 = |((x_3 - x_1)|P_1^{-1}|_{P_3} - V_2)\ |P_2^{-1}|_{P_3}|_{P_3} \\ = |(2^{n-1} \times (x_3 - x_1) - V_2)\ |_{2^n-1} \\ = |2^{n-1} \times x_3|_{2^n-1} + |-2^{n-1} \times x_1|_{2^n-1} + |-V_2|_{2^n-1} \\ = V_{31} + V_{32} + V_{33} \tag{21}$$

Where,

$$V_{31} = |2^{n-1} \times x_3|_{2^n-1} = x_{3,0}\ \underbrace{x_{3,n-1} \dots x_{3,1}}_{(n-1)bits} \tag{22}$$

$$V_{32} = |-2^{n-1} \times x_1|_{2^n-1} = \left\{ \begin{matrix} \overbrace{\bar{x}_{1,0}}^{1\ bit}\ \overbrace{\bar{x}_{1,n-1} \dots \bar{x}_{1,2}}^{(n-2)bits}\ \overbrace{\bar{x}_{1,1}}^{1\ bit} + \\ \bar{x}_{1,n}\ \bar{x}_{1,2n-1} \dots \bar{x}_{1,n+2}\ \bar{x}_{1,n+1} + \\ \underbrace{\bar{x}_{1,2n}}_{1\ bit}\ \underbrace{1 \dots .1 \dots 1 \dots 1}_{(n-2)bits}\ \underbrace{\bar{x}_{1,2n+1}}_{1\ bit} \end{matrix} \right\} \tag{23}$$

$$V_{33} = |-V_2|_{2^n-1} = \left\{ \begin{matrix} \overbrace{\bar{V}_{2,n-1} \dots \bar{V}_{2,1}}^{(n-1)bits}\ \bar{V}_{2,0} + \\ \bar{V}_{2,2n-1} \dots \bar{V}_{2,n+1}\ \bar{V}_{2,n} + \\ \underbrace{1 \dots 1 \dots 1}_{(n-1)bits}\ \bar{V}_{2,2n} \end{matrix} \right\} \tag{24}$$

Finally, for find X based on (14), we have:

$$\begin{aligned} X &= x_1 + P_1 (V_2 + V_3 P_2) \\ &= x_1 + (2^{2n+1} + 2^n - 1) \underbrace{(V_2 + (2^{2n+1} - 1)V_3)}_{C} \\ &= x_1 + (2^{2n+1} + 2^n - 1)C \end{aligned} \tag{25}$$

$$C = V_2 + (2^{2n+1} - 1)V_3 \tag{26}$$

$$C = \left\{ \begin{array}{l} \overbrace{V_{3,n-1} \dots V_{3,0}}^{n\ bits}\ \overbrace{V_{2,2n}}^{1\ bit}\ \overbrace{V_{2,2n-1} \dots V_{2,n}}^{n\ bits}\ \overbrace{\bar{V}_{3,n-1} \dots \bar{V}_{3,0}}^{n\ bits} + \\ \underbrace{1 \dots 1 \dots 1 \dots 1 \dots 1 \dots 1 \dots 1}_{(2n+1)bits}\ \underbrace{V_{2,n-1} \dots V_{2,0}}_{n\ bits} \end{array} \right\} \tag{27}$$

$$\begin{aligned} X &= x_1 + \mathrm{P}_1 (\mathrm{V}_2 + \mathrm{V}_3 P_2) = x_1 + \left(2^{2n+1} + 2^n - 1\right) C \\ &= \left\{ \begin{array}{l} \overbrace{C_{3n} \dots C_{2n}}^{(n+1)bits}\ \overbrace{C_{2n-1} \dots C_n}^{n\ bits}\ \overbrace{C_{n-1} \dots C_0}^{n\ bits}\ \overbrace{\bar{C}_n}^{1\ bit}\ \overbrace{C_{n-1} \dots C_0}^{n\ bits}\ \overbrace{\bar{C}_{n-1} \dots \bar{C}_0}^{n\ bits} + \\ 1 \dots 1 \dots 1\ C_{3n} \dots C_{2n+1}\ C_{2n} \dots C_{n+1}\ \bar{C}_{2n}\ \bar{C}_{2n-1} \dots \bar{C}_n\ x_{1,n-1} \dots x_{1,0} + \\ \underbrace{0 \dots 0 \dots 0 \dots 0}_{(n+1)bits}\ \underbrace{1 \dots 1 \dots 1}_{n\ bits}\ \underbrace{\bar{C}_{3n} \dots \bar{C}_{2n+1}}_{n\ bits}\ \underbrace{x_{1,2n}}_{1\ bit}\ \underbrace{x_{1,2n-1} \dots x_{1,n}}_{n\ bits}\ \underbrace{0 \dots 0}_{n\ bits} \end{array} \right\} \end{aligned} \tag{28}$$

Using FAs, carry save adders (CSAs) with end around carry (EAC) and carry propagation adder (CPA) with EAC, the proposed reverse converter is implemented. Implementation of reverse converter for the 3-modulus set $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1\}$ is based on (18, 21, 26, 28). Firstly, the operand preparation unit (1) (OPU (1)) prepares the required operands (19, 20, 22, 23) and these preparation rely on simply manipulating the routing of the bits of the residues. Also, we need (2n+2) NOT gates for performing the inversions in (20, 23). Implementation of (18, 21) requires one modulo $(2^{2n+1}-1)$ adder and one modulo (2^n-1) adder, respectively. These modulus adders can be implemented with different methods [13]. In this paper, Carry propagate adder (CPA) with end-around carry (EAC) is considered. The delay of a CPA with EAC is twice the delay of a regular CPA, while it has the same hardware complexity [14]. Realization of (19, 20) relies on a (2n+1)-bit carry save adder and a (2n+1)-bit carry propagate adder with end around carry (CSA (1) and CPA (1) with EAC). (n- bit)- CSA (2) and CSA (3) with EAC are used to implementation of equations (22, 23). It is also observed that there are strings of consecutive "1"s embedded in the binary expressions of (23, 24, 27, 28). These constant inputs "1" can be reduced to a pair of two-input XNOR and OR gates. In (28), some inputs are "0", and can substitute FAs with XOR and AND gates. OPU (2), OPU (3), and OPU (4) prepare the required operands for equation (24), equation (27), and equation (28), respectively. Realization of (24) relies on 3-operand (n-bit) CSA (4, 5, 6) with

EAC and a modulo (2^n-1) adder for implementation V_3. Regular CPA (1) is used to implementation of (27). Finally, implementation of (28) requires a (5n+2)-bit CSA (7) with EAC and a (5n+2)-bit regular CPA (2). Area and delay specifications of each part of the converter are shown in Table 1.

Table 1: Characterization of each part of the proposed converter for the new three modulus set $\{2^{2n+1}+2^n-1, 2^{2n+1}-1, 2^n-1\}$

Parts	FA	not	And/ Xor	Or/Xnor	Delay
OPU (1)	-	(2n+2)	-	-	t_{Not}
CSA (1)	1	-	-	(2n)	t_{FA}
CPA (1)	(2n+1)	-	-	-	(4n+2) t_{FA}
CSA (2)	n	-	-	-	t_{FA}
CSA (3)	2	-	-	(n-2)	t_{FA}
OPU (2)	-	(2n+1)	-	-	t_{Not}
CSA (4)	n	-	-	-	t_{FA}
CSA (5)	n	-	-	-	t_{FA}
CSA (6)	1	-	-	(n-1)	t_{FA}
CPA (2)	n	-	-	-	(2n) t_{FA}
OPU (3)	-	n	-	-	t_{Not}
R- CPA (1)	n	-	-	(2n+1)	(3n+1) t_{FA}
OPU (4)	-	(3n+1)	-	-	t_{Not}
CSA (7)	(2n+1)	-	(2n+1)	n	t_{FA}
R- CPA (2)	(5n+2)	-	-	-	(5n+2) t_{FA}

$$\text{Total Area} = (14n+8)\,A_{FA} + (8n+4)\,A_{NOT} + (2n+1)\,A_{AND} + (2n+1)\,A_{XOR} + (7n-2)\,A_{OR} + (7n-2)\,A_{XNOR} \quad (29)$$

$$\text{Total delay} = (14n+12)\,t_{FA} + 4\,t_{NOT} \quad (30)$$

Now, performance of the proposed reverse converter for new modulus set $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1\}$ is compared with the reverse converters with the same dynimc range or less. This comparison is shown in Table 2.

Table 2: Area and Delay comparison between the proposed reverse converter and related works

	Modulus set	DR	Area (A_{FA})	Delay (t_{FA})
[15]	$\{2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1\}$	5n	16n - 1	18n + 7
[16]	$\{2^n, 2^n1, 2^n+1, 2^{n+1}+1\}$	4n+1	n^2 + 12n + 12	16n + 22
[17] - CICE	$\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$	4n	25.5n + 12	18n + 23
[17] - C3CE	$\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$	4n	23n + 11	16n + 14
Proposed	$\{2^{2n+1}+2^n-1, 2^{2n+1}-1, 2^n-1\}$	5n+2	14n + 8	14n + 12

3 Performance Evaluation

In this section, the performance of the proposed modulus set $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1,\ 2^{3n},\ 2^{3n+1}-1\}$ is evaluated in terms of error detection and error correction capability. Firstly we must to explain that why these two redundant modulus were selected for these main modulus. Note that the redundant modulus must be greater than the main modulus, and all of the modulus including the main and redundant modulus must to prime together. In the proposed modulus set, $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1,\ 2^{3n},\ 2^{3n+1}-1\}$, it is obvious that the redundant modulus are greater than the main modulus. For indicate that the modulus in the proposed modulus set are pair wise prime, only need to show each redundant modulo is prime respect to each main modulo, and the redundant modulus are prime together. Because before, in the previous section, we demonstrate that the main modulus are prime together. Based on Euclid's theorem (equation (7)), we have:

$$\gcd(2^{3n}, 2^{2n+1} + 2^n - 1) = 1 \tag{31}$$

$$\gcd(2^{3n}, 2^{2n+1} - 1) = \gcd(2^{2n+1} - 1, 2^{n-1}) = \gcd(2^{n-1}, -1) = 1 \tag{32}$$

$$\gcd(2^{3n}, 2^n - 1) = \gcd(2^n - 1, 1) = 1 \tag{33}$$

$$\gcd(2^{3n+1} - 1, 2^{2n+1} + 2^n - 1) = 1 \tag{34}$$

$$\gcd(2^{3n+1} - 1, 2^{2n+1} - 1) = \gcd(2^{2n+1} - 1, 2^n - 1) = \gcd(2^n - 1, 1) = 1 \tag{35}$$

$$\gcd(2^{3n+1} - 1, 2^n - 1) = \gcd(2^n - 1, 1) = 1 \tag{36}$$

$$\gcd(2^{3n+1} - 1, 2^{3n}) = \gcd(2^{3n}, -1) = 1 \tag{37}$$

It is obvious from (31-37) that the redundant modulus are pair wise prime together and are prime against the main modulus, thus these redundant modulus are suitable for these three main modulus.

Now, error control capability of the proposed modulus set, $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1,\ 2^{3n},\ 2^{3n+1}-1\}$ is evaluated. Error control includes ”error detection” and ”error detection & correction”. Error(s) bit in the received packets will be detect, locate and correct, if the error(s) occurs in not more than one modulo. In the other word, the error control algorithm can correct error(s) if the error bit(s) located at the one received remainder in the packet, because redundant residue number system can correct "burst errors". If the errors occur in more than one modulo, the algorithm can not correct the error, but with a high probability, can detect error. The hidden errors that don't recognize by the algorithm will be cause to false response, and a wrong data will be acquired.

In the proposed modulus set, $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1,\ 2^{3n},\ 2^{3n+1}-1\}$, the main three modulus are $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1\}$ and two modulus $\{2^{3n},\ 2^{3n+1}-1\}$ are the redundant modulus. By using the redundant modulus, the RRNS error control algorithm can control the errors. In the three main modulus, the first modulo $(2^{2n+1}+2^n-1)$, the second modulo $(2^{2n+1}-1)$, and the third modulo (2^n-1) has $(2n+2)$, $(2n+1)$ and (n) bits respectively. Thus, the error bits can be detect and correct if they sat on the first, second, or in the third modulo, up to $(2n+2)$, $(2n+1)$, and n bits, respectively. Therefore, the maximum power of error detection and correction algorithm is correction of $(2n+2)$ error bits that occur in the first modulo. It is obvious that the minimum power of correction using the error control algorithm is equal to correction of one bit error that can occurs in each bit of the received packet.

For study the error detection capability using the proposed modulus set in the situations that errors occur in more than one modulo, and thus we ensure that error correction is impossible, we use simulation of error control algorithm's functionality using C++ programming language and test 10'000 different error bit states by setting $n=2$ in the proposed modulus set. The proposed modulus set $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1,\ 2^{3n},\ 2^{3n+1}-1\}$ in a special case with setting $n=2$ is equal to $\{35,\ 31,\ 3,\ 64,\ 127\}$ that $\{64,\ 127\}$ are the redundant modulus. The results show that the average percent of error detection capability using the modulus set $\{35, 31, 3, 64, 127\}$ is equal to 77.97%.

4 Conclusion

In this paper, a new 3-modulus set $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1\}$ presents and an efficient reverse converter based on mixed radix conversion algorithm designs for it. Comparison with the other reverse converters shows the proposed reverse converter is preferable in terms of speed of operations and hardware requirements. Moreover, two redundant modulus $\{2^{3n},\ 2^{3n+1}-1\}$ add to the proposed modulus set and a new modulus $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1,\ 2^{3n},\ 2^{3n+1}-1\}$ constructs for redundant residue number systems that has error control capability. If error occurs in the received message, using the redundant modulus, error control algorithm can detect or correct the error bit(s) and enhancement in reliability of data delivery can be provided. The modulus set $\{2^{2n+1}+2^n-1,\ 2^{2n+1}-1,\ 2^n-1,\ 2^{3n},\ 2^{3n+1}-1\}$ can correct error bits up to $(2n+2)$ bits. For evaluation the error detection ability of the proposed modulus set in the situation that error correction is not possible, simulation of error control algorithm's functionality is used that is based on C++ programming language. The results based on study of different 10'000 error bits states show average percent of error detection capability using the proposed modulus set by setting $n=2$ is equal to 77.97%.

References

[1] W. Wei, M.N.S. Swamy, M.O. Ahmad, ”RNS application for digital image processing”, *Proceedings of the 4th IEEE international workshop on system-on-chip for real time applications, Canada*, (2004), pp. 77-80.

[2] F. Taylor, ”A single modulus ALU for signal processing”, *IEEE Transactions on Acoustics, Signal Processing*, Vol. 33, (1985), pp. 1302-1315.

[3] S. Yen, S. Kim, S. Lim, S. Moon, ”RSA speedup with Chinese remainder theorem immune against hardware fault cryptanalysis”, *IEEE Transactions on Computers*, Vol. 52, No. 4, (2003), pp. 461-472.

[4] E. Kinoshita, K. Lee, ”A residue arithmetic extension for reliable scientific computation”, *IEEE Transactions on Computers*, Vol. 46, No. 2, (1997), pp. 129-138.

[5] R. Convey, J. Nelson, ”Improved RNS FIR filter architectures”, *IEEE Transactions on Circuits and Systems-II*, Vol. 51, No. 1, (2004), pp. 26-28.

[6] K.M. Elleithy, M.A. Bayoumi, Fast and flexible architectures for RNS arithmetic decoding”, *IEEE Transactions on Circuits and Systems-II*, Vol. 39, No. 4, (1992), pp. 226-235.

[7] C.H. Huang, ”A fully parallel mixed radix conversion algorithm for residue number applications”, *IEEE Transactions on Computer*, Vol. 32, No. 4, (1983), pp. 398-402.

[8] P.V.A. Mohan, ”Evaluation of fast conversion technique for binary-residue number systems”, *IEEE Transactions on Circuit and Systems-I*, Vol. 45, No. 10, (1998), pp. 1107-1109.

[9] P.V.A. Mohan, ”The digit parallel method for fast RNS to weighted number system conversion for specific moduli (2^n-1, 2^n, 2^n+1)”, *IEEE Transactions on Circuits and Systems-II*, Vol. 47, No. 9, (2000), pp. 972-974.

[10] P.V.A. Mohan, ”Breaking the 2n-bit carry propagation barrier in residue to binary conversion for the (2^n-1, 2^n, 2^n+1) moduli set”, *IEEE Transactions on Circuits and Systems-II*, Vol. 48, No. 8, (2001), pp. 1031-1035.

[11] F.J. Taylor, ”Residue arithmetic: A tutorial with examples”, *IEEE Computer Magazine*, Vol. 17, No. 5, (1984), pp. 50-62.

[12] S.J. Piestrak, ”A high speed realization of a residue to binary converter”, *IEEE Transactions on Circuits and Systems. II, Analogue and digital Signal Processing*, Vol. 42, No. 10, (1995), pp. 661-663.

[13] A. Hariri, K. Navi, R. Rastegar, ”A new high dynamic range moduli set with efficient reverse converter”, *Elsevier/ Journal of Computation and Mathematics with Applications*, Vol. 55, No. 4, (2008), pp. 660-668.

[14] A.S. Molahossenin, K. Navi, C. Dadkhah, S. Timarchi, ”Efficient reverse converter designs for the new 4-moduli sets {2^n-1, 2^n, 2^n+1, 2^{2n+1}-1} and {2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1} based on new CRTs”, *IEEE Transactions on Circuits and Systems-I: Regular papers*, Vol. 57, No. 4, (2010), pp. 1-13.

[15] B. Cao, C.H. Chang, T.H. Srikanthan, ”A residue to binary converter for a new five moduli set”, *IEEE Transactions on Circuits and Systems – I: Regular papers*, Vol. 54, No.5, (2007), pp. 1041-1049.

[16] P.V.A. Mohan, A.B. Premkumar, ”RNS to binary converters for two four moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ and $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$”, *IEEE Transactions on Circuits and Systems- I: Regular papers*, Vol. 24, No.6, (2007), pp. 1245-1254.

[17] P.V.A. Mohan, ”New reverse converters for the moduli set $\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$”, *Elsevier/ International Journal of Electronics and Communications (AEU)*, Vol. 62, No. 9, (2008), pp. 643- 658.