Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 7 (23): 3729-3735, 2007
ISSN 1812-5654

A New Moduli Set for Residue Number System in Ternary Valued Logic

[1]M. Hosseinzadeh and [2]K. Navi
[1]Islamic Azad University, Science and Research Branch,Tehran, Iran
[2]Department of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran

Abstract: During the past few years, the Residue Number System (RNS) has been receiving considerable interest due to its parallel and fault-tolerant properties. This system is a useful tool for Digital Signal Processing (DSP) since it can support parallel, carry-free, high-speed and low power arithmetic. One of the most important considerations when designing RNS systems is the choice of the moduli set. This is due to the fact that the systems speed, its dynamic range, as well as its hardware complexity depend on both the forms and the number of the chosen moduli. In this study a new moduli set $\{3^n-2, 3^n-1, 3^n\}$ is introduced. This moduli set includes pair wise relatively prime moduli, so it offers the maximum possible dynamic range. Ternary Valued Logic (TVL) is a potential and actual alternative to conventional Binary Logic (BL) in order to implement logical operators and arithmetic circuits. Using TVL the chip area and overall delay can be reduced. For this moduli set, the related circuits are simply realized in the Ternary Valued Logic and arithmetic in this moduli set enjoys very high speed operations and simple reverse/forward conversion (RNS to TVL/TVL to RNS). Comparisons demonstrate that we have achieved a significant improvement in terms of speed and time complexity.

Key words: Computer arithmetic, high-speed arithmetic, residue number system, Ternary Valued Logic (TVL), VLSI

INTRODUCTION

The Residue Number System is a non weighted number system, which speeds up arithmetic operations by dividing them into smaller parallel operations. A Residue Number System is characterized by a moduli set $\{m_1, m_2, ..., m_L\}$, where the modulo, m_i,(i = 1, 2, ...L), are pair wise relatively prime (Garner, 1959; Parhami, 2001). Any integer X in the dynamic range, $M= m_1' m_2' ...'m_L$, is represented by an L-tuple $(x_1, x_2, x_3, ..., x_{L-1}, x_L)$, where, x_i is the residue of X in modulo m_i for i = 1, 2, ..., L. An integer X is represented by an L-tuple where, x_i is a nonnegative integer satisfying $X = m_i q_i + x_i$ and $0 \le x_i < m_i$.

By converting the arithmetic of large numbers to a set of the parallel arithmetic of smaller numbers, the RNS representation yields significant speed up in some cases. As we can carry out operations such as addition, subtraction and multiplication very rapidly in RNS representation, if the percentage of these operations is high enough then it is convenient to use it. Since in the Residue Number System the arithmetic operation in each modulo is independent of the others, there is no carry propagation among them. Some arithmetic applications of the RNS are real time processing, digital filters (Conway and Nelson, 2004), digital signal processing (Freking and Parhi, 1997; Soderstand *et al*., 1986), image processing (Szabo and Tanaka, 1967; Taylor, 1985), the RSA encoding algorithm (Yen *et al*., 2003) and digital communication (Ramirez *et al*., 2002). The architecture of the RNS is naturally fault tolerant and consequently, it is used to detect and correct the fault (Kinoshita and Lee, 1997).

Regarding to the application of RNS, we need large dynamic range for more accuracy and security. Two important issues in order to achieve a high dynamic range in RNS, are the selection of moduli set and the number of items.

Beside the selection and the number of moduli set must satisfy the appropriate speed and hardware complexity. Some notes to be notified in choosing the optimum modulo are:

- The moduli must be pair wise relatively prime to gain the maximum dynamic range.
- Computational and conversion circuits must be simple.
- The moduli circuits have to be chosen to have nearly the same delay.

In addition in order to obtain a high dynamic range there are two general methods cited in literature:

- Using many small moduli.
- Using a few large moduli.

Heach selection has it drawbacks.

Corresponding Author: K. Navi, Department of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran

The problem with the first method: As conversion from RNS to weighted number system is done based on Chinese Remainder Theorem (CRT) and its related delay is proportional to the number of moduli, the more the number of moduli the more the conversion delay will be. Therefore choosing many pair wise relatively prime moduli is difficult task and beside the power of this modulo will be too large and will be resulting in different delays.

The problem concerning second method: The delay of internal computation of RNS is proportional to the modulo dimension. The more the modulo dimension the more delay of internal computation will be causing a reduction in speed of internal computation RNS and hence a reducing of overall speed.

Principally, some moduli sets such as $\{2^n+1,2^n,2^n-1\}$, $\{r^n-2,r^n-1,r^n\}$ $\{2^n-2^n-1,2^{n-1}-1\}$, $\{2^n-3,2^n+1,2^n-1,2^n+3\}$ (Paliouras and Stouraitis, 2000; Cao *et al*., 2007; Wang *et al*., 2002; Bhardwaj *et al*., 1998; Gallaher *et al*., 1997; Wang *et al*., 2000; Hiasat and Zohdy, 1998; Sheu *et al*., 2004; Hosseinzadeh *et al*., 2007), or $\{r^a, r^b-1, r^c+1\}$ (Abdallah and Skavantzos, 2005) are used. In this study a new moduli set $\{3^n-2,3^n-1,3^n\}$ is presented.

TERNARY VALUED LOGIC

Despite binary logic in which logical levels are restricted to two possible states, namely false and true, there exists an alternative named Multiple Valued Logic (Hurst, 1984). In this system, theoretically, one can define an unlimited number of logical levels, but in reality it is limited and this limitation mainly depends on the used technology. In Ternary valued logic with 3 levels comprising {0,1,2}, we can introduce a new era.

It is obvious that the positional weights of any two succeeding columns are power of 3. Figure 1 shows the positional value of each location.

Each location in an TVL component can store much more information than a binary logic component can, the dynamic range of the moduli set $\{3^n-2,3^n-1,3^n\}$ is much greater than its equivalent in the binary representation. Now the question is how to present the related hardware which is clearly answered in (Gonzalez and Mozumdar, 2000). For example if we compare two different moduli sets with equal number of locations, the results shown in Table 1 will be obtained.

Based on these comparisons, we conclude that the TVL with the same number of locations has a much larger dynamic range.

MODULI SET $\{3^n-2,3^n-1,3^n\}$ AND CIRCUITS

Moduli set $\{3^n-2,3^n-1,3^n\}$: Here, we first show that the moduli set $\{3^n-2,3^n-1,3^n\}$ includes pair wise relatively prime moduli. It is well known that two consecutive integer numbers are relatively prime. So it is clear that 3^n and 3^n-1 are relatively prime and the same is true for 3^n-2 and 3^n-1. We show that 3^n and 3^n-2 are relatively prime.

Fig. 1: Positional weight value of each location

Table 1: Comparison of dynamic range and number of location

Moduli set	No. of position	M
$\{3^n-2, 3^n-1, 3^n\}$	3n	$3^{3n}-3^{2n+1}+2\times 3^n$
$\{2^n-1,2^n, 2^{n-1}, 1\}$	3n	$2^{3n-1}-1.5(2^{2n})+2^n$
$\{2^n-1,2^n, 2^n+1\}$	3n+1	$2^{3n}-2^n$

These moduli set has the largest possible dynamic range, cause regarding the theorem 1, its modulo are pair wise relatively prime.

Theorem 1: all the moduli set are pair wise relatively prime.

Proof: using Euclid's theorem:

- if a, b are two integer number the :

$$GCD(a, b) = GCD (a, b \bmod a)$$

- if a is an even number the: $\langle 3^n-\alpha\rangle = 1$

So:

$$GCD(3^n-2,3^n-1) = GCD(3^n-2,1) = 1,$$
$$GCD(3^n-2,3^n-1) = GCD(3^n-2,2) = 1,$$
$$GCD(3^n-1,3^n) = GCD(3^n-1,1) = 1,$$

Regarding the above equations these three moduli are pair wise relatively prime.

The main properties of the proposed moduli set can be concluded as follows:

- Pair wise relatively prime moduli.
- Simple realization of related circuit.
- Maximum memory usage.
- Larger dynamic range.
- N positional circuits for each modulo.

RNS adder $\{3^n-2,3^n-1,3^n\}$: In the RNS, two-operand arithmetic operations are defined as below:

$$\{z_1, z_2, z_3, \ldots, z_{L-1}, z_L\} = \{x_1, x_2, x_3, \ldots, x_{L-1}, x_L\} \circ \{y_1, y_2, y_3, \ldots, y_{L-1}, y_L\}$$

where, i = {1, 2, ..., L}, $z_i = x_i$ o $y_i \bmod m_i$ and o can be any of addition, subtraction, or multiplication operations (Hariri *et al*., 2005; Timarchi *et al*., 2006; Hosseinzadeh *et al*., 2006a). In applications where addition subtraction and multiplication are frequently used, RNS is a very good candidate to realize these operations. Between these three arithmetic operations the most important is addition. The two other operations could be realized based on addition. So we are going to describe the new algorithms and circuits for addition and conversion.

If we consider two numbers A, B as the residues in respect of modulo m, then $0 \le A \le m-1$, $0 \le B \le m-1$. For doing the addition operation in the modulo m, as the results lie between 0 and 2 m-2, all we need to do is as below:

$$\begin{cases} A + B < m & \Rightarrow \quad A + B \\ A + B \ge m & \Rightarrow \quad A + B - m \end{cases} \tag{3}$$

In other words, if the result is greater than or equal to the moduli, we add it to the complement of the moduli and ignore the carry out. For performing the addition operation in the modulo 3^n, we add up two numbers and as the carry out is a multiple of 3^n, we simply ignore the carry out. The corresponding circuit is illustrated in Fig. 2.

In modulo 3^n-1 if the result is greater than or equal to 3^n-1 then the result will be added to the complement of modulo 3^n-(3^n-1) = 1. For speeding up using parallelism, the result and the same result plus one are generated simultaneously and by using a multiplexer, the correct value will be directed to the output. The corresponding circuit is as below (Fig. 3).

In the modulo 3^n-2, if the result of addition is greater than or equal to 3^n-2 then it will be added to the complement of the modulo which is 3^n-(3^n-2) =2. An interesting property of TVL is the possibility of having a carry in generated equal to two (in an adder in the base 3, carry in can be between zero and 2). Using the proposed method for 3^n-2, the adder can be realized as presented in Fig. 4 (Hosseinzadeh *et al*., 2006b).

Converterts

Ternary valued logic approach to RNS converters: Suppose that the number A has 2n digits in its TVL representation. We would like to compute its residue in the modulo 3^n. So we can write:

$$\begin{aligned} &(\overbrace{a_{2n-1} \ldots a_{n+1}a_n}^{A_2}\,\overbrace{a_{n-1} \ldots a_1a_0}^{A_1}) \bmod 3^n = \\ &\left[(a_{2n-1} \ldots a_{n+1}a_n) \times 3^n + (a_{n-1} \ldots a_1a_0)\right] \bmod 3^n \\ &= a_{n-1} \ldots a_1a_0 \end{aligned}$$

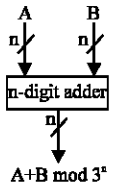

Fig. 2: Addition circuit for RNS with 3^n modulo

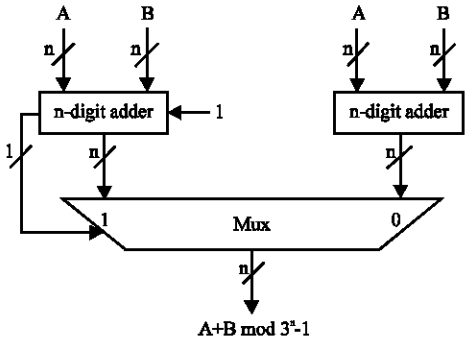

Fig. 3: The RNS addition in 3^n modulo

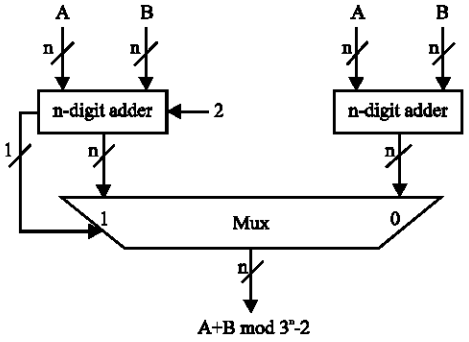

Fig. 4: The RNS addition in 3^n-2 modulo

Therefore, it is enough to consider the n right most digits and the residue of the rest of the digits will be ignored as they are multiples of 3^n.

Now we investigate the modulo 3^n-1. The residue of a 2n digit TVL number in the modulo 3^n-1 is calculated as below:

$$\begin{aligned} &(a_{2n-1} \ldots a_{n+1}a_na_{n-1} \ldots a_1a_0) \bmod (3^n - 1) = \\ &\begin{bmatrix} (a_{2n-1} \ldots a_{n+1}a_n) \times 3^n \\ +(a_{n-1} \ldots a_1, a_0) \end{bmatrix} \bmod (3^n - 1) = \end{aligned} \tag{4}$$

$$\begin{aligned} &\begin{bmatrix} (a_{2n-1} \ldots a_{n+1}a_n) \times (3^n - 1 + 1) \\ +(a_{n-1} \ldots a_1a_0) \end{bmatrix} \bmod (3^n - 1) = \\ &= (a_{2n-1} \ldots a_{n+1}a_n) + (a_{n-1} \ldots a_1a_0) \end{aligned} \tag{5}$$

In other words, we should add up each n digit partition with its succeeding n digit partition. Three cases may occur:

$$\begin{cases} 0 \le A_1 + A_2 < 3^n - 1 \\ 3^n - 1 \le A_1 + A_2 < 2(3^n - 1) \\ A_1 + A_2 = 2(3^n - 1) \end{cases}$$

In reality the weights of the i-th and (i+jn)th positions are equal where, (j = 1,2,...). In the first case, where, $0 \le A_1+A_2 < 2(3^n-1)$, the carry out will be equal to zero because the sum is less than the modulo. So, the result is valid and no extra process is needed. In the second case, where, $3^n-1 \le A_1+A_2 < 2(3^n-1)$, to correct the result we should increment the result by one. In the third case, where, $A_1+A_2 < 2(3^n-1)$, we should add 2 to the result to correct it. For speeding up the addition, we do this procedure in parallel and by using a multiplexer, the correct output will be chosen from the outputs of three adders. We know that if there is no carry out, then the sum is less than 3^n-1, so in this case, we can choose the corresponding adder easily. In the second case, where the result is between 3^n-1 and $2(3^n-1)$, the maximum value of carry out will be 1 even if 2 has been added to it. So this time, the related adder can be chosen too. In the last case, the carry out will be equal to 2, which means that we should select the last adder with an input of 2 as the carry in Fig. 5. represents the structure of these three parallel adders.

If we want to obtain the residue of an TVL number in the modulo 3^n-1 then:

$$\begin{aligned} &(a_{2n-1}\cdots a_{n+1}a_n a_{n-1}\cdots a_1 a_0) \bmod (3^n-2) = \\ &\begin{bmatrix} (a_{2n-1}\cdots a_{n+1}a_n)\times 3^n \\ +(a_{n-1}\cdots a_1 a_0) \end{bmatrix} \bmod (3^n-2) = \\ &\begin{bmatrix} (a_{2n-1}\cdots a_{n+1}a_n)\times (3^n-2+2) \\ +(a_{n-1}\cdots a_1 a_0) \end{bmatrix} \bmod (3^n-2) = \\ &= \left[2\times(a_{2n-1}\cdots a_{n+1}a_n)+(a_{n-1}\cdots a_1 a_0)\right] \end{aligned}$$

In result, we should add up n most significant digits with the next n digits multiplied by two (2 times the next n digit). The positional weight of the (i+n) th position is 2 times of the ith position. So the positional weight of the carry out will be 2 times of the carry in Fig. 6, 7 shown the related circuit structures.

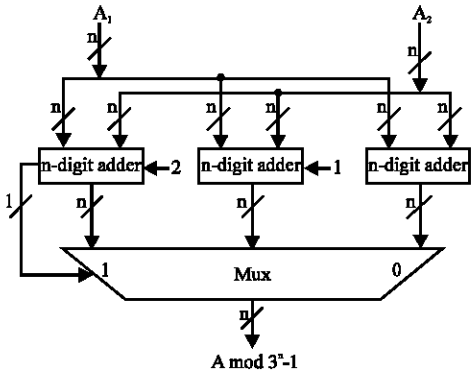

Fig. 5: Converter circuit from TVL to RNS 3^n-1 modulo

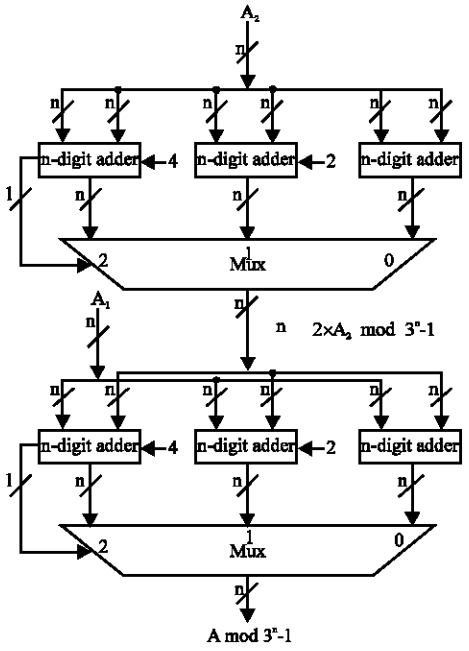

Fig. 6: Serial TVL to RNS converter for the modulo 3^n-2

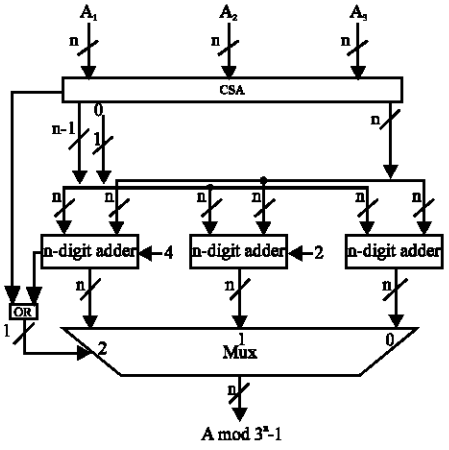

Fig. 7: Parallel TVL to RNS converter for the modulo 3^n-2

Conversion from RNS to TVL: To calculate the number X from its residues (x_1, x_2, x_3, ..., x_{L-1},x_L) by using the Chinese Reminder Theorem, the following equation is used:

$$X = \left\langle \sum_{i=1}^{L} (x_i N_i)_{m_i} \times M_i \right\rangle_M$$

$$M = \prod_{i=1}^{L} M_i$$

$$M_i = \frac{M}{m_i} \quad , \quad N_i = \left\langle M_i^{-1} \right\rangle_{m_i} \quad , \quad i = 1,2,3,...,L$$

in which $\langle M_i^{-1} \rangle_{m_i}$ is the multiplicative inverse of M_i in the modulo m_i.

For our proposed moduli set, we can rewrite the above equation as:

$$X = \left\langle \begin{array}{l} \left[(x_1 N_1)_{3^n-2'} 3^n - 1' 3^n\right] + \\ \left[(x_2 N_2)_{3^n-1'} 3^n - 2' 3^n\right] + \\ \left[(x_3 N_3)_{3^{n'}} 3^n - 2' 3^n - 1\right] \end{array} \right\rangle_M \quad (9)$$

$$= \left\langle \begin{array}{l} \left[(x_1 N_1)_{3^n-2'} (3^{2n} - 3^n)\right] + \\ \left[(x_2 N_2)_{3^n-1'} (3^{2n} - (2' 3^n))\right] + \\ \left[(x_3 N_3)_{3^{n'}} (3^{2n} - 3^{n+1} + 2)\right] \end{array} \right\rangle_M$$

To obtain $(x_i N_i)_{mi}$, we do the multiplication (x_i, N_i) by using a conventional multiplier. Then the residue will be simply obtained by using the conversion algorithm. In other words, these operations are done by using simple shift and add operations and finally the residue in the modulo M is obtained very simply. The complement of M is:

$$3^{3n}-(3^{3n}-3^{3n+1}+2' 3^n) = (3^{2n+1}-3^n) \quad (10)$$

By using a conventional adder, one can calculate the residue in the modulo M. It is necessary to mention that calculations are done in the radix 3. So the adders have 3 inputs. Without considering the details, the required circuit for the conversion from the RNS to the weighted radix 3 in the proposed moduli set has been shown in Fig. 8. In (Abdallah and Skavantzos, 2005) another method has been used. They have used a reduction factor which has been applied to the critical path (Fig. 9).

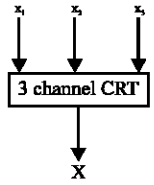

Fig. 8: Proposed RNS to TVL

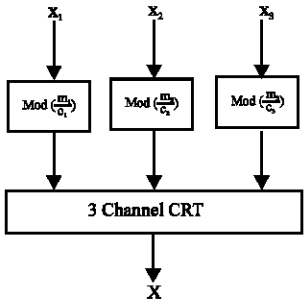

Fig. 9: RNS to TVL converter of (10) Comparison

EXAMPLE

In order to make the mentioned subject clear we propose an example. If we consider the RNS with moduli $\{3^3\text{-}2, 3^3\text{-}1, 3^3\}$ which is equal to {25, 26, 27}, as it is obvious the moduli are close enough and the difference between the largest and the smallest modulo is 2.

Consider two number X and Y:

$$X = (1012121)_3$$
$$Y = (0111021)_3$$

In order to convert numbers in radix 3 to RNS:

$$x_1 = \langle X \rangle_{3^3-2} = \langle 4\times(1) + 2\times(012) + 121 \rangle_{3^3-2} = 012$$
$$x_2 = \langle X \rangle_{3^3-1} = \langle 1 + 012 + 121 \rangle_{3^3-1} = 211$$
$$x_3 = \langle X \rangle_{3^3} = 121$$
$$y_1 = \langle Y \rangle_{3^3-2} = \langle 2\times(111) + 021 \rangle_{3^3-2} = 022$$
$$y_2 = \langle Y \rangle_{3^3-1} = \langle 111 + 021 \rangle_{3^3-1} = 202$$
$$y_3 = \langle Y \rangle_{3^3} = 021$$

For addition operation:

$$z_1 = \langle x_1 + y_1 \rangle_{3^3-2} = \langle 012 + 022 \rangle_{3^3-2} = 111$$
$$z_2 = \langle x_2 + y_2 \rangle_{3^3-1} = \langle 211 + 202 \rangle_{3^3-1} = 121$$
$$z_3 = \langle x_3 + y_3 \rangle_{3^3} = \langle 121 + 021 \rangle_{3^3} = 212$$

If we want to have the final number in radix-3 using the conversion of RNS to radix-3 number system:

$$N_i = \left\langle M_i^{-1} \right\rangle_{m_i}$$

Therefore:

$$Z = \left\langle \begin{matrix} (13\times N_1)_{25}\times 702 \\ +(16\times N_2)_{26}\times 675 \\ +(23\times N_3)_{27}\times 650 \end{matrix} \right\rangle_{17550} = 1238$$

After the reverse conversion the number 1238 will result.

Beside to verify the validation of the above method:

$$X = 880$$
$$Y = 358$$
$$Z = X+Y = 880+358 = 1238$$
$$\langle 1238 \rangle_{3^3-2} = 13 = 111$$
$$\langle 1238 \rangle_{3^3-1} = 16 = 121$$
$$\langle 1238 \rangle_{3^3} = 23 = 212$$

As is show both the method result in the same conclusion.

COMPARISONS

In the moduli set $\{r^a, r^b-1, r^c+1\}$ has been presented. In that work, a comprehensive study has been done to compare these moduli set to binary moduli sets. The comparison results demonstrate some improvements in terms of speed and dynamic range. As the comparison of Abdallah and Skavantzos (2005) was extensive, general and exhaustive, we only compare our proposed design to Abdallah and Skavantzos (2005). First we compare the dynamic ranges. As shown in Table 2, the dynamic range of our proposed design is nearly four times the one presented by Abdallah and Skavantzos (2005). One interesting advantage of our proposed moduli sets is that all moduli are pair wise relatively prime and therefore, we need no reduction factor. So the dynamic range is obviously greater.

In Table 3 the comparison has been done based on the memory utilization. As shown in this table, the proposed moduli set needs one less location to represent the same number.

For comparing the addition speeds, we suppose t_{FA3} be the delay of addition of two TVL locations. We assume the conventional adders are used but one can use any carry acceleration techniques. Table 4 shows the result of this comparison of different moduli sets.

Table 2: Comparison of dynamic ranges

Dynamic range	Moduli set
3^{3n}-3^{2n+1}+2´ 3^n	$\{3^n-2,3^n-1, 3^n\}$
$(3^{3n}-3^n)/4$	$\{3^n-1,3^n, 3^n+1\}$

Table 3: Comparison of the number of locations

No. of positions	Moduli set
n+n+n	$\{3^n-2,3^n-1, 3^n\}$
n+n+(n+1)	$\{3^n-1,3^n, 3^n+1\}$

Table 4: Comparison of delay in addition

Maximum delay	Modulo
$nt_{FA3}+t_{MUX(2´1)}$	3^n-2
$nt_{FA3}+t_{MUX(2´1)}$	3^n-1
nt_{FA3}	3^n
$(n+1)t_{FA3}+t_{MUX(2´1)}$	3^n-1

Table 5: Comparison of delay conversion RNS to TVL

Dealy	Moduli set
t_{3CRT}	$\{3^n-2,3^n-1, 3^n\}$
$t_{Scale\text{-}Down\,Factor} + t_{SCRT}$	$\{3^n-1, 3^n, 3^n+1\}$

The maximum delays of the mentioned moduli set are:

$$\tau_{(3^n-2,3^n-1,3^n)} = n\tau_{FA3} + \tau_{MUX(2\times1)}$$
$$\tau_{(3^n-1,3^n,3^n+1)} = (n+1)\tau_{FA3} + \tau_{MUX(2\times1)}$$

So the new proposed moduli set are faster. In comparison to Abdallah and Skaventzos (2005), we have reduced the overall delay by the delay of an adder cell.

In the Table 5 the comparison of RNS to TVL delay conversion circuit is illustrated. It demonstrates that we have achieved a significant improvement in terms of overall delay.

CONCLUSION

In this study, a new moduli set $\{3^n-2, 3^n-1, 3^n\}$ has been represented for RNS. The moduli are pair wise relatively prime for all different values of n, so the dynamic range is greater than those of other similar moduli sets. The addition operation has been speeded up which consequently speeds up all other related operations. The conversion from the RNS to positional weighted system and vice versa is rapid. The memory is used optimally because in these moduli set only 3 units of memory will be unused, while for the other moduli sets, memory waste is nearly 3^n units. The complements of the moduli presented in this study are {0, 1, 2} while the set $\{0, 1, 2r^r-1\}$ must be used as the complement in Abdallah and Skaventzos (2005). We have used n positional circuits for all moduli whereas in the other moduli sets, n+1 positional circuit must be used.

REFERENCES

Abdallah, M. and A. Skavantzos, 2005. On multi moduli residue number systems with moduli of forms $\{r^a, r^b-1, r^c+1\}$. IEEE Trans. Circuits Syst. I: Regular Paper, 52: 1253-1266.

Bhardwaj, M., A.B. Premkumar and T. Srikanthan, 1998. Breaking the 2n-bit carry propagation barrier in residue to binary conversion for the $\{2^n-1, 2^n, 2^n+1\}$ module set. IEEE Trans. Circuits Syst. II, 45: 998-1002.

Cao, B., C.H. Chang and T. Srikanthan, 2007. A residue-to-binary converter for a new five-moduli set. IEEE Trans. Circuits Syst. -I: Regular Papers, 54: 5.

Conway, R. and J. Nelson, 2004. Improved RNS fir filter architectures. IEEE Trans. Circuits Syst. II, 51: 26-28.

Freking, W.L. and K.K. Parhi, 1997. Low-power FIR digital filters using residue arithmetic. 31st Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA., 1: 739-743.

Gallaher, D., F. Petry and P. Srinivasan, 1997. The digital parallel method for fast RNS to weighted number system conversion for specific moduli $\{2^n-1, 2^n+1\}$. IEEE Trans. Circuits Syst. II, 44: 53-57.

Garner, H., 1959. The residue number system. IEEE. Trans. Electron. Comput., 8: 140-147.

Gonzalez, A.F. and P. Mazumdar, 2000. Redundant arithmetic, algorithms and implementations. Integration. The VLSI J., 30: 13-53.

Hariri, A., K. Navi and R. Rastegar, 2005. A simplified modulo (2^n-1) squaring scheme for residue number system. IEEE International Conference on Computer as a Tool.

Hiasat, A. and H. Zohdy, 1998. Residue-to-binary arithmetic converter for the moduli $\{2^n, 2^n-1, 2^{n-1}\}$. IEEE Trans. Circuits Syst. II, 45: 204-209.

Hosseinzadeh, M., K. Navi and S. Timarchi, 2006a. Design residue number system circuits in current mode. 14th Iranian Conference of Electrical Engineering.

Hosseinzadeh, M., K. Navi and S. Timarchi, 2006b. New design of 4-3 compressor. 11th International CSI Computer Conference of Iran.

Hosseinzadeh, M., K. Navi and S. Gorgin, 2007. A new moduli set for RNS: $\{r^n-2, r^n-1, r^n\}$. International Conference on Electrical Engineering.

Hurst, S.L., 1984. Multiple-valued logic-Its status and its future. IEEE. Trans. Comput., pp: 1160-1179.

Kinoshita, E. and K. Lee, 1997. A residue arithmetic extension for reliable scientific computation. IEEE Trans. Comput., 46: 129-138.

Paliouras, V. and T. Stouraitis, 2000. Novel high-radix residue number system architectures. IEEE Trans. Circuits Syst. II: Analog Digital Signal Processing, 47: 1059-1073.

Parhami, B., 2001. Computer Arithmetic: Algorithms and Hardware Designs, Oxford.

Ramirez, J. *et al*., 2002. Fast RNS FPL-based communications receiver design and implementation. Proceeding 12th International Conference Field Programmable Logic, pp: 472-481.

Sheu, M.H., S.H. Lin, C. Chen and S.W. Yang, 2004. An efficient vlsi design for a residue to binary converter for general balance moduli $\{2^n-3, 2^n+1, 2\}$. IEEE Trans. Circuits Syst. II: Express Briefs, 51: 152-155.

Soderstrand, M.A. *et al*., 1986. Residue Number System Arithmetic: Modern Applications in Digital Signal Processing. IEEE Press, New York.

Szabo, N. and R. Tanaka, 1967. Residue Arithmetic and its Applications to Computer Technology. McGraw-Hill, New York.

Taylor, F., 1985. A single modulus ALU for signal processing. IEEE Transactions on Acoustics, Speech, Signal Processing, 33: 1302-1315.

Timarchi, S., K. Navi and M. Hosseinzadeh, 2006. New design of RNS subtractor for modulo $\{2^n+1\}$. 2nd IEEE International Conference on Information and Communication Technologies: From Theory To Applications.

Wang, W. *et al*., 2000. A high-speed residue-to-binary converter for three-moduli $\{2^n.2^n-1, 2^{n-1}-1\}$ RNS and a scheme for its VLSI implementation. IEEE Trans. Circuits Syst. II, 47: 1576-1581.

Wang, Y. *et al*., 2002. Adder based residue to binary number converters for $\{2^n-1, 2^n-2^n+1\}$. IEEE Trans. Signal Process., 50: 1772-1779.

Yen, S., S. Kim, S. Lim and S. Moon, 2003. RSA speedup with chinese remainder theorem immune against hardware fault cryptanalysis. IEEE. Trans. Comput., 22: 461-472.