2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

Residue to Weighted Converter for the Quinary Moduli Set $\{5^n - 2, 5^n - 1, 5^n\}$

Hasan Amin Oseily[†] & Ali Massoud Haidar[††]
[†]Electrical Department, [††] Computer Department, Faculty of Engineering
Beirut Arab University, Beirut-Lebanon
E-mail: sasoha@yahoo.com, ari@bau.edu.lb

Abstract— The residue number system (RNS) is a carry-free number system which can support high-speed and parallel arithmetic. One of the major issues in efficient design of RNS systems is the residue to weighted conversion which is an important issue concerning the utilization of RNS numbers in digital signal processing (DSP) applications. We present here an efficient design of residue to weighted converter for the newly introduced quinary moduli set $\{5^n - 2, 5^n - 1, 5^n\}$, based on mixed-radix conversion (MRC) algorithm. The proposed residue to weighted converter is adder-based and memory-less which can result in high-performance hardware. The proposed residue to weighted converter has better performance and also eliminates the use of multiplier, compared to the last work [10].

1. Introduction

The usage of Residue Number System (RNS) in Digital Signal Processing (DSP) applications has received considerable attention due to its attractive carry-free property which yields arithmetic processors that are inherently parallel, modular and fault isolating [1],[2],[3]. For successful application of RNS, data conversion must be very fast so that the conversion overhead doesn't nullify the RNS advantages [3]. The residue number system (RNS) is a non-weighted number system which speeds up arithmetic operations by dividing them into smaller parallel operations. Since the arithmetic operations in each moduli are independent of the others, there is no carry propagation among them and so RNS leads to carry-free addition, multiplication and borrow-free subtraction [4]. The RNS is mostly used in encryption and decryption techniques for its advantages in the computation process. One of the major issues in efficient design of RNS systems is the residue to weighted conversion. The algorithms of residue to weighted conversion are mainly based on chinese remainder theorem (CRT), mixed-radix conversion (MRC) [4] and new chinese remainder theorems (New CRTs) [5]. In addition to these, novel conversion algorithms [6] which are designed for some special moduli sets have been proposed. Multiple-valued logic (MVL) has been proposed as a means for reducing the power, improving the speed, and increasing the packing density of VLSI circuits [7]. In MVL, the number of discrete signal values or logic states extends beyond two. Arithmetic units implemented with MVL achieve more efficient use of silicon resource and circuit interconnections [8]. There is a clear mathematical attraction of using multiple-valued number representation in RNS. The modular arithmetic that is inherent in MVL can be match with modular arithmetic needed in RNS. The first MVL-RNS system was introduced by Soderstrand [9] to design a high speed Finite Impulse Filter (FIR). The residue to weighted converter proposed in [9] is based on chinese remainder theorem (CRT) and implemented with read-only memories (ROM's). This converter is practical to implement small and medium RNS dynamic ranges and it is not appropriate for large dynamic ranges. This paper develops a two-level MRC algorithm for designing an efficient residue to weighted converter for the moduli set $\{5^n - 2, 5^n - 1, 5^n\}$. The proposed hardware architecture for residue to weighted converter has better performance in terms of area and delay since it is multiplier-free and memory-less in comparison with the residue to weighted converter proposed in [10].

2. Background

A residue number system is defined in terms of a relatively-prime moduli set $\{P_1, P_2, \dots, P_n\}$ that is greater common divisor GCD $(P_i, P_j)= 1$ for $i \neq j$ and $i, j = 1,2,\dots,n$. A weighted number X can be represented as $X = (x_1,x_2, \dots ,x_n)$, 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 $[0, M\text{-}1]$, where $M=P_1P_2\dots P_n$ is the dynamic range of the moduli set $\{P_1,P_2, \dots,P_n\}$ [10].

Then, the equivalent representation of X=32 is (x1, x2, x3) = (2, 2, 4). Addition, subtraction and multiplication on residues can be performed in parallel without any carry propagation among the residue digits. Hence, by converting the arithmetic of large numbers to a set of the parallel arithmetic of smaller numbers, the RNS representation yields significant speed up.

The algorithms of residue to weighted conversion are based mainly on Chinese remainder theorem (CRT) and mixed-radix conversion (MRC).

1) *Chinese Remainder Theorem:* by CRT, the number *X* is calculated from residues by

$$X = \left| \sum_{i=1}^{n} \left| x_i N_i \right|_{P_i} M_i \right|_M \quad (2)$$

where $M_i = M / _{Pi}$ and $N_i = | M_i^{-1} |_{Pi}$ is the multiplicative inverse of M_i modulo P_i.

2) *Mixed-Radix Conversion:* the weighted number *X* can be computed by

$$X = a_n \prod_{i=1}^{n} (P_i + \ldots + a_3 P_2 P_1 + a_2 P_1 + a_1) \quad (3)$$

Where $a_i s$ are called the mixed-radix coefficients and they can be obtained from the residues by

$$a_n = \left| (((x_n - a_1) | P_1^{-1} |_{Pn} - a_2) | P_2^{-1} |_{Pn} - \ldots - a_{n-1}) | P_{n-1}^{-1} |_{Pn} \right|_{Pn} \quad (4)$$

Where $n > 1$ and $a_1 = x_1$.
For a simple 2-moduli set $\{P_1, P_2\}$, the number *X* can be converted from its residue representation (x_1, x_2) by

$$X = a_1 + a_2 P_1 = x_1 + P_1 \left| (x_2 - x_1) \left| P_1^{-1} \right|_{P2} \right|_{P2} \quad (5)$$

where $| P_1^{-1} |_{P2}$ is the multiplicative inverse of P_1 modulo P_2.

3. RNS with Moduli Set {5ⁿ – 2, 5ⁿ – 1, 5ⁿ }

In [10], a ternary moduli set $\{3^n - 2, 3^n - 1, 3^n\}$ was introduced for RNS. Here we will introduce also the moduli set with quinary numbers set $\{5^n - 2, 5^n - 1, 5^n\}$. This moduli set contains pair-wise relatively prime and balanced moduli which can offer large dynamic range and fast internal RNS processing. Because of using of high radix ($r = 5$), this RNS can be simply realized in quinary-valued logic (QVL). Addition circuits for moduli set $\{5^n - 2, 5^n - 1, 5^n\}$ can be obtained by using the same method of [11]. If we consider three numbers *A*, *B* and *C* as the residues in respect of the modulo *m*, then addition of these numbers in modulo *m*, can be performed as

$$\begin{aligned} &A + B + C < m \Rightarrow A + B + C \\ &A + B + C \geq m \Rightarrow A + B + C - m \end{aligned} \quad (6)$$

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.

Example 1: If we perform the addition operation on the residues of the two number X(2,2,4) and Y(0,4,3), we found that
2+0 = 2 because 2+0 < 3 (then the residue is 2)
2+4≥5 then the residue is 2+4-5 = 1
4+3≥7 then the residue is 4+3-7=0
The final residue of addition is (2,1,0)
For performing the addition operation in the modulo 5^n, we add up two numbers and as the carry out is a multiple of $5^{n,}$ we simply ignore the carry out. The corresponding circuit is illustrated in Figure 1.

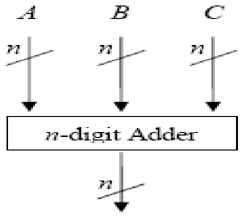

Figure 1. Modulo 5^n Quinary Adder

In modulo 5^n–1 if the result is greater than or equal to the 5^n–1 then the result will be added to the complement of the modulo, i.e. $5^n - (5^n-1) = 1$. By 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 presented in Figure 2.

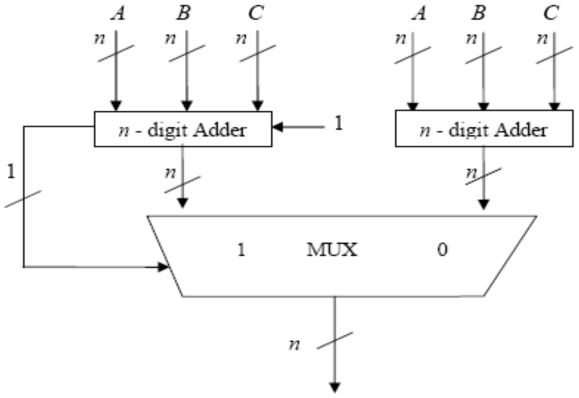

Figure 2. Modulo 5^n-1 Quinary Adder

In modulo 5^n–2, if the result of addition is greater than or equal to 5^n–2 then it will be added to the complement of the modulo which is $5^n-(5^n-2) = 2$. In an adder in the base 5, carry in can be between zero and 4. Figure 3 shows the circuit of this modulo 5^n–2 adder.

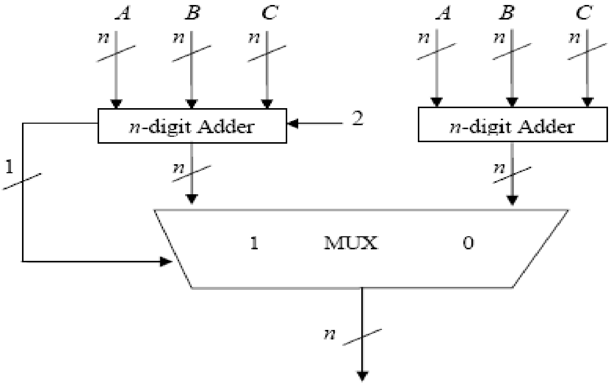

Figure 3. Modulo 5^n-2 Quinary Adder

We propose a two-level conversion algorithm for the residue to weighted conversion of the moduli set $\{5^n - 2, 5^n - 1, 5^n\}$. In the first level we use a MRC block for combining the two residues. The second level consists of another MRC block combining the result of the first level with the third residue. Figure 4. shows the block diagram of the proposed residue to weighted converter.

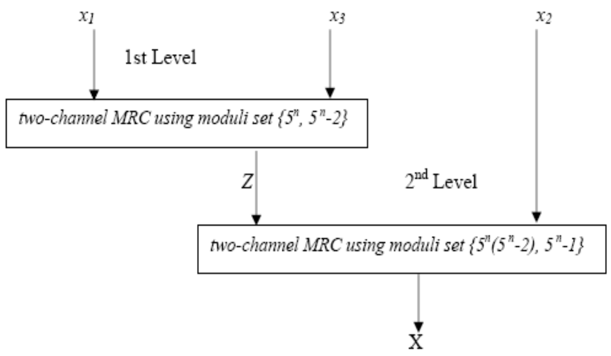
x_1
x_3
x_2
1st Level
two-channel MRC using moduli set $\{5^n, 5^n\text{-}2\}$
Z
2nd Level
two-channel MRC using moduli set $\{5^n(5^n\text{-}2), 5^n\text{-}1\}$
X

Figure 4. Block diagram of the proposed converter

The following propositions are needed for the derivation of our algorithm.

Proposition 1: the multiplicative inverse of $5^n – 2$ modulo 5^n is $k_0 = (5^n – 1) / 2$.

Proof: : it is clear that $\left| 5^n -2 \right|_{5^n} = -2$, so

$\left| k_0 \times (5^n - 2) \right|_{5^n} = \left| ((5^n -1)/2) \times (5^n -2) \right|_{5^n}$

$= \left| -1/2 \times -2 \right|_{5^n} = 1$ (7)

Proposition 2: the multiplicative inverse of $5^n (5^n – 2)$ modulo $5^n – 1$ is $k_1 = –1$.

Proof:

Since $\left| 5^n \right|_{5^n-1} = 1$ and $\left| 5^n-2 \right|_{5^n} -1 = -1$, *we have*

$\left| k_1 \times 5^n \times (5^n - 2) \right|_{5^n-1} = \left| -1 \times 5^n \times (5^n - 2) \right|_{5^n-1}$ (8)

$= \left| -1 \times 1 \times (-1) \right|_{5^n-1} = 1$

Consider the three-moduli set $\{5^n – 2, 5^n – 1, 5^n\}$ and let the corresponding residues of the integer X be (x_1, x_2, x_3). onsider the moduli set $\{5^n – 2, 5^n\}$ and $Z = (x_1, x_3)$. Using the MRC conversion algorithm (5), Z can be calculated by

$Z = x_1 + (5^n\text{-}2) \left| k_0 (x_3\text{-}x_1) \right|_{5^n}$ (9)

Where

$\left| k_0 (5^n\text{-}2) \right|_{5^n} = 1$ (10)

Substituting the value of k_0 from proposition (1) into (9) gives

$Z = x_1 + (5^n\text{-}2) \left| ((5^n\text{-}1)/2) \times (x_3\text{-}x_1) \right|_{5^n}$ (11)

The above equation can be rewritten as

$Z = x_1 + (5^n\text{-}2)T$ (12)

Where

$T = \left| ((5^n\text{-}1)/2) \times (x_3\text{-}x_1) \right|_{5^n}$ (13)

We know that

$(5^n\text{-}1) / 2 = 2\text{x} (5^o + 5^1 + \ldots + 5^{(n-1)})$ (14)

Therefore (13) can be written as,

$T = \left| 2\text{x}(5^o + 5^1 + \ldots + 5^{(n-1)}) \times (x_3\text{-}x_1) \right|_{5^n}$

$= \left| 2 \text{ x } (5^o + 5^1 + \ldots + 5^{(n-1)}) \times V \right|_{5^n}$

$= \left| 2\text{x} (V^o + V^1 + \ldots + V^{(n-1)}) \right|_{5^n}$ (15)

Where

$V = \left| x_3\text{-}x_1 \right|_{5^n}$ (16)

V^i s in Equation (15) can be obtained by the i digit left shifting of V. Since the final result of the addition of V^i temrs must be reduced in modulo 5^n, we only need to consider the least significant n digits of V^i terms, and the other digits are ignored as they are multiples of 5^n. The equation (12) can be rewritten as

$Z = x_1 + 5^n T - 2T$ (17)

Now, consider the moduli set $\{5^n(5n–2), 5^n–1\}$and $X=(Z,x_2)$. Using the derivation like before, X can be calculated by

$X = Z + 5^n (5^n - 2) \left| k_1(x_2 - Z) \right|_{5^n-1}$ (18)

Where

$\left| k_1 \times 5^n (5^n - 2) \right|_{5^n-1} = 1$ (19)

By substituting the value of k_1 from proposition 2, we have

$X = Z + 5^n (5^n - 2) \left| Z\text{-} x_2 \right|_{5^n-1}$ (20)

So, (20) can be rewritten as

$X = Z + 5^{2n} D - 5^n 2D = (Z + 5^{2n} D) + 5^n (-D -D)$ (21)

Where

$D = \left| Z\text{-} x_2 \right|_{5^n-1}$ (22)

Since Z is a $2n$-digit number, we can write

$D = \left| Z_1 5^n + Z_0 - x_2 \right|_{5^n-1} = \left| Z_1 + Z_0 - x_2 \right|_{5^n-1}$ (23)

Where Z_1 and Z_0 have digit level representation as

$Z_1 = (z_{2n-1} \ldots z_{n+1} z_n)$ (24)

$Z_0 = (z_{n-1} \ldots z_1 z_0)$ (25)

Example 2: Given the moduli set $\{5^n– 2, 5^n – 1, 5^n\}$ where $n=2$. The residue number (1,4,2) converted into its equivalent weighted number as follows: For $n=2$ the moduli set is {23,24,25}. So, by substituting values in (11) and (20) we have

$Z = 1 + 23 \left| 12 \times 1 \right|_{25} = 277$, $X = 277 + 25 \times 23 \left| 277 - 4 \right|_{24} = 5452$

To verify the result, we have

$x_1 = \left| 5452 \right|_{23} = 1$, $x_2 = \left| 5452 \right|_{24} = 4$, $x_3 = \left| 5452 \right|_{25} = 2$

Therefore, the weighted number 5452 has RNS representation as (1,4,2) in the RNS with moduli set {23,24,25}.

4. Hardware Implementation

The MRC block of the first level are represented by equations (15)–(17) whereas equations (21) and (23) represent the MRC block of the second level. Details on the first-level and second-level are as follow.

1) The First Level: Equation (16) can be calculated by a regular n-digit quinary adder. Then, (15) is implemented by an *n*-digit quinary multi-operand adder which is consists of a *n*-digit quinary carry save adder (CSA) tree followed by a regular *n*-digit quinary adder. Finally, (17) can be calculated by a 2*n*-digit regular quinary adder. It should be noted that since x_1 is an *n*-digit number, no extra hardware is needed for computation of x_1+5^nT. The desired result can be obtained by concatenating *x*1 with *T*. Figure 5(a) shows the hardware implementation of the first level of the residue to weighted converter.

2) The Second Level: Equation (23) can be performed by an *n*-digit modulo (5^n–1) quinary adder which is shown in Fig. 2. Calculation of equation (21) relies on an *n*-digit quinary adder followed by a 5*n*-digit regular quinary adder. Like before, since *Z* is a 2*n*-digit number, no extra hardware is needed for computation of $Z+5^{2n}D$. Figure 5(b) shows the hardware implementation of the second level of the residue to weighted converter.

As shown in Figure 5(a) and (b), the proposed residue to weighted converter for the moduli set $\{5^n - 2, 5^n - 1, 5^n\}$ is multiplier-free and consists of quinary adders.

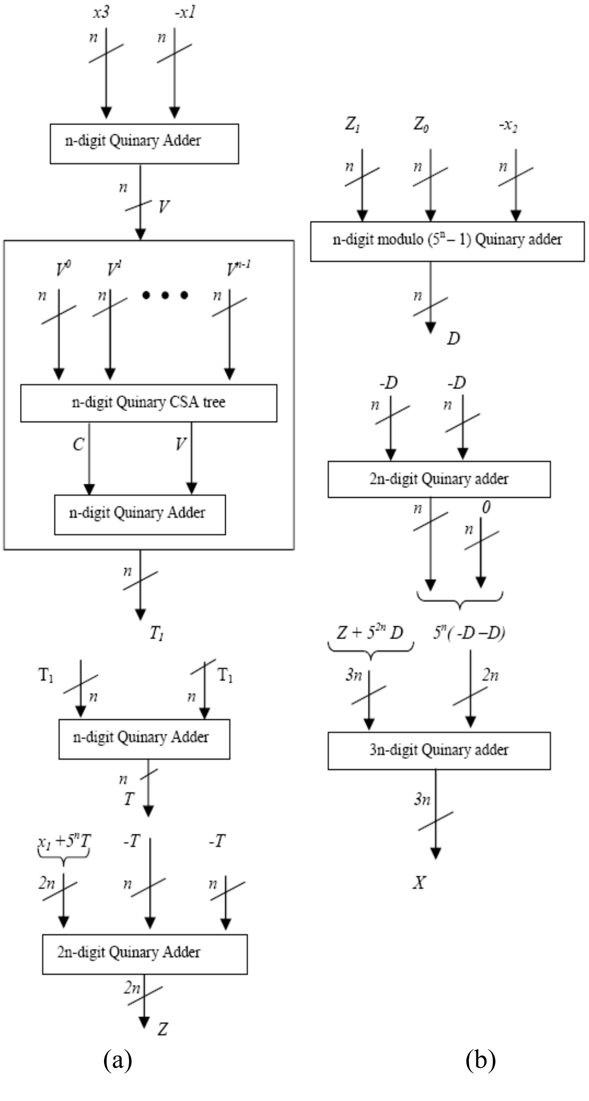

Figure 5. Hardware architecture of the first level (a) and the second level (b) of the converter

5. Results

The residue to weighted converter for the moduli set $\{3^n – 2, 3^n – 1, 3^n\}$ which is presented in [10], is based on direct implementation of the CRT algorithm and requires *n*-digit ternary multipliers and a modulo $(3^n – 2)\ (3^n – 1)\ (3^n)$ ternary adder for final reduction. So, as a result, the converter of [10] achieve long conversion delay and high hardware cost. But the larger modulo adder used in our converter is a modulo (5^n–1) adder and also the proposed design eliminates the use of multiplier. Therefore, the proposed residue to weighted converter has better performance than the residue to weighted converter of [10] due to the reduction of delay and hardware cost for more than 100% for the same converted number. For larger moduli set than $\{5^n – 2, 5^n – 1, 5^n\}$, the same procedures will be followed to conclude the conversion method but with some differences in the equation of Z. The proposed hardware that can implement this conversion method is FPGA (field programmable gate array)

6. Conclusion

In this chapter an efficient design of the residue to weighted converter for the moduli set $\{5^n – 2, 5^n – 1, 5^n\}$ is presented. The proposed hardware implementation of the residue to weighted converter is multiplier-free and memory-less, which can be efficiently implemented in VLSI. In comparison with the last residue to weighted converter for the moduli set $\{3^n – 2, 3^n – 1, 3^n\}$, the proposed design has better performance especially that quinary is easier than other systems for conversion into decimal (decimal is multiplier of quinary).

References

[1] H.L. Garner, The residue Number System, IRE Trans. On Electronic Computers, pp. 140-147, 1959.

[2] Szabo, N., and Tanaka, R. : Residue arithmetic and its application to computer technology, McGraw-Hill, New York, 1967.

[3] M. A. Soderstrand,W. K. Jenkins, G. A. Jullien, and F. J. Taylor, Residue Number System Arithmetic: Modern Applications in Digital Signal Processing. New York, IEEE press, 1986.

[4] B. Parhami, Computer arithmetic: algorithms and hardware designs, Oxford University Press, 2000.

[5] Y. Wang, Residue-to-Binary Converters Based on New Chinese remainder theorems, IEEE Trans. Circuits Syst.-II, 47, 197-205, 2000.

[6] M. Hosseinzadeh, A. S. Molahosseini, K. Navi, A Fully Parallel Reverse Converter, International Journal of Electrical, Computer, and Systems Engineering, 1 , 183-187, 2007.

[7] K.W. Current, V.G. Oklobdzija, D. Maksimovic, Low-energy logic circuit techniques for multiple valued logic, Proceedings of 26th International Symposium on Multiple-Valued Logic, pp.86-90, 1996.

[8] E. Dubrova, Multiple-Valued logic in VLSI: Challenges and opportunities, Proceedings of NORCHIP'99, Norway, pp.340-350, 1999.

[9] M. A. Soderstrand and R. A. Escott, VLSI implementation in multiple-valued logic of an FIR digital filter using residue number system arithmetic, IEEE Trans. Circuits Syst., 33, pp.5–25, 1986.

[10] M. Hosseinzadeh and K. Navi, A New Moduli Set for Residue Number System in Ternary Valued Logic, Journal of Applied ciences, 7 , pp.3729-3735, 2007.