# MODULI SELECTION GUIDELINES FOR EFFICIENT RESIDUE-TO-DECIMAL CONVERSION 

KAZEEM ALAGBE GBOLAGADE<br>Department of Applied Mathematics and Computer Science<br>University for Development Studies<br>Navrongo, Ghana<br>e-mail: gkazy1@yahoo.com


#### Abstract

This paper investigates selection of moduli in Residue Number System (RNS) which is an important issue in the design of digital systems. We propose Moduli Selection Guidelines (MSG) for moduli sets $$
\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}, \quad\{2 n+1,2 n, 2 n-1\} \quad \text { and } \quad\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\} .
$$

We design a program based on the MSG for computing all possible multiplicative inverses that may be needed when using different conversion techniques. Based on this experiment, we deduce that there is a well established relationship between the form of the moduli set, the moduli and the multiplicative inverses. These deductions are used to come up with the moduli representations of the various multiplicative inverses. Hence, using the MSG, the cost of computing the multiplicative inverses is eliminated. The experimental results reveal that some of these multiplicative inverses will always be unity so their computations are no more needed. Consequently, the usage of these guidelines in building a RNS-to-decimal converter results in a considerable reduction in the number of arithmetic operations required during the conversion process. These results provide the possibility of a wide range of applications of RNS in Digital Signal Processing.


Keywordsandphrases: residue number system, moduli selection guidelines, RNS-decimal converter, arithmetic operations.

## I. Introduction

The use of RNS in highly intensive computation has received considerable attention in the last two decades. RNS is a UNS with inherent parallel characteristics which supports carry-free addition, borrow-free subtraction and single step multiplication without partial products [2-4]. RNS has not found widespread use in general-purpose signal processor architecture because the following operations are very difficult to perform: magnitude comparison, overflow detection, sign detection and also division are generally slow. However, where these operations are not needed, special-purpose Digital Signal Processing (DSP) architectures based on RNS have been built [1]. In addition, intermodular operation and conversion between numbers are awkward and hence RNS is not widely used. The input numbers provided are either in standard binary or decimal which need to be converted to RNS before performing the operations and which must be presented in the same way as the input at the end of the operations. This implies that data conversion is greatly required. Generally, conversion process is very slow. For RNS processor to compete favourably with the conventional processor, a very fast converter is required. Moduli selection is of crucial importance in RNS design because the speed and the hardware complexity of RNS architecture are affected by the moduli selected. Particularly, it is the magnitude of the largest modulus that determines the speed of the resulting RNS architecture. This implies that moduli must be selected in such a way that the largest modulus will be made as small as possible. Another factor which contributes to the cost of building a converter is computation of multiplicative inverses. The cost of computing multiplicative inverses in the existing converters is very high. It is in this line of reasoning that we propose MSG for moduli sets

$$
\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}, \quad\{2 n+1,2 n, 2 n-1\} \quad \text { and } \quad\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}
$$

which result in the elimination of the computation of the multiplicative inverses required by the existing conversion techniques such as the Mixed Radix Conversion (MRC), Chinese Remainder Theorem (CRT), Yassine and Moore's MRC and so on. We wrote a C++ program based on the MSG, the results of which reveal that there is a well established relationship between the form of moduli set, the moduli and the multiplicative inverses. With this, we come up with the moduli representations of the various multiplicative inverses. Consequently, the usage of the MSG in building an RNS-to-decimal converter results in a considerable reduction in the number of arithmetic operations required during the conversion process.

The rest of the article is organized as follows: In Section II, we introduce the necessary background. Section III presents MSG. Section IV gives the effect of the MSG on some existing conversion methods. In Section V, we briefly evaluate the performance of the MSG and the paper is concluded in Section VI.

## II. Background

RNS is an integer system having the capabilities of performing carry-free addition and multiplication, borrow-free subtraction, fault tolerance and in general high speed arithmetic [3]. The main characteristics of RNS is the fact that it is an Unweighted Number System (UNS) where digits have no ordering significance. This implies that performing arithmetic operations between RNS numbers depend solely on the corresponding digits of its suboperations. Errors do not propagate between RNS digits because the parallel operations are independent of each other [7]. All these properties make RNS suitable in building high speed special purpose processor. As mentioned in Section I, an efficient RNS-decimal converter is required for RNS processor to compete favourably with the conventional processor. The study of moduli selection is of paramount importance because the dynamic range, the speed as well as RNS system implementation depend on the form as well as the number of moduli chosen [6].

RNS is defined in terms of a set of relatively prime moduli set $\left\{m_{1}, m_{2}, m_{3}, \ldots, m_{n}\right\}$ such that $\operatorname{gcd}\left(m_{i}, m_{j}\right)=1$ for $i \neq j$, where gcd means the greatest common divisor of $m_{i}$ and $m_{j}$ while $M=\pi_{i=1}^{n} m_{i}$ is the dynamic range. The residues of a decimal number $X$ can be obtained as $x_{i}=|X|_{m_{i}}$ and represented as $\left(x_{1}, x_{2}, x_{3}, \ldots, x_{n}\right)$ and also $0 \leq x_{i}<m_{i}$. This representation is unique for any integer $X \in[0, M-1]$. We note here that in this paper, we use $|X|_{m_{i}}$ to denote the $X \bmod m_{i}$ operation and the operator $\Theta$ to represent the operation of addition, subtraction and multiplication. If there exists any two integers $K$ and $L$ represented by $K=\left(k_{1}, k_{2}, k_{3}, \ldots, k_{n}\right)$ and $L=\left(l_{1}, l_{2}, l_{3}, \ldots, l_{n}\right)$. Given that $W=K \theta L$ and $W=\left(w_{1}, w_{2}, w_{3}, \ldots, w_{n}\right)$, then the arithmetic in RNS can be implemented as follows: $w_{i}=\left|k_{i} \theta l_{i}\right|_{m_{i}}$. For the sake of completeness, we briefly review the following conversion methods: CRT, MRC, and Yassine and Moore's MRC. We are aware that there are series of other methods emanating from either CRT or MRC. We only intend to study the effects of the MSG on these traditional methods and the
observation is also applicable to other conversion methods. The inclusion of Yassine and Moore's MRC is because it works in a very similar manner to MRC and more in particular, we observe that both MRC, and Yassine and Moore's MRC are exactly the same with regards to the number of arithmetic operations if our MSG is applied.

CRT. If we have the moduli set $\left\{m_{1}, m_{2}, m_{3}, \ldots, m_{n}\right\}$ and the dynamic range $M=\pi_{i=1}^{n} m_{i}$, then the residue number $\left(x_{1}, x_{2}, x_{3}, \ldots, x_{n}\right)$ is converted into the decimal number $X$ by

$$
X=\left.\left.\left|\sum_{i=1}^{n} M_{i}\right| M_{i}^{-1} x_{i}\right|_{m_{i}}\right|_{M},
$$

where $M_{i}=M / m_{i}$ and $M_{i}^{-1}$ is the multiplicative inverse of $M_{i}$ with respect to $m_{i}$. The purpose of the paper is to eliminate the cost of computing $M_{i}^{-1}$.

MRC. The conversion from RNS to decimal using MRC can be formulated as follows:

Given an $n$-digit number $X=\left(x_{1}, x_{2}, x_{3}, \ldots, x_{n}\right)$ in an RNS with the set of relatively prime integer moduli $\left\{m_{i}\right\}_{i=1, \ldots, n}$, find a set of digits $\left\{a_{1}, a_{2}, a_{3}, \ldots, a_{n}\right\}$, which are the Mixed Radix Digits (MRD) such that the following equation holds true:

$$
X=a_{1}+a_{2} m_{1}+a_{3} m_{1} m_{2}+\cdots+a_{n} m_{1} m_{2} m_{3} \cdots m_{n-1} .
$$

The MRD can be computed as follows [7]:

$$
\begin{aligned}
& a_{1}=x_{1} \\
& a_{2}=\left.\left.\left|\left(x_{2}-a_{1}\right)\right| m_{1}^{-1}\right|_{m_{2}}\right|_{m_{2}} \\
& a_{3}=\left.\left.\left|\left(\left(x_{3}-a_{1}\right)\left|m_{1}^{-1}\right|_{m_{3}}-a_{2}\right)\right| m_{2}^{-1}\right|_{m_{3}}\right|_{m_{3}} \\
& \vdots \\
& a_{n}=\left.\left.\left|\left(\left(\cdots\left(x_{n}-a_{1}\right)\left|m_{1}^{-1}\right|_{m_{n}}-a_{2}\right)\left|m_{2}^{-1}\right|_{m_{n}}-\cdots-a_{n-1}\right)\right| m_{n-1}^{-1}\right|_{m_{n}}\right|_{m_{n}}
\end{aligned}
$$

Again, the purpose of this paper with this conversion technique is to eliminate the cost of computing the multiplicative inverses and also to reduce the number of arithmetic multiplications required during the conversion process.

Yassine and Moore's MRC. Suppose a number system is defined by the set of
moduli $\left\{m_{1}, m_{2}, m_{3}, \ldots, m_{n}\right\}$, associated $\operatorname{MRD}\left(a_{1}, a_{2}, a_{3}, \ldots, a_{n}\right)$ of a given residue number $\left(x_{1}, x_{2}, x_{3}, \ldots, x_{n}\right)$ must be computed. It has been shown that it is more efficient to find the related variable $U_{i}$ instead of finding $a_{i}$ directly and $a_{i}$ is given by $a_{i}=\left|U_{i} V_{i}\right|_{m_{i}}$, where $V_{i}$ are constant predetermined factors given by

$$
\begin{aligned}
& V_{1} \equiv 1 \\
& V_{2} \equiv\left|\left(m_{1}\right)^{-1}\right|_{m_{2}} \\
& V_{3} \equiv\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}} \\
& \vdots \\
& V_{n} \equiv\left|\left(m_{1}, m_{2}, m_{3}, \ldots, m_{n-1}\right)^{-1}\right|_{m_{n}}
\end{aligned}
$$

and $U_{i}$ are computed by using the formula:

$$
\begin{aligned}
& U_{1}=x_{1} \\
& U_{2}=\left|x_{2}-x_{1}\right|_{m_{2}} \\
& U_{3}=\left.\left.\left|x_{3}-x_{1}-W_{2}\right| U_{2} V_{2}\right|_{m_{2}}\right|_{m_{3}} \\
& U_{4}=\left.\left|x_{4}-x_{1}-W_{2}\right| U_{2} V_{2}\right|_{m_{2}}-\left.W_{3}\left|U_{3} V_{3}\right|_{m_{3}}\right|_{m_{4}} \\
& \vdots \\
& U_{n}=\left.\left|x_{n}-x_{1}-W_{2}\right| U_{2} V_{2}\right|_{m_{2}}-W_{3}\left|U_{3} V_{3}\right|_{m_{3}}-\cdots-\left.W_{n-1}\left|U_{n-1} V_{n-1}\right|_{n-1}\right|_{m_{n}}
\end{aligned}
$$

The decimal equivalent to the residues will be given by

$$
X=\sum_{i=1}^{n} U_{i} W_{i}
$$

where $W_{j}=\prod_{i=1}^{j-1} m_{i}$ and $W_{1}=1, \quad j=2,3,4, \ldots$.
This algorithm has been compared earlier with Szabo and Tanaka's MRC algorithm and it was stated that there is a reduction in the number of arithmetic operations required during the conversion process in this algorithm compared to the traditional MRC. Using the MSG proposed in this paper, we observe that there is no
difference in these two techniques with regards to the number of arithmetic operations required during the conversion process.

## III. Moduli Selection Guidelines (MSG)

In this section, we present selection guidelines for moduli sets

$$
\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}, \quad\{2 n+1,2 n, 2 n-1\} \quad \text { and } \quad\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}
$$

With these three moduli sets, two things are common apart from the fact that they are either power of 2 moduli or factor of 2 moduli sets and these two things are: (i) $m_{1}>m_{2}>m_{3}>1$, (ii) $m_{1}-m_{2}=1$. If moduli sets are arranged in such a way that points (i) and (ii) are always true, then $\left|m_{1}^{-1}\right|_{m_{2}}=1$ will always be true. This implies that when building a converter that requires the computation of $\left|m_{1}^{-1}\right|_{m_{2}}$, this computation will be eliminated following the above arrangement. In this paper, we divide the guidelines into three similar classes, one for each of the three moduli sets under considerations based on some assumptions and experimental results which will be explained in Section V. We are aware that there are many other moduli sets that exhibit similar properties and obey Conditions (i) and (ii). For the sake of brevity, we limit our discussion to the above three moduli and with little adjustment similar rules may apply to other moduli sets with similar characteristics.

Class 1. $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}, n \geq 2$.
(a) Based on this arrangement, the following assumptions must always hold true: (i) $m_{1}>m_{2}>m_{3}>1$, (ii) $m_{1}-m_{2}=m_{2}-m_{3}=1$. Assuming this is true then: (b) The computation of $\left|m_{1}^{-1}\right|_{m_{2}}=\left|m_{2}^{-1}\right|_{m_{3}}=1$ will no more be necessary if required in building a converter as they will always be unity. More specifically, (b) will always be true because Assumption (ii) is always true. (c) The computation of $\left|m_{1}^{-1}\right|_{m_{3}}$ and $\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}}$ always yields the same result which will be equal to $2^{n-1}$ which is also the same as $m_{2} / 2$ meaning that $m_{2} / 2$ can simply be used to replace these multiplicative inverses if their computation is required. Hence the usage of Euclidean algorithm for this purpose is no more necessary as the computation of the multiplicative inverses is totally eliminated. (d) The computation
of $\left|\left(m_{2} m_{3}\right)^{-1}\right|_{m_{1}}$ which gives $2^{n-1}+1$ and which is equivalent to $\left(m_{2} / 2+1\right)$ will no more be needed if required during the conversion process. This multiplicative inverse whenever required will only be replaced by $\left(m_{2} / 2+1\right)$. (e) The computation of $\left|\left(m_{1} m_{3}\right)^{-1}\right|_{m_{2}}$ which is $2^{n}-1$ and which is equivalent to $m_{3}$ will not be needed if required during the conversion process, $m_{3}$ will be used in replacing its value.

Class 2. $\{2 n+1,2 n, 2 n-1\}, n \geq 2$.
Based on this arrangement and the same assumptions as given in Class 1(a), then (a) The computation of $\left|m_{1}^{-1}\right|_{m_{2}}=\left|m_{2}^{-1}\right|_{m_{3}}=1$ will no more be needed since they will always be unity. (b) The computation of $\left|m_{1}^{-1}\right|_{m_{3}}$ and $\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}}$ always yields the same result which will be equal to $n$ which is also the same as $m_{2} / 2$. This implies that whenever $\left|m_{1}^{-1}\right|_{m_{3}}$ or $\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}}$ is required, $m_{2} / 2$ will be used. (c) The computation of $\left|\left(m_{2} m_{3}\right)^{-1}\right|_{m_{1}}$ which gives $n+1$ which is equivalent to $\left(m_{2} / 2+1\right)$ will no more be needed if required, instead $\left(m_{2} / 2+1\right)$ will be used. (d) The computation of $\left|\left(m_{1} m_{3}\right)^{-1}\right|_{m_{2}}$ which is $2 n-1$ which is equivalent to $m_{3}$ will not be needed if required, $m_{3}$ will be used instead.

Class 3. $\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}, n \geq 3$.
(a) Suppose that (i) $m_{1}>m_{2}>m_{3}>1$ as in Class 1(a)-(i) and (ii) $m_{1}-m_{2}=1$ but $m_{2}-m_{3} \neq 1$. Then (b) The computation of $\left|m_{1}^{-1}\right|_{m_{2}}=\left|m_{2}^{-1}\right|_{m_{3}}=1$ will not be needed again as they will always be unity. (c) The computation of $\left|m_{1}^{-1}\right|_{m_{3}}$ and $\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}}$ always yields the same result which will be equal to $2^{n-2}$ which is also the same as $m_{1} / 4$. This means that whenever $\left|m_{1}^{-1}\right|_{m_{3}}$ or $\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}}$ is required, $m_{1} / 4$ will be used instead. (d) The computation of $\left|\left(m_{2} m_{3}\right)^{-1}\right|_{m_{1}}$ which gives $2^{n-1}+1$ and which is equivalent to $\left(m_{3}+2\right)$ will no more be needed if required instead $\left(m_{3}+2\right)$ will be used. (d) The computation of $\left|\left(m_{1} m_{3}\right)^{-1}\right|_{m_{2}}$
which is $2^{n}-3$ which is equivalent to $m_{2}-2$ or $m_{1}-3$ will not be needed if required, $m_{2}-2$ or $m_{1}-3$ will be used instead.

We place the multiplicative inverses and their corresponding values for the three moduli sets in Tables I, II and III as shown below:

With the results summarized in Tables I, II and III, the moduli sets $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}$ and $\{2 n+1,2 n, 2 n-1\}$ have the same representational values for the multiplicative inverses and so they will have similar effects on conversion methods. In this line of reasoning, their examination will be carried out jointly in the next section and the moduli set $\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}$ which behaves slightly differently will be separately examined.

Table I. Multiplicative inverse values for $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}$

| $\mathrm{S} / \mathrm{N}$ | Multiplicative inverses | Equivalent values |
| :---: | :---: | :---: |
| 1 | $\left\|m_{1}^{-1}\right\|_{m_{2}}$ | 1 |
| 2 | $\left\|m_{2}^{-1}\right\|_{m_{3}}$ | 1 |
| 3 | $\left\|\left(m_{1} m_{2}\right)^{-1}\right\|_{m_{3}}$ | $m_{2} / 2$ |
| 4 | $\left\|m_{1}^{-1}\right\|_{m_{3}}$ | $m_{2} / 2$ |
| 5 | $\left\|\left(m_{2} m_{3}\right)^{-1}\right\|_{m_{1}}$ | $m_{2} / 2+1$ |
| 6 | $\left\|\left(m_{1} m_{3}\right)^{-1}\right\|_{m_{2}}$ | $m_{3}$ |

Table II. Multiplicative inverse values for $\{2 n+1,2 n, 2 n-1\}$

| $\mathrm{S} / \mathrm{N}$ | Multiplicative inverses | Equivalent values |
| :---: | :---: | :---: |
| 1 | $\left\|m_{1}^{-1}\right\|_{m_{2}}=\left\|m_{2}^{-1}\right\|_{m_{3}}$ | 1 |
| 2 | $\left\|m_{1}^{-1}\right\|_{m_{3}}=\left\|\left(m_{1} m_{2}\right)^{-1}\right\|_{m_{3}}$ | $m_{2} / 2$ |
| 3 | $\left\|\left(m_{2} m_{3}\right)^{-1}\right\|_{m_{1}}$ | $m_{2} / 2+1$ |
| 4 | $\left\|\left(m_{1} m_{3}\right)^{-1}\right\|_{m_{2}}$ | $m_{3}$ |

## IV. The Effects of MSG on Some Conversion Methods

In this section, we study the effects of the MSG on the three conversion methods described in Section II and we follow the same order:

CRT. The CRT presented in Section II can be described for moduli set of length three as follows:

$$
X=\left.\left|M_{1}\right| M_{1}^{-1}\right|_{m_{1}} x_{1}+M_{2}\left|M_{2}^{-1}\right|_{m_{2}} x_{2}+\left.M_{3}\left|M_{3}^{-1}\right|_{m_{3}} x_{3}\right|_{M}
$$

With the definition given in Section II, we have

$$
X=\left.\left|M_{1}\right|\left(m_{2} m_{3}\right)^{-1}\right|_{m_{1}} x_{1}+M_{2}\left|\left(m_{1} m_{3}\right)^{-1}\right|_{m_{2}} x_{2}+\left.M_{3}\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}} x_{3}\right|_{M}
$$

For the moduli sets $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}$ and $\{2 n+1,2 n, 2 n-1\}$, substituting the values for the multiplicative inverses presented in Table I into the above equation, we have

$$
X=\left|\left(\frac{m_{2}}{2}+1\right) M_{1} x_{1}+m_{3} M_{2} x_{2}+\frac{m_{2}}{2} M_{3} x_{3}\right|_{M}
$$

Table III. Multiplicative inverse values for $\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}$

| $\mathrm{S} / \mathrm{N}$ | Multiplicative inverses | Equivalent values |
| :---: | :---: | :---: |
| 1 | $\left\|m_{1}^{-1}\right\|_{m_{2}}=\left\|m_{2}^{-1}\right\|_{m_{3}}$ | 1 |
| 2 | $\left\|m_{1}^{-1}\right\|_{m_{3}}=\left\|\left(m_{1} m_{2}\right)^{-1}\right\|_{m_{3}}$ | $m_{1} / 4$ |
| 3 | $\left\|\left(m_{2} m_{3}\right)^{-1}\right\|_{m_{1}}$ | $m_{3}+2$ |
| 4 | $\left\|\left(m_{1} m_{3}\right)^{-1}\right\|_{m_{2}}$ | $m_{2}-2$ |

Similarly, for the moduli set $\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}$, we obtain

$$
X=\left|\left(m_{3}+2\right) M_{1} x_{1}+\left(m_{3}-2\right) M_{2} x_{2}+\frac{m_{1}}{4} M_{3} x_{3}\right|_{M}
$$

The above result produces a very efficient converter if used in conjunction with [5] where the large modulo $M$ has been reduced and here the multiplicative inverses have been eliminated. The cost of computing multiplicative inverses is no more
required when transforming residue operands to decimal/binary. For example, suppose we are given a residue number $(8,3,0)_{\mathrm{RNS}(9|8| 7)}$, the decimal equivalent can be obtained as follows:

Solution. From the definition given in Section II, we have $M_{1}=m_{2} m_{3}$, $M_{2}=m_{1} m_{3}$ and $M_{3}=m_{1} m_{2}$. Using the newly obtained equation for the moduli set $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}$ with these values, we obtain

$$
\begin{aligned}
& X=|5 * 56 * 8+7 * 63 * 3+4 * 72 * 0|_{504} \text { meaning that: } \\
& X=|3563|_{504}=35 .
\end{aligned}
$$

Next, we proceed to MRC: The MRC presented in Section II can be described for moduli set of length three as follows:

Given a 3-digit number $X=\left(x_{1}, x_{2}, x_{3}\right)$ in an RNS with the set of relatively prime integer moduli $\left\{m_{i}\right\}_{i=1,3}$, find a set of digits $\left\{a_{1}, a_{2}, a_{3}\right\}$, which are the MRD such that the following equation holds true:

$$
X=a_{1}+a_{2} m_{1}+a_{3} m_{1} m_{2} .
$$

The MRD can be computed as follows:

$$
\begin{aligned}
& a_{1}=x_{1}, \\
& a_{2}=\left.\left.\left|\left(x_{2}-a_{1}\right)\right| m_{1}^{-1}\right|_{m_{2}}\right|_{m_{2}}, \\
& a_{3}=\left.\left.\left|\left(\left(x_{3}-a_{1}\right)\left|m_{1}^{-1}\right|_{m_{3}}-a_{2}\right)\right| m_{2}^{-1}\right|_{m_{3}}\right|_{m_{3}} .
\end{aligned}
$$

Again, for the moduli sets $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}$ and $\{2 n+1,2 n, 2 n-1\}$, substituting the values for the multiplicative inverses presented in Table I into the above equation results into

$$
\begin{aligned}
& a_{1}=x_{1}, \\
& a_{2}=\left|\left(x_{2}-a_{1}\right)\right|_{m_{2}}, \\
& a_{3}=\left|\left(\frac{m_{2}}{2}\left(x_{3}-a_{1}\right)-a_{2}\right)\right|_{m_{3}} .
\end{aligned}
$$

In a similar manner, with the moduli set $\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}$, we obtain

$$
\begin{aligned}
& a_{1}=x_{1}, \\
& a_{2}=\left|\left(x_{2}-a_{1}\right)\right|_{m_{2}}, \\
& a_{3}=\left|\left(\frac{m_{1}}{4}\left(x_{3}-a_{1}\right)-a_{2}\right)\right|_{m_{3}}
\end{aligned}
$$

The effects of the MSG on MRC are that: (i) The cost of computing the multiplicative inverses is eliminated and (ii) There is a considerable reduction in the number of arithmetic multiplications. For example, given a residue number $(8,3,0)_{\operatorname{RNS}(9|8| 7)}$, the decimal equivalent can be obtained as follows:

Solution. For the moduli set $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}$, we have

$$
\begin{aligned}
& a_{1}=8 \\
& a_{2}=|(3-8)|_{8}=3 \\
& a_{3}=|(4(0-8)-3)|_{7}=0
\end{aligned}
$$

Hence the decimal equivalent is $X=8+3 * 9+0 * 9 * 8=35$.
Finally, in this section, we describe Yassine and Moore's MRC for moduli set of length three. Suppose that a number system is defined by the set of moduli $\left\{m_{1}, m_{2}, m_{3}\right\}$, the associated $\operatorname{MRD}\left(a_{1}, a_{2}, a_{3}\right)$ of a given residue number $\left(x_{1}, x_{2}, x_{3}, \ldots, x_{n}\right)$ must be computed. The algorithm for computing the MRD for moduli set of length three is summarized as follows: $a_{i}=\left|U_{i} V_{i}\right|_{m_{i}}$, where $V_{i}$ are constant predetermined factors given by

$$
\begin{aligned}
V_{1} & \equiv 1 \\
V_{2} & \equiv\left|\left(m_{1}\right)^{-1}\right|_{m_{2}} \\
V_{3} & \equiv\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}}
\end{aligned}
$$

and $U_{i}$ are computed by using the formulas:

$$
U_{1}=x_{1},
$$

$$
\begin{aligned}
& U_{2}=\left|x_{2}-x_{1}\right|_{m_{2}} \\
& U_{3}=\left.\left.\left|x_{3}-x_{1}-W_{2}\right| U_{2} V_{2}\right|_{m_{2}}\right|_{m_{3}}
\end{aligned}
$$

For the moduli sets $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}$ and $\{2 n+1,2 n, 2 n-1\}$, substituting the values for the multiplicative inverses presented in Table I into the above equation gives

$$
\begin{aligned}
V_{1} & \equiv 1 \\
V_{2} & \equiv 1 \\
V_{3} & \equiv \frac{m_{2}}{2}
\end{aligned}
$$

and $U_{i}$ are computed by using the formulas:

$$
\begin{aligned}
& U_{1}=x_{1} \\
& U_{2}=\left|x_{2}-x_{1}\right|_{m_{2}} \\
& U_{3}=\left.\left.\left|x_{3}-x_{1}-W_{2}\right| U_{2}\right|_{m_{2}}\right|_{m_{3}}
\end{aligned}
$$

The moduli set $\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}$ gives the same result as given above except for $V_{3}$ which is computed as

$$
V_{3} \equiv \frac{m_{1}}{4}
$$

The effects of the MSG on Yassine and Moore's MRC and the traditional MRC are exactly the same giving equal number of arithmetic operations for the computations of MRD. Again, suppose that we have a residue number $(8,3,0)_{\operatorname{RNS}(9|8| 7)}$, the decimal equivalent can be obtained as follows:

$$
\begin{aligned}
V_{1} & \equiv 1 \\
V_{2} & \equiv 1 \\
V_{3} & \equiv 4
\end{aligned}
$$

and $U_{i}$ are computed by using the formulas:

$$
U_{1}=8
$$

$$
\begin{aligned}
& U_{2}=|3-8|_{8}=3 \\
& U_{3}=\left.\left.|0-8-9| 3\right|_{8}\right|_{7}=0
\end{aligned}
$$

Hence the decimal equivalent is given by

$$
X=8 * 1+3 * 9+0 * 72=35
$$

Experimentally, we obtain that $\left|m_{1}^{-1}\right|_{m_{3}}=\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}}$ for all values of $n$. Hence $\left|\left(m_{1} m_{2}\right)^{-1}\right|_{m_{3}}$ will not be displayed in the following tables:

Table IV. Multiplicative inverse values for moduli set $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\}$

| $n$ | $\left\|m_{1}^{-1}\right\|_{m_{3}}$ | $\left\|\left(m_{2} m_{3}\right)^{-1}\right\|_{m_{1}}$ | $\left\|\left(m_{1} m_{3}\right)^{-1}\right\|_{m_{2}}$ |
| :---: | :---: | :---: | :---: |
| 2 | 2 | 3 | 3 |
| 3 | 4 | 5 | 7 |
| 4 | 8 | 9 | 15 |
| 5 | 16 | 17 | 31 |
| 6 | 32 | 33 | 63 |
| 7 | 64 | 65 | 127 |
| 8 | 128 | 129 | 255 |
| 9 | 256 | 257 | 511 |
| 10 | 512 | 513 | 1023 |

Table V. Multiplicative inverse values for moduli set $\{2 n+1,2 n, 2 n-1\}$

| $n$ | $\left\|m_{1}^{-1}\right\|_{m_{3}}$ | $\left\|\left(m_{2} m_{3}\right)^{-1}\right\|_{m_{1}}$ | $\left\|\left(m_{1} m_{3}\right)^{-1}\right\|_{m_{2}}$ |
| :---: | :---: | :---: | :---: |
| 2 | 2 | 3 | 3 |
| 3 | 3 | 4 | 5 |
| 4 | 4 | 5 | 7 |
| 5 | 5 | 6 | 9 |
| 6 | 6 | 7 | 11 |
| 7 | 7 | 8 | 13 |
| 8 | 8 | 9 | 15 |
| 9 | 9 | 10 | 17 |
| 10 |  | 11 | 19 |

## V. Performance Evaluation

We wrote a C++ program which accepts $n$ as input and generates moduli for the three categories of moduli sets under investigation. All possible multiplicative inverses that may be required when building a converter are computed using the program. First, based on the MSG, it was observed that $\left|m_{1}^{-1}\right|_{m_{2}}=\left|m_{2}^{-1}\right|_{m_{3}}=1$ for different values of $n$. Next, based on the results of our experiments, some of which are displayed in Tables IV, V and VI, we deduce that there is a well established relationship between the moduli form, the moduli and the multiplicative inverses. These deductions are used to come up with the moduli representations of the various multiplicative inverses. This is shown in Tables I, II and III.

Table VI. Multiplicative inverse values for moduli set $\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}$

| $n$ | $\left\|m_{1}^{-1}\right\|_{m_{3}}$ | $\left\|\left(m_{2} m_{3}\right)^{-1}\right\|_{m_{1}}$ | $\left\|\left(m_{1} m_{3}\right)^{-1}\right\|_{m_{2}}$ |
| :---: | :---: | :---: | :---: |
| 3 | 2 | 5 | 5 |
| 4 | 4 | 9 | 13 |
| 5 | 8 | 17 | 29 |
| 6 | 16 | 33 | 61 |
| 7 | 32 | 65 | 125 |
| 8 | 64 | 129 | 253 |
| 9 | 128 | 257 | 509 |
| 10 | 256 | 513 | 1021 |

## VI. Conclusions

In this paper, we investigated moduli selection in RNS which is an important issue in the design of digital systems. We proposed MSG for the three moduli sets $\left\{2^{n}+1,2^{n}, 2^{n}-1\right\},\{2 n+1,2 n, 2 n-1\}$ and $\left\{2^{n}, 2^{n}-1,2^{n-1}-1\right\}$. We wrote a C++ program which accepts $n$ as input and generates moduli for these three moduli sets. All possible multiplicative inverses that may be required when building a converter are computed using the program. Based on this experiment, we deduced that there is a well established relationship between the moduli form, the moduli and
the multiplicative inverses. These deductions were used to come up with the moduli representations of the various multiplicative inverses. Hence, using the MSG, the cost of computing the multiplicative inverses is eliminated. It was observed that some of these multiplicative inverses will always be unity so their inclusion in the computation is no more necessary. Consequently, the usage of these guidelines in building an RNS to decimal converter results in a considerable reduction in the number of arithmetic operations required during the conversion process. These results provide the possibility of a wide range of applications of RNS in DSP.

## References

[1] W. K. Jenkins and B. J. Leon, The use of residue number systems in the design of finite impulse response digital filters, IEEE Trans. Circuits Syst. 24 (1977), 191-200.
[2] Mi Lu, Arithmetic and Logic in Computer Systems, John Wiley \& Sons, New Jersey, 2004.
[3] M. A. Soderstrand, W. K. Jenkins, G. A. Jullien and F. J. Taylor, Residue Number Arithmetic: Modern Application in Digital Signal Processing, IEEE Press, 1986.
[4] N. Szabo and R. Tanaka, Residue Arithmetic and its Application to Computer Technology, MC-Graw-Hill, New York, 1967.
[5] Y. Wang, New Chinese Remainder Theorems, Proc. Asilomar Conference, USA, 1998, pp. 165-171.
[6] W. Wang, M. N. S. Swamy and M. O. Ahmad, Moduli selection in RNS for efficient VLSI implementation, Inter. Symp. Circuits Syst. 4 (2003), 512-515.
[7] H. M. Yassine and W. R. Moore, Improved mixed-radix conversion for residue number system architectures, Proc. IEEE pt-G 138(1) (1991), 120-124.

