$\{2^n + 1, 2^{n+k}, 2^n - 1\}$: A New RNS Moduli Set Extension

Ricardo Chaves
IST/INESC-ID
R. Alves Redol,9, 1000-029 Lisboa, Portugal
rjfc@sips.inesc-id.pt

Leonel Sousa
IST/INESC-ID
R. Alves Redol,9, 1000-029 Lisboa, Portugal
las@inesc-id.pt

Abstract

The increasing usage of Residual Number System (RNS) in signal processing applications demands the development of new and more adaptable RNS moduli sets and arithmetic units. This paper presents a new adaptable moduli set extension for the traditional moduli set $\{2^n + 1, 2^n, 2^n - 1\}$*. As it will be shown, this new moduli set extension (*$\{2^n + 1, 2^{n+k}, 2^n - 1\}$*) allows the balancing of the binary channel (*2^{n+k}*) in relation to the other two channels. Moreover, it does not require the development of new addition and multiplication units, since it is possible to reuse the already developed and well studied units for these moduli operations.*

1 The Traditional Moduli Set and the New Proposed Extension

One of the drawbacks of the RNS usage is the need to use non-binary arithmetic units that *still* have inferior performances compared to the binary arithmetic architectures [1, 2]. In order to compensate for this difference in the data path of the non binary channels, the binary channel of the traditional moduli set $\{2^n + 1, 2^n, 2^n - 1\}$ can be overloaded, resulting in the new moduli set extension $\{2^n + 1, 2^{n+k}, 2^n - 1\}$ with $0 \leq k \leq n$.

While with $k = 0$, one obtains in the traditional moduli set, with $k = n$ one obtains the maximum length for this new moduli set extension, resulting in the $\{2^n+1, 2^{2n}, 2^n - 1\}$ moduli set, for which the binary channel has the *same* amount of bits as the others two channels combined. For this particular case ($k = n$) the arithmetic units length becomes better adapted to the existing fast calculation architectures. Since Digital Signal Processor (DSP) has a dynamic range that is a power of two (2^m), the dimension of n for the traditional moduli set, in order to maintain an approximate dynamic range, has to be $n = \lceil \frac{\log_2(2^m)}{3} \rceil = \lceil \frac{m}{3} \rceil$, which in the majority of the situations is neither an integer number nor a power of two. In this new moduli set ($k = n$), the dimension of n is given by $n = \lceil \frac{m}{4} \rceil$. Thus, when m is a power of two the division by another power of two ($4 = 2^2$) is still a power of two, therefore allowing the dimension of the arithmetic units to be optimized [3, 2, 4].

For a given RNS moduli set to be valid, it is required for all the elements of the base to be co-prime [5, 6]. Consequently the new 2^{n+k} channel has to be a co-prime number of $2^n - 1$ and $2^n + 1$. In order for two numbers to be relatively prime the greatest common divisor (gcd) for all n and k has to be the unitary value. The gcd for a pair of numbers (a,b), can be calculated by the Euclidian algorithm [7], which can be described by the following recursive equation:

$$\begin{aligned}
&r_1 =< a >_b \ ; \ r2 =< b >_{r_1} \ ; \ r_3 =< r_1 >_{r_2} \ ; \ \ldots \\
&r_k =< r_{k-2} >_{r_{k-1}} \ ; \ r_{k+1} =< r_{k-1} >_{r_k} = 0 \\
&gcd(a, b) = r_k. \qquad (1)
\end{aligned}$$

$\mathbf{gcd(2^{n+k}, 2^n - 1)}$:

$$\begin{aligned}
< 2^{n+k} >_{2^n-1} =& < 2^n \cdot 2^k >_{2^n-1} = \\
&= < 2^k >_{2^n-1} \ ; \qquad (2a)
\end{aligned}$$

In order to conclude this algorithm two cases have to be considered depending on the value of k. In the first case, when $k = n$:

$$< 2^n >_{2^n-1} = < 1 >_{2^n} \ ; \qquad (3a)$$

$$< 2^n >_1 = 0 \qquad (3b)$$

$$gcd(2^{n+n}, 2^n - 1) = 1. \qquad (3c)$$

When $k < n$, algorithm 2 can be concluded as:

$$\begin{aligned}
< 2^n - 1 >_{2^k} =& < 2^{n-k} \cdot 2^k - 1 >_{2^k} = \\
&= < -1 >_{2^k} = < 2^k - 1 >_{2^k} \ ; \qquad (4a)
\end{aligned}$$

$$< 2^k >_{2^k-1} = < 1 >_{2^k-1} \ ; \qquad (4b)$$

$$< 2^k - 1 >_1 = 0 \qquad (4c)$$

$$gcd(2^{n+k}, 2^n - 1) = 1. \qquad (4d)$$

Proceedings of the EUROMICRO Systems on Digital System Design (DSD'04)
0-7695-2203-3/04 $ 20.00 IEEE

IEEE COMPUTER SOCIETY

$\mathbf{gcd(2^{n+k}, 2^n + 1)}$:

$$< 2^{n+k} >_{2^n+1} = < 2^n \cdot 2^k >_{2^n+1} =$$
$$= < -2^k >_{2^n+1} = < 2^n + 1 - 2^k >_{2^n+1} \; ; \quad (5a)$$
$$< 2^n + 1 >_{2^n+1-2^k} = < 2^k >_{2^n+1-2^k} \; ; \quad (5b)$$
$$< 2^n + 1 - 2^k >_{2^k} = < 1 >_{2^k} \; ; \quad (5c)$$
$$< 2^k >_1 = 0 \quad (5d)$$
$$gcd(2^{n+k}, 2^n - 1) = 1. \quad (5e)$$

From these results it can be concluded that the new moduli set extension is a valid RNS moduli set, since the pair $2^n + 1$ and $2^n - 1$ from the traditional RNS moduli set is also coprime [8].

In order for a number to be uniquely represented in a given RNS moduli set, the value to be converted (X) has to be within the dynamic range of the moduli set used. With the use of this new moduli set extension, a slight decrease on the dynamic range occurs for the same number of total bits (m). While the dynamic range ($DR = [0, M - 1]$) obtained through the product of the several channels for the traditional moduli set ($\{m_3, m_2, m_1\} = \{2^n + 1, 2^n, 2^n - 1\}$) is:

$$M = \prod_{i=1}^{3} m_i = (2^n + 1) \times (2^n) \times (2^n - 1) = 2^{3n} - 2^n, \quad (6)$$

the minimum dynamic range obtained by the new moduli set extension ($\{2^n + 1, 2^{2n}, 2^n - 1\}$) is given by:

$$M' = (2^n + 1) \times (2^{2n}) \times (2^n - 1) = 2^{4n} - 2^{2n}, \quad (7)$$

thus for the same identical dynamic range results the difference :

$$\begin{aligned} \triangle &= M - M' = 2^{3n} - 2^n - 2^{4n'} + 2^{2n'} \\ &= 2^{3(\frac{m}{3})} - 2^{(\frac{m}{3})} - 2^{4(\frac{m}{4})} + 2^{2(\frac{m}{4})} \\ &= -2^{(\frac{m}{3})} + 2^{(\frac{m}{2})} > 0. \end{aligned} \quad (8)$$

Given that the channels of this new moduli set extension are essentially the same as the ones in the traditional moduli set, only differing in the length, all arithmetic units characteristic to each channel can be reused, only requiring the development of new conversion units.

2 Conversion

Since most (if not all) the values that need to be processed are represented in binary (or equivalent), it is necessary to convert them to an RNS representation, thus binary to RNS conversion units and RNS to binary are demanded in this type of systems. The proposed fast converters are based on memoryless architectures [6, 8].

2.1 Binary to RNS Conversion

An integer X in the range $[0, M - 1]$, represented in 2^n-ary notation as:

$$X = \sum_{i=0}^{4n-1} X_i 2^i = N_3 2^{3n} + N_2 2^{2n} + N_1 2^n + N_0, \quad (9)$$

where

$$N_3 = \{X_{3n+k-1} \cdots X_{3n}\} \text{ for } 1 \leq k \leq n, \quad (10)$$

and

$$N_3 = 0 \text{ for } k = 0, \quad (11)$$

can be uniquely represented in RNS by the set $\{x_1, x_2, x_3\}$ for the moduli set $\{m_1, m_2, m_3\}$ where $M = m_1 \times m_2 \times m_3$. Three converters are required in order to obtain the RNS representation of the integer, one for each base element.

$\mathbf{2^{n+k}}$ Channel

The simplest one is the converter for the m_2 channel. The value x_2 can be obtained by the remainder of the division of X by 2^{n+k}, which can by accomplished by truncating the value X, since:

$$x_2 = \langle X \rangle_{2^{n+k}} = X_{n+k-1} \cdots X_0. \quad (12)$$

$\mathbf{2^n - 1}$ Channel

For the $2^n - 1$ and $2^n + 1$ channels the calculation of the corresponding residues is more complex, since the final result of the conversion depends on the value of all the X bits. Instead of using a division operation to calculate the $2^n - 1$ residue, which is a complex operation and expensive both in terms of area and speed, this calculation can be performed as a sequence of additions, as described below:

$$\begin{aligned} x_1 &= \langle X \rangle_{2^n-1} \\ &= \left\langle N_3 2^{3n} + N_2 2^{2n} + N_1 2^n + N_0 \right\rangle_{2^n-1}. \end{aligned} \quad (13)$$

By taking the equation:

$$< 2^n >_{2^n-1} = < 1 >_{2^n-1}, \quad (14)$$

equation (13) can be rewritten as:

$$x_1 = < N_3 + N_2 + N_1 + N_0 >_{m_1} . \quad (15)$$

Thus the conversion of X to moduli $2^n - 1$ can be performed simply by adding modulo $2^n - 1$ the N_i components of X.

2^n+1 Channel

In an identical manner, the 2^n+1 residue can be calculated as:

$$\begin{aligned} x_3 &= \langle X \rangle_{2^n+1} && (16)\\ &= \left\langle N_3 2^{3n} + N_2 2^{2n} + N_1 2^n + N_0 \right\rangle_{2^n+1} . \end{aligned}$$

Since:

$$< 2^n >_{2^n+1} = < -1 >_{2^n+1}, \qquad (17)$$

equation (16) can be simplified to:

$$x_3 \;=\; < -N_3 + N_2 - N_1 + N_0 >_{m_3} . \qquad (18)$$

In order to calculate equation (18), additions and subtractions are required. However this equation can be further simplified in order to use only additions, since a subtraction modulo 2^n+1 can be expressed by:

$$\begin{aligned} < -N_i >_{2^n+1} &= < (2^n+1) - N_i >_{2^n+1}\\ &= < (2^n - 1 - N_i) + 2 >_{2^n+1}\\ &= < \overline{N_i} + 2 >_{2^n+1} . && (19) \end{aligned}$$

Thus equation (18) can be written as a series of modulo 2^n+1 (=m_3) additions:

$$\begin{aligned} x_3 &= < \overline{N_3} + 2 + N_2 + \overline{N_1} + 2 + N_0 >_{2^n+1}\\ &= < 2 + < 1 + \overline{N_3} + < 1 + N_2 + \overline{N_1}\\ &\quad + N_0 >_{m_3} >_{m_3} >_{m_3} && (20) \end{aligned}$$

2.2 RNS to Binary Conversion

The RNS to binary conversion is performed based on the Chinese Remainder Theorem (CRT) [8]:

$$X = \left\langle \sum_{i=1}^{3} \hat{m}_i \left\langle \frac{x_i}{\hat{m}_i} \right\rangle m_i \right\rangle_M , \qquad (21)$$

where $\hat{m}_i = M/m_i$ and $\left\langle \frac{1}{\hat{m}_i} \right\rangle_{m_i}$ is the multiplicative inverse of $\hat{m}_i$. Equation (21) can be written as [6]:

$$X \;+ MA(X) = \sum_{i=1}^{3} \hat{m}_i \left\langle \frac{1}{\hat{m}_i} \right\rangle_{m_i} x_i . \qquad (22)$$

For the proposed moduli set, it can be proved that the multiplicative inverse of $\hat{m}_i$ ($i = 1, 2, 3$) is [9]:

$$\left\langle \frac{1}{\hat{m}_1} \right\rangle_{m_1} = 2^{n-1} \qquad (23a)$$

$$\left\langle \frac{1}{\hat{m}_2} \right\rangle_{m_2} = 2^{n+k} - 1 \qquad (23b)$$

$$\left\langle \frac{1}{\hat{m}_3} \right\rangle_{m_3} = 2^{n-1} . \qquad (23c)$$

By replacing these values in equation (22):

$$\begin{aligned} X \;+ MA(X) = \quad & 2^{n+k}(2^n-1)(2^{n-1})x_3\\ & + (2^{2n}-1)(2^{n+k}-1)x_2\\ & + 2^{n+k}(2^n+1)(2^{n-1})x_1 && (24) \end{aligned}$$

and dividing X by 2^{n+k}, it is possible to obtain:

$$X = 2^{n+k} \left\lfloor \frac{X}{2^{n+k}} \right\rfloor + x_2 , \qquad (25)$$

where:

$$\begin{aligned} \left\lfloor \frac{X}{2^{n+k}} \right\rfloor = \Bigg\langle & \underbrace{\left\langle (2^{2n-1} + 2^{n-1})x_1 \right\rangle_{2^{2n}-1}}_{A}\\ & - \; 2^n x_3 \; + \underbrace{\left\langle (2^{2n} - 2)x_2 \right\rangle_{2^{2n}-1}}_{B}\\ & + \underbrace{\left\langle (2^{2n-1} + 2^{n-1})x_3 \right\rangle_{2^{2n}-1}}_{C} \Bigg\rangle_{2^{2n}-1} . && (26) \end{aligned}$$

Finally, with the simplification of each partial expression A, B, C, $-2^n x_3$:

$$\begin{aligned} A &= x_{1,0}x_{1,n-1}\ldots x_{1,1}x_{1,0}x_{1,n-1}\ldots x_{1,1}\\ B &= \overline{x}_{2,n+k-1}\ldots\overline{x}_{2,0}1\ldots 1\\ C &= x_{3,x}x_{3,n-1}\ldots x_{3,1}x_{3,x}x_{3,n-1}\ldots x_{3,1}\\ -2^n x_3 &= -r_3 = \overline{x}_{3,n-1}\ldots\overline{x}_{3,0}1\ldots 1\overline{x}_{3,n} && (27) \end{aligned}$$

the upper 2^{n+k} bits of X can therefore be calculated with two $3-2$ compressors and one full adder modulo $2^{2n}-1$:

$$\left\lfloor \frac{X}{2^{n+k}} \right\rfloor = \langle C + B + A + (-2^n x_3) \rangle_{2^{2n}-1} . \qquad (28)$$

Note that the mathematical complexity of this converter is exactly the same as the converter for the traditional moduli set [8, 10], differing only in the partial expression A.

3 Hardware Implementation

In this section, the equations obtained in the previous section are used to design the hardware structures for the conversion units.

3.1 Addition and Multiplication Units

As mentioned before, this new moduli set extension has the characteristic of not requiring the development of new multiplication and addition units, since the elements of this moduli set are individually the same as the ones in the traditional moduli set $\{2^n+1, 2^n, 2^n-1\}$ [2, 4].

3.2 Binary to RNS Converters

Since the conversion from binary to the modulo 2^{n+k} is accomplished simply by truncating the value of X, the conversion unit for this channel is nothing more than a feed through network for the first $n+k$ bits of X.

The conversion for the 2^n-1 channel is performed with the addition of 4 bit groups formed from the binary value X, as represented in equation (15). However, in order to perform these 4 additions one does not need to use 4 modulo 2^n-1 carry-propagate adders, which would be slow and inefficient. It is possible to group the values to be added as:

$$x_1 = << N_3 + < N_2 + N_1 + N_0 >_{m_1} >_{m_1} >_{m_1} \qquad (29)$$

with the first and second additions ($< N_2 + N_1 + N_0 >_{2^n-1}$ and $< N_3 + S_1 + C_1 >_{2^n-1}$, respectively), being performed by a 3-2 compressor (CSA), without a significant increase in the converters delay nor area. Nevertheless, the third and last modulo 2^n-1 addition ($< S_2 + C_2 >_{2^n-1}$), requires a full modulo 2^n-1 adder. The resulting architecture for this conversion is depicted in figure 1.

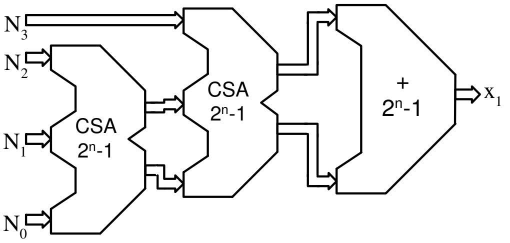


Figure 1. Modulo (2^n-1) Binary to RNS converter

For modulo 2^n+1 one can use two different types of representation: *i)* the standard representation, where the value of the 2^n+1 channel is represented directly as the remainder of the division of X by 2^n+1; *ii)* an alternative representation, designated by diminished-1, in which the remainder of the division is represented as $x_3 = < X - 1 >_{2^n+1}$. In this last representation, the obtained value is always the correct value minus one. Although this representation might seem more complicated and unnecessary, it allows the design of more efficient modulo 2^n+1 multipliers and adders [2, 11, 4].

Diminished-1 representation

As explained above, the diminished-1 representation uses the modulo of $X-1$ instead of X. Computing the value x_3 for this representation with an identical approach as in equation (20), results in the expression:

$$\begin{aligned} x_3 &= < X - 1 >_{2^n+1} \qquad (30) \\ &= < \overline{N_3} + 2 + N_2 + \overline{N_1} + 1 + N_0 >_{2^n+1} . \end{aligned}$$

Like in the 2^n-1 modulo conversion, the additions can be grouped and partially added by a 3-2 compressor. However, the fastest 2^n+1 modulo compressor not only adds the input values, but it also adds one extra unit [6], which happens to be useful in the addition of the extra constant values in equation (31). Grouping the values and the constants in an appropriate structure for the hardware implementation:

$$\begin{aligned} x_3 &= < 1 + < 1 + \overline{N_3} + < 1 + N_2 \qquad (31) \\ &\quad + \overline{N_1} + N_0 >_{2^n+1} >_{2^n+1} >_{2^n+1} \end{aligned}$$

results in the hardware structure presented in figure 2.

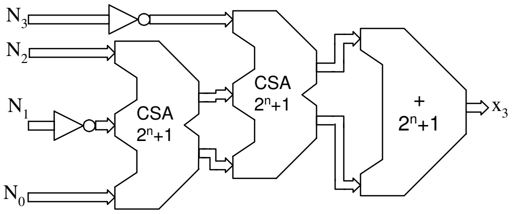


Figure 2. Modulo (2^n+1) Binary to d-RNS converter for the diminished-1 representation

With the diminished-1 representation, a very efficient binary to diminished-1 RNS converter is proposed for the $\{2^n+1, 2^{n+k}, 2^n-1\}$ moduli set. Only two CSA and one full modulo 2^n+1 adder is required for the modulo 2^n+1 converter.

Standard representation

In the standard representation, the value x_3 represents the 2^n+1 residue of the value X, instead of the value $X-1$, which in fact causes some difficulties in the implementation of the conversion unit, due to the constant value that has to be added (4 units in total).

Using the characteristic of the 2^n+1 modulo fast addition (that adds one extra unit per addition), the value x_3 can be computed as:

$$\begin{aligned} x_3 &= < 1 + < 1 + < 1 + \overline{N_3} + < 1 + N_2 \qquad (32) \\ &\quad + \overline{N_1} + N_0 >_{2^n+1} >_{2^n+1} >_{2^n+1} >_{2^n+1} . \end{aligned}$$

Note that the last CSA is only used to add one unit to $S_2 + C_2 + 0$, resulting in a simplified 2-2 compressor.

The binary to modulo 2^n+1 converter is depicted in figure 3, resulting in a converter very similar to the one depicted in figure 2, although a bit less efficient.

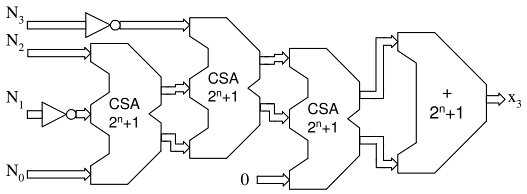


Figure 3. Modulo $(2^n + 1)$ Binary to RNS converter

3.3 RNS to Binary Converter

As observed in equation (28), the conversion from RNS to binary using the new extended moduli set can be performed by adding modulo $2^{2n} - 1$ the four partial expressions in equation (27), that were directly obtained from the 3 RNS channels. In order to optimize this conversion only one full adder (modulo $2^{2n} - 1$) is used in the final part of the computation. The other additions are executed by two modulo $2^{2n} - 1$ CSA, as depicted in figure 4, resulting in a very efficient RNS decoder with the exact same structure as the RNS decoder for the traditional moduli set [8].

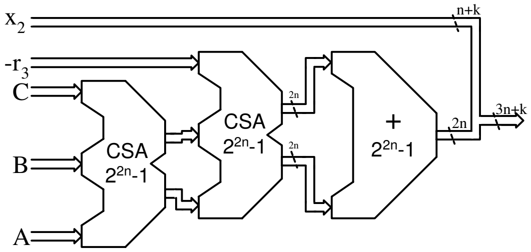


Figure 4. RNS decoder for the standard representation

To decode a RNS value encoded with the diminished one representation, the value from the $2^n + 1$ channel (x_3) has to be incremented by one, which is the same as adding $2^{2n-1} - 2^{n-1}$ (see equation (26)). This extra value can be added by an additional CSA, with the particularity that one of the inputs is a constant value, thus, only a HA is added to the critical path of the decoder (see figure 5).

4 Experimental Results

In order to evaluate the performance of the proposed moduli set extension, all the units – converters, adders and multipliers – were compared, using efficient architectures [2, 4] described in VHDL and implemented in the

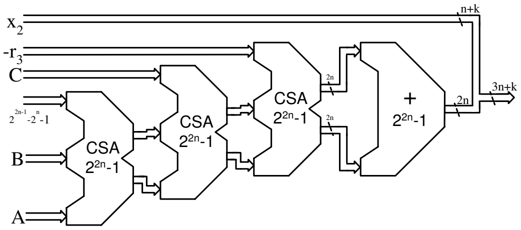


Figure 5. RNS decoder for the diminished representation

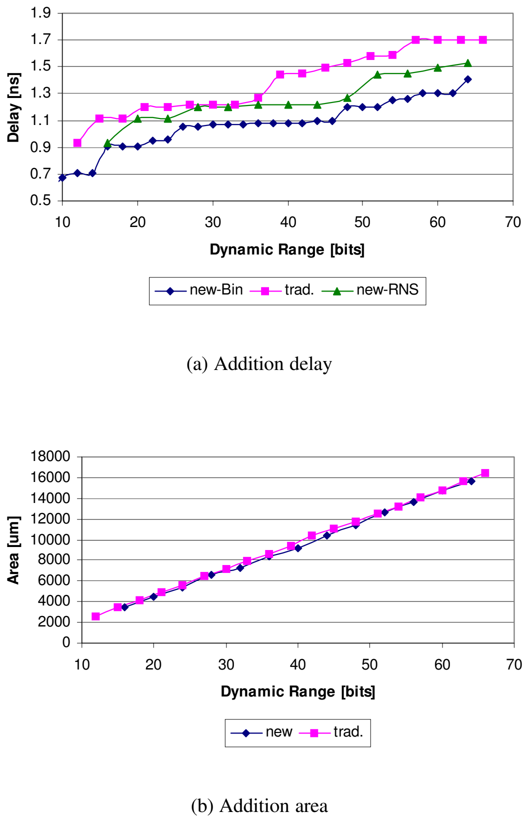
$0.25\mu m$ Standard Cell technology from UMC (1P5M). The results presented are for $k = n$.

The delay and area-occupation values for the adders for both moduli sets are presented in figure 6. The implementation of these adders was optimized for time. These graphics confirm that the usage of the new moduli set extension always results in faster adders. However the gain is widely dependent on the number of bits used. It is possible to obtain an improvement up to 36% (for a dynamic range of m=48 bits); while for m=32 bits the improvement is only of 1.6%. For common dynamic ranges (m=16, 64), the gain is usually above 10%. With the new moduli set extension the addition is performed faster without increasing the global required circuit area (figure 6(b)).

The experimental results, depicted in figure 7, show a significant improvement in the multiplication time, in relation to the traditional moduli set. The most relevant value occurs for a dynamic range of $m = 16$ bits, improving the multiplication time by 23%, with a reduction in the circuit area of 37%. This reduction on the circuit area is local, and mostly caused by the straightforwardness of the four bits $2^n \pm 1$ moduli multiplies. With the increase of bits in the dynamic range, the occupied area in the new moduli set extension increases. Nevertheless, the increase in the circuit area is less than 10% for an average gain in the multipliers acceleration above the 10%.

As expected, the Binary-to-RNS conversion unit for the new moduli set extension is slower than the conversion unit for the traditional moduli set (see figure 8(a)), some times up to 20% slower (9% in average). Even so, the conversion unit for the new moduli set is a relatively fast unit and with approximately the same conversion time as the RNS decoder. Moreover, this conversion is only performed once, since the subsequent operations are additions and multiplications.

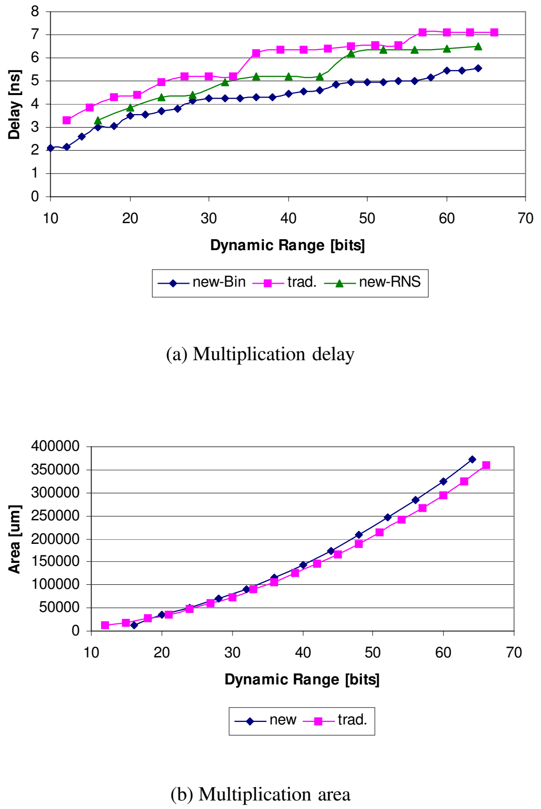
On the other hand, the RNS decoder has an average improvement of about 10%, both in area (figure 9(b)) and in the propagation delay (figure 8(b)). As previously explained, this is due to the fact that more LSBs of the result belong to the 2^{n+k} channel, being directly obtained from



(a) Addition delay

(b) Addition area

Figure 6. ASIC delay for addition.



(a) Multiplication delay

(b) Multiplication area

Figure 7. ASIC delay for multiplication.

the RNS value.

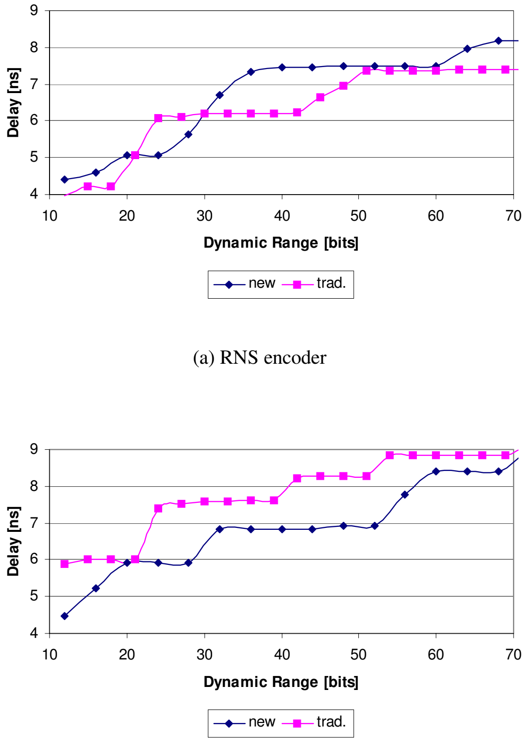
As depicted in figure 10, the RNS encoder, using the diminished-1 representation, is more efficient in the new moduli set, being in average faster than the RNS encoder for the traditional moduli set, and without an area increase. In fact, unlike the traditional moduli set, the RNS encoder for the new moduli set is faster for the diminished one representation than for the standard representation.

Since the $2^n + 1$ adder in the diminished-1 representation is simpler and faster, the gains in the new moduli set extension are not as significant. From the graphic depicted in figure 11, it is noticeable that the new moduli set, with a $k = n$, is 4% slower (for $m = 32$) than the traditional moduli set. However the value k does not have to be equal to n, thus by using a value k equal to 2, both moduli sets will have the same propagation delay. On average, the addition in the new moduli set extension is faster; nevertheless this speedup can be improved by varying the value k. For example the addition in the new moduli set extension, for a dynamic range of $m = 44$ bits, is 15% faster using $k = n = 11$, while using $k = 8$ for the same dynamic range ($m = 44$ with $n = 12$) the gain is 17% (2% better).

5 Conclusions

In this paper a new RNS moduli set extension $\{2^n + 1, 2^{n+k}, 2^n - 1\}$ is proposed. By increasing the number of bits in the binary channel (2^{n+k}) a better balance between this channel and the $2^n \pm 1$ channels is achieved, thus resulting in a faster and more adaptable moduli set (by varying the k number of bits), with an identical circuit area. Regarding other extensions, another advantage of this new moduli set extension is that it uses the same arithmetic units (adders and multipliers), thus allowing the improvements accomplished for the traditional moduli set to be directly used in this new set.

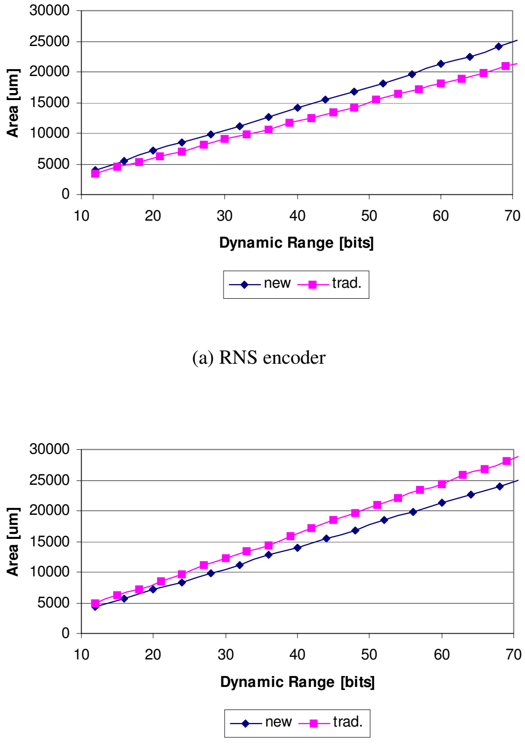
From the experimental results, it can be concluded that with this new moduli set it is possible to obtain a reduction in the delay up to 36%, being in average above 10% for both the addition and multiplication.



(a) RNS encoder

(b) RNS decoder

Figure 8. ASIC delay in the RNS conversion.



(a) RNS encoder

(b) RNS decoder

Figure 9. ASIC area for the RNS conversion.

References

[1] P. Mohan, *Residue Number Systems: Algorithms and Architectures, volume 677 of Engineering and Computer Science*. Kluwer Academic, 2002.

[2] R. Zimmermann, *Binary adder architectures for cell-based VLSI and synthesis*. PhD thesis, Swiss Federal Institute of Technology, 1997.

[3] J. Hennessy and D. Patterson, *Computer Architecture : a Quantitative Approach*. Morgan Kaufmann Publishers, 3 ed., 2003.

[4] R. Chaves and L. Sousa, “Improved $2^n + 1$ multipliers,” *Technical Report RT/14/2003, INESC-ID*, July 2003.

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

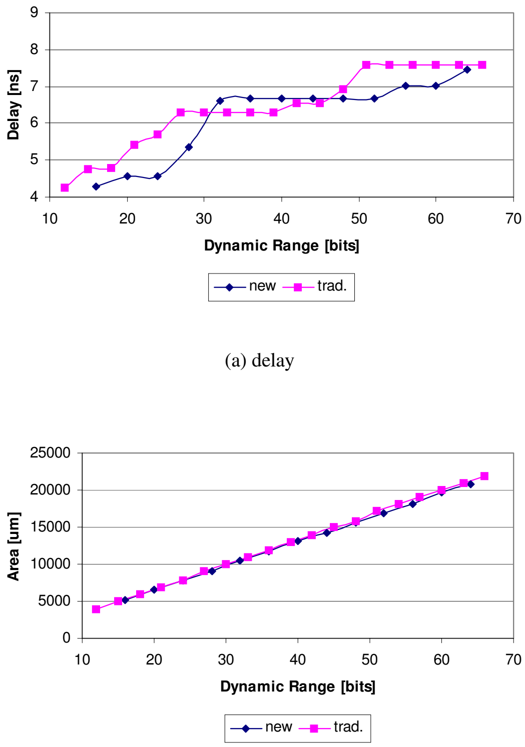
[6] A.S.Ashur, M.K.Ibrahim, and A. Aggoun, “Novel RNS structures for the moduli set $(2^n - 1, 2^n, 2^n + 1)$ and their application to digital filter implementation,” *Signal Processing*, vol. 46, 1995.

[7] D. E. Knuth, *The Art of Computer Programming, vol. 2: Seminumerical Algorithms*. Addison-Wesley, 2 ed., 1996.

[8] S. Andraos and H. Ahmad, “A new efficient memoryless residue to binary converter,” *IEEE Transactions on Circuits and Systems*, vol. 35, November 1988.

[9] R. Chaves, “RDSP: A digital signal processor with suport for residue arithmetic,” Master’s thesis, Instituto Superior Técnico, Lisboa, 2003. written in Portuguese.

[10] S. J. Piestrak, “A high-speed realization of a residue to binary number system converter,” *IEEE Transactions on Circuits and Systems –II: Analog and Digital Signal Processing*, vol. 42, October 1995.



(a) delay

(b) area

Figure 10. ASIC results for the diminished-1 RNS encoder.

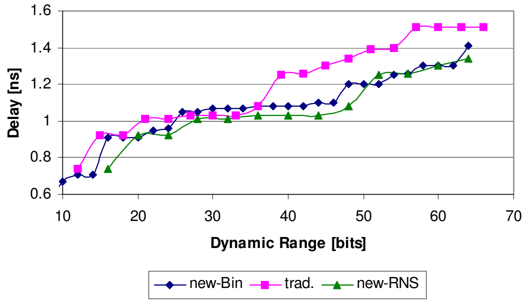


Figure 11. ASIC results for the diminished-1 adder.

[11] Y. Ma, "A simplified architecture for modulo $(2^n + 1)$ multiplication," *IEEE Transactions on Computers*, vol. 47, March 1998.