Characterization of RNS Multiply-Add Units for Power Efficient DSP

Gian Carlo Cardarilli, Alberto Nannarelli$^{(1)}$, Massimo Petricca and Marco Re
Department of Electronic Engineering, University of Rome “Tor Vergata”, Rome, Italy
$^{(1)}$ DTU Compute, Technical University, Kongens Lyngby, Denmark

Abstract—**In the past decades, the Residue Number System (RNS) has been adopted in DSP as an alternative to the traditional two's complement number system (TCS) because of the savings in area, higher speed and reduced power dissipation. In this work, we perform a comprehensive Design Space Exploration (DSE) for a fused multiply-add unit by taking into account four metrics: area, delay, power consumption, and switching activity. The results of the DSE are verified against the TCS and RNS implementation of parallel FIR filters of different characteristics. In both the DSE and the filter implementation, we consider two design corners: maximum speed and minimum area.**

The experimental results demonstrate that for high data rates and high order filters, the RNS implementation is more power efficient than the TCS because of the reduced switching activity and the larger amount of low-power cells placed in the unit.

I. Introduction

The Residue Number System (RNS) as an alternative to binary representation in digital systems has been studied extensively in the last decades [1], [2]. The attractive feature of the RNS is that for operations such as addition and multiplication, a large dynamic range (bit-width) can be divided into sub-ranges of reduced bit-width by splitting the datapath in independent parallel channels where the operations can be executed faster (shorter carry propagation) [1].

The main drawback is that converters are required to interface a RNS datapath to the conventional systems based on binary representation.

Over the years, several applications in Digital Signal Processing (DSP), such as digital filters, have benefit from the RNS implementation, e.g., [3], [4], [5], [6].

In this work, we compare the implementation of digital filters in RNS to the implementation in the conventional two's complement system (TCS). We use state-of-the-art CAD tools and contemporary standard cell libraries to evaluate if the RNS benefits have been reduced or enhanced with respect to previous designs.

Our analysis is based on a Design Space Exploration (DSE) to investigate the performance of the TCS and RNS multiply-add (MADD) units in terms of delay, area and power dissipation for a set of different dynamic ranges.

By the DSE, we identify design corners, for each operator, where the RNS is advantageous over the TCS. Then, we verify and validate the results of the DSE for selected corners by implementing actual digital filters in both RNS and TCS.

II. RNS Background

A Residue Number System (RNS) is defined by a set of P co-prime integers $\{m_1, m_2, \ldots, m_P\}$ which identify the RNS base. Its dynamic range is given by the product

$$M = m_1 \cdot m_2 \cdot \ldots \cdot m_P \quad . \tag{1}$$

Any integer $X \in \{0, 1, 2, \ldots M-1\}$ has a unique RNS representation given by:

$$X \overset{RNS}{\to} (\ \langle X \rangle_{m_1}, \langle X \rangle_{m_2}, \ldots, \langle X \rangle_{m_P}\ ) \tag{2}$$

where $\langle X \rangle_{m_i}$, or X_{m_i}, denotes the operation $X \ mod \ m_i$.

The choice of the moduli set, or RNS base, is very important to obtain an efficient RNS implementation. From (1) we can roughly assume that the average number of bits (bit-width) of each modulus composing the RNS base is

$$d_{ave} = \frac{\sum_{i=1}^{P} \lceil \log_2 m_i \rceil}{P} \tag{3}$$

To cover a given dynamic range D (such as $D \le M$) we can select the RNS base by using two different criteria:

1) RNS base composed of a small number P of moduli. In this case, by (3) the dynamic range of each modulus in the base is medium/large.
2) RNS base composed of a relative large number P of moduli. In this case, the dynamic range of each modulus in the base is medium/small.

The main advantage in using criterion 2) is that we can implement modular multiplication by isomorphism without using an actual multiplier [1].

The input/output converters constitute an overhead independently of the chosen RNS base, but, generally, the complexity of the converters does not have a significant impact for applications in which the number of RNS operations (additions and multiplications) is large.

The Coding Overhead (COH) is defined as the amount of extra bits required in the RNS representation of an integer compared to its TCS representation [7]. A large COH has a significant impact on registers and multiplexers.

Moreover, the coding overhead heavily penalizes the RNS in datapaths requiring the storage of the operands of multiplication. In a $n \times m$-bit TCS multiply unit the required storage for the operands is $n + m$ bits (e.g., $n + m$ flip-flops). In RNS, both operands must be represented in the RNS base requiring $2 \times (n + m + COH)$ bits of storage.

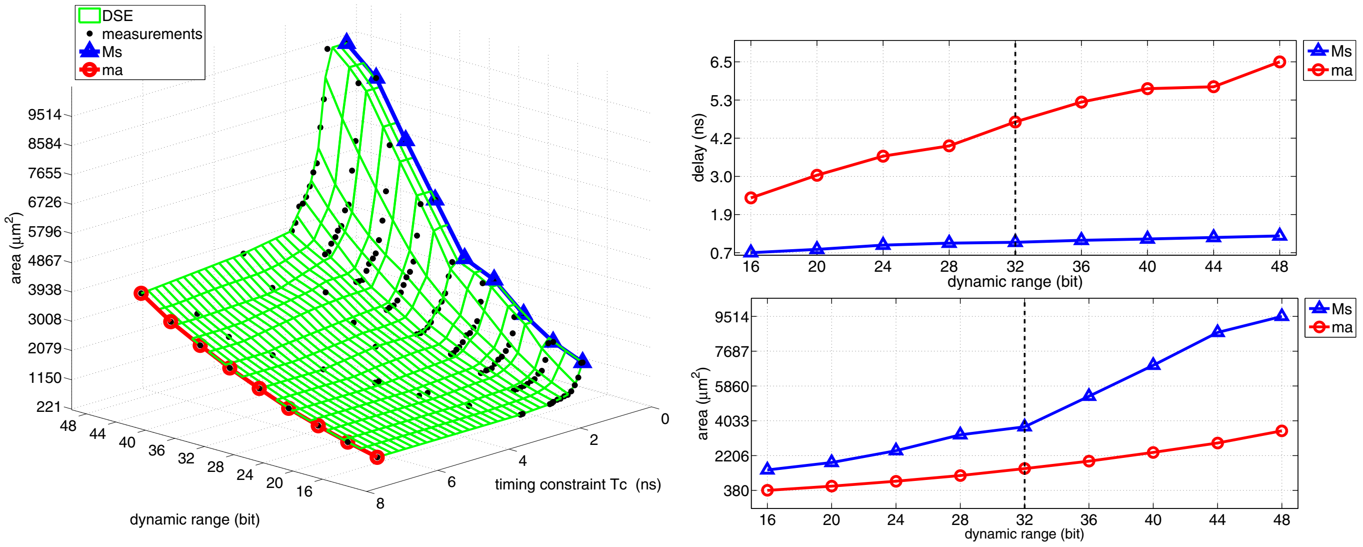

Fig. 1. TCS MADD plots. 3D Area (left). Maximum speed (Ms) and minimum area (ma) sides, for delay (top right) and area (bottom right).

III. Design Space Exploration

The DSE is based on the characterization in a commercial 45 nm library of standard cells of a fused (merged) multiply-add (MADD) unit in both TCS and RNS. The characterization is done on a set of dynamic ranges which are typical of the implementation of digital filers. Namely, 16, 20, 24, 28, 32, 36, 40, 44 and 48 bits.

In the DSE, we consider two design corners:

- *Ms* the maximum speed corner, obtained by synthesizing the functional unit to achieve the maximum speed.
- *ma* the minimum area corner, obtained by synthesizing the unit to achieve the minimum area.

The two corners for the TCS MADD are illustrated in Fig. 1. The 3D plot (area, dynamic range, timing constraint) is projected on the *area–dynamic range* plane in Fig. 1 (bottom right). Similarly, Fig. 1 (top right) is the projection of a 3D delay plot (not shown in the figure) on the *delay–dynamic range* plane.

In the DSE, for each dynamic range, we perform a delay, area and power dissipation characterization for 12 different synthesis timing constraints T_C, ranging from the *ma* to the *Ms* corners.

As discussed in Sec. II, to cover a given dynamic range in RNS several sets of co-prime moduli (RNS base) can be chosen.

Among the several selection criteria, we choose one based on the minimization of a specific cost function. To cover the given dynamic range D, we consider all the possible combinations of co-prime moduli[1] which cover the range and do not exceed D too much (e.g., 2 bits): $2^d = D \le \prod^P m_i = M \le 2^{d+2}$ Then, we choose the set (RNS base) which minimizes the cost function (e.g. area or power dissipation of the RNS operator) at a given timing constraint T_C.

A. TCS and RNS MADDs

In TCS, the MADD $Z = X \cdot Y + W$ is efficiently implemented by adding W (properly aligned) in the partial products reduction tree of the multiplier. We opted for a radix-4 multiplication to reduce the number of partial products.

The RNS MADD architecture is implemented by cascading a isomorphic multiplier and a modular adder, for each modulus of the RNS base.

Because we use prime numbers, we can transform a modular multiplication into a modular addition of the isomorphic indices [1]. The *isomorphism* method is similar to the method of multiplication by logarithms: the two operands are transformed into their indices of the isomorphism (logarithms), the indices (logarithms) are added, and their sum is transformed back (anti-logarithm).

If the modulus is small (its bit-width is 6–8 bits), the forward and reverse isomorphic transformations can be implemented by look-up tables of reasonable size which are then synthesized into multi-level combinational logic.

For the implementation of the isomorphism, we considered the two alternatives of [8] and [9], and selected the one with the lower cost for the specific RNS base.

B. Characterization Results

Fig. 2 shows the plots of the ratio between the TCS and RNS values for DSE of the MADD units. In both the *Ms* and the *ma* corners, the RNS MADD is always faster, smaller and more power efficient than the TCS MADD (ratio > 1.0).

The RNS MADD implementation is 30–60% faster than the corresponding TCS implementation in the *Ms* corner and about two times faster in the *ma* corner. Moreover, the RNS MADDs consume roughly one quarter of the power in the *Ms* corner and about one third of the power in the *ma* corner.

The characterization of the MADD is carried out by applying random test-vectors uniformly distributed in the whole dynamic range of the input signals.

[1] Prime numbers from 3 to 71 and at most one of power-of-two (2^k) from 4 to 256.

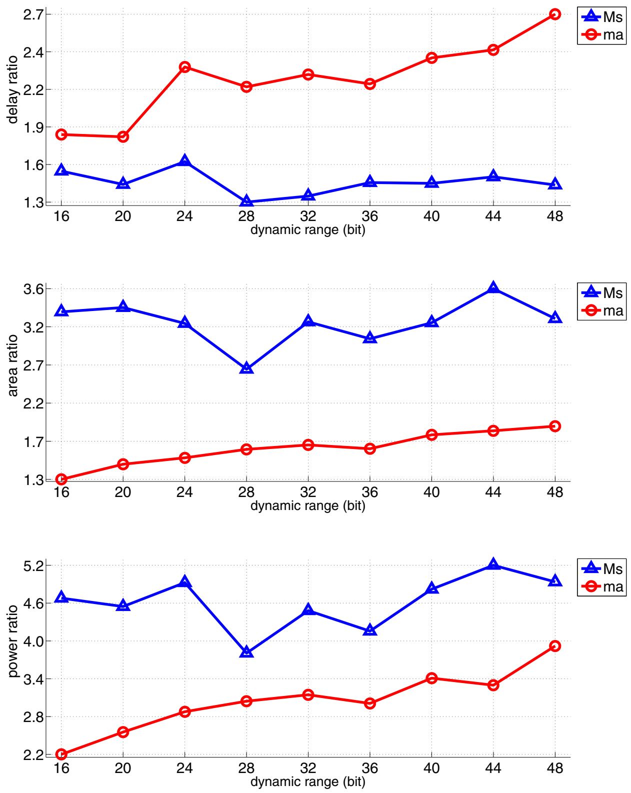

Fig. 2. DSE multiply-add: ratios TCS/RNS in *Ms* and *ma* corners.

However, when a MADD unit is used in FIR filters, one of inputs is connected to one of the coefficients specifying the filter mask. As an example, in linear-phase frequency filters with N coefficients, the coefficients closer to the mid-point $N/2$ have a full dynamic range, while the ones closer to the end-points (i.e., 0 and $N-1$), also called *filter's tails*, have a reduced dynamic range.

To simulate the multiplication by a reduced dynamic range operand, we generate random test-vectors at a progressively reduced dynamic range. The dynamic range reduction characterizing this experiment only affects input Y of the MADD unit ($Z = X \times Y + W$) while at inputs X and W we apply random test-vectors at full dynamic range.

The results of the experiment are shown in Fig. 3. On the x-axis we indicate how many Most-Significant Bits (MSBs) are held constant (to 0 or 1) to reduce the dynamic range of the test-vectors. In the mid-chart point (labeled '0') in Fig. 3 the MADD works at full dynamic range. The dynamic range is reduced moving from the center of the chart to the right for positive values (MSBs held at 0), and to the left for negative values (MSBs held at 1).

Fig. 3 shows that the TCS MADD is sensitive to the variations in the dynamic range of one of the operands, while the RNS MADD is almost insensitive to those variations.

In summary, although the data of Fig. 2 points to a better power efficiency of the RNS over the TCS MADD, the benefits are reduced in actual filters since in the filter's tails the TCS MADDs have a reduced power dissipation due to the smallest dynamic range of one of the operands.

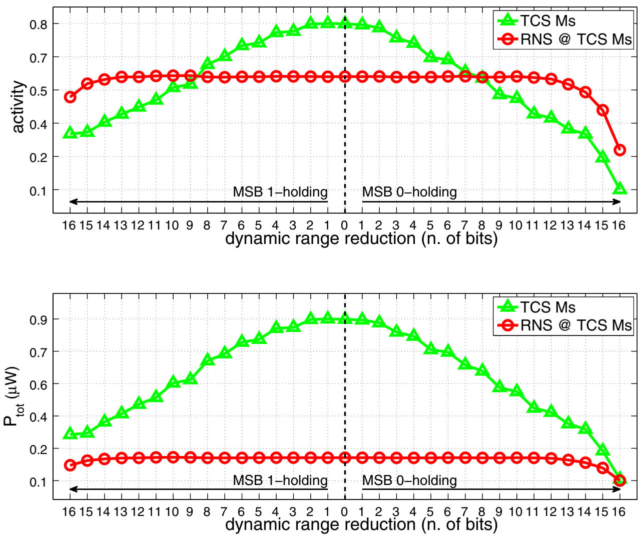

Fig. 3. Switching activity (top) and total power consumption (bottom) in 32-bit MADD when the dynamic range of test-vectors is progressively reduced.

IV. Case Study: FIR Filter

A FIR filter is described by $y(n) = \sum_{k=0}^{N-1} a(k) \cdot x(n-k)$ where N is the order of the filter, $x(n)$ and $y(n)$ are respectively the input and the output sequences, and $a(k)$ are the filter coefficients (filter mask).

The TCS FIR filter is implemented by using the transposed form architecture. The filter is composed of a regular repetition of blocks (multiply-add and register), a so called *filter's tap*.

In RNS, the FIR filter is implemented by P parallel filters in transposed form for each modulus. Each tap is implemented by a modular MADD and the architecture is completed by input and out converters.

Since the coefficients $a(k)$ are loaded in their registers at filter initialization (e.g., serially) and do not change value until a new mask is selected, the registers holding the coefficients are clock-gated to save power (both in TCS and RNS).

For the implementations, we used Synopsys's IC Compiler and the same standard cell library used for the DSE (CMOS 45 nm standard cell). We simulated by Synopsys's VCS the layout extracted netlists to generate the switching activity.

For the RNS filters, implemented by [10], the overhead of the input/output converters is always taken into account.

We implemented the following real coefficient FIR filters:

- **EXP-1**: Low dynamic range (16 bits) and low order filter (16 coefficients)
- **EXP-2**: High dynamic range (48 bits) and high order filter (64 coefficients)

As done for the DSE, we show the performance of the filter implementations for two distinct design corners: maximum speed corner (*max speed, Ms*) and the minimum area corner (*min area, ma*). For the *Ms* corner, both TCS and RNS filters are implemented to operate at a maximum speed achievable by the TCS filter (always slower than RNS maximum speed), while for the *ma* corner, the clock period of the units is not constrained. In all experiments, the average power consumption is estimated at clock frequency 100 MHz.

EXP-1							
max speed	TCS			RNS			$\frac{\text{TCS}}{\text{RNS}}$
Td $[ns]$	1.26			1.26 (*Ms* 0.9)			1.00
area $[mm^2]$	comb. 0.023	seq. 0.003	tot. 0.026	comb. 0.010	seq. 0.004	tot. 0.014	1.86
power $[mW]$	dyn. 2.477	leak. 0.004	tot. 2.481	dyn. 1.460	leak. 0.001	tot. 1.461	1.70
activity (ave.)	0.44			0.47			0.94
min area	TCS			RNS			$\frac{\text{TCS}}{\text{RNS}}$
Td $[ns]$	3.67			2.73			1.34
area $[mm^2]$	comb. 0.008	seq. 0.003	tot. 0.011	comb. 0.007	seq. 0.004	tot. 0.012	0.91
power $[mW]$	dyn. 0.986	leak. 0.001	tot. 0.987	dyn. 1.137	leak. 0.001	tot. 1.138	0.87
activity (ave.)	0.47			0.52			0.90

TABLE I
POST-LAYOUT RESULTS OF EXP-1 (16-TAP FILTER) IN TCS MAX SPEED (TOP) AND MIN AREA (BOTTOM) DESIGN CORNERS.

EXP-2							
max speed	TCS			RNS			$\frac{\text{TCS}}{\text{RNS}}$
Td $[ns]$	2.10			2.10 (*Ms* 1.8)			1.00
area $[mm^2]$	comb. 0.504	seq. 0.028	tot. 0.532	comb. 0.130	seq. 0.044	tot. 0.174	3.06
power $[mW]$	dyn. 69.61	leak. 0.083	tot. 69.69	dyn. 19.74	leak. 0.015	tot. 19.76	3.53
activity (ave.)	0.63			0.55			1.13
min area	TCS			RNS			$\frac{\text{TCS}}{\text{RNS}}$
Td $[ns]$	9.07			7.92			1.15
area $[mm^2]$	comb. 0.244	seq. 0.028	tot. 0.272	comb. 0.112	seq. 0.044	tot. 0.156	1.74
power $[mW]$	dyn. 40.35	leak. 0.018	tot. 40.36	dyn. 18.21	leak. 0.010	tot. 18.22	2.22
activity (ave.)	0.83			0.65			1.27

TABLE II
POST-LAYOUT RESULTS OF EXP-2 (64-TAP FILTER) IN TCS MAX SPEED (TOP) AND MIN AREA (BOTTOM) DESIGN CORNERS.

The results for EXP-1 are reported in Table I. In the table, the rightmost column shows the ratios TCS/RNS for the corresponding metric. At *Ms* corner, the RNS filter, synthesized at the maximum operating speed of the TCS, is almost half the area and consumes 40% less power than the TCS filter. In contrast, at *ma* corner, the TCS filter consumes less power.

Fig. 4 shows the power dissipation in each tap of the filter and confirms that the power advantages of RNS are reduced, or nullified, in the tails of the filter.

Moreover, in the *ma* corner, the power savings in the RNS taps are offset by the overhead of input/output converters.

The results of EXP-2 (64-tap FIR filter) are reported in Table II. In this case, the results for the *Ms* corner are even more advantageous for the RNS implementation, but also in the *ma* corner the RNS filter is advantageous over the TCS. In this latter case, the power savings in the RNS taps are not offset by the input/output converters.

The results of these experiments on real coefficients FIR filters confirms that for high-order and high-speed parallel filters a RNS implementation is preferable over an implementation in TCS, while for low-order filters with relaxed time requirements a TCS implementation is more appropriate.

V. CONCLUSIONS

Our DSE confirmed that for the dynamic ranges typical of DSP applications, the units implemented in RNS are faster than their TCS counterparts, and that for implementations at the same delay constraints, the combinational part of RNS units is less complex (smaller area). For the storage part (registers) the overhead introduced by the RNS is significant in some cases, but it can be almost neutralized from a power dissipation perspective by using clock gating.

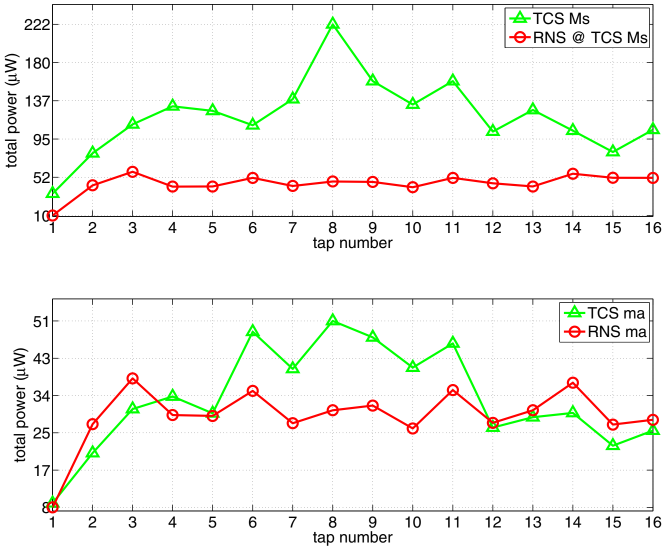

Fig. 4. Power dissipation per tap in EXP-1. Max speed (top) and min area (bottom) design corners.

The DSE also confirmed that the switching activity is significantly reduced in RNS, unless one of the MADD operands has a reduced dynamic range (e.g., tails of the FIR filter).

The case study confirmed that if the system has to operate at high frequency rates RNS offers the best tradeoffs speed/power, while, when the execution rates are not too demanding, implementing the system in RNS does not give significant advantages because of the added complexity (converters and representation overhead).

REFERENCES

[1] N. Szabo and R. Tanaka, *Residue Arithmetic and its Applications to Computer Technology*. McGraw-Hill, 1967.

[2] P. V. Ananda Mohan, *Residue Number Systems: Algorithms and Architectures*. Kluwer Academic Publishers, 2002.

[3] W. Jenkins and B. Leon, "The use of residue number systems in the design of finite impulse response digital filters," *IEEE Transactions on Circuits and Systems*, vol. 24, no. 4, pp. 191–201, 1977.

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

[5] G. C. Cardarilli, A. Del Re, A. Nannarelli, and M. Re, "Residue Number System Reconfigurable Datapath," *Proc. of IEEE Int.l Symposium on Circuits and Systems (ISCAS 2002)*, vol. 2, pp. 756–759, 2002.

[6] R. Conway and J. Nelson, "Improved RNS FIR Filter Architectures," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, no. 1, pp. 26 – 28, Jan. 2004.

[7] G. C. Cardarilli, A. D. Re, A. Nannarelli, and M. Re, "Impact of RNS Coding Overhead on FIR Filters Performance," *Proc. of 41st Asilomar Conf. on Signals, Systems, and Computers*, pp. 1426–1429, Oct. 2007.

[8] D. Radhakrishnan and Y. Yuan, "Novel approaches to the design of vlsi rns multipliers," *IEEE Transaction on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 39, no. 1, pp. 52–57, Jan. 1992.

[9] A. Nannarelli, G. C. Cardarilli, and M. Re, "Power-Delay Tradeoffs in Residue Number System," *Proc. of IEEE Int.l Symposium on Circuits and Systems (ISCAS 2003)*, vol. 5, pp. 413–416, 2003.

[10] A. Del Re, A. Nannarelli, and M. Re, "A tool for automatic generation of RTL-level VHDL description of RNS FIR filters," *Proc. of Design, Automation and Test in Europe Conf. (DATE)*, pp. 686–687, 2004.