

Twenty Years of Research on RNS for DSP: Lessons Learned and Future Perspectives

Pietro Albicocco, Gian Carlo Cardarilli, Alberto Nannarelli⁽¹⁾, and Marco Re
Department of Electronic Engineering, University of Rome “Tor Vergata”, Rome, Italy
⁽¹⁾ DTU Compute, Technical University, Denmark

Abstract—In this paper, we discuss a number of issues emerged from our twenty-year long experience in applying the Residue Number System (RNS) to DSP systems.

In early days, RNS was mainly used to reach the maximum performance in speed. Today, RNS is also used to obtain power-efficient (tradeoffs speed-power) and reliable systems (redundant RNS). Advances in microelectronics and CAD tools play an important role in favoring one technology over another, and a winning choice of the past may become at disadvantage today.

In this paper, we address a number of factors influencing the choice of RNS as the winning solution. From technology platforms (ASIC and FPGA), to issue related to modern design tools, from cost of memory, to cost of wiring, from power dissipation to thermal issues. Moreover, we mention how RNS can be helpful in implementing reliable architectures (fault detection and correction) in future VLSI systems.

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].

In this work, we review the results obtained for RNS implementations of DSPs with emphasis on the impact of today’s integrated circuit technology and state-of-the-art CAD tools in the design of such DSPs. The RNS processors are compared to processors implemented in the conventional two’s complement system (TCS).

In this review, our main intent is to show how the impact of RNS in reducing the power consumption in DSP applications has changed over the years because of advances in technology and CAD tools.

We differentiate the results between the two major platforms used in DSP: ASIC and FPGA, as the use of RNS on these two platforms seems to lead to diverging conclusions.

Moreover, based on the trends seen in these about twenty years of research on RNS, we try to forecast future scenarios for RNS in DSP.

II. RESIDUE NUMBER SYSTEM

A residue number system is based on the definition of a set of P coprime integer numbers that represent a set of moduli $\{m_1, \dots, m_i, \dots, m_P\}$. The representable dynamic range is defined as the product of the moduli:

$$M = \prod_i m_i \quad (1)$$

and represent the number of elements that can be univocally represented in RNS. If we assume to represent only positive integer values the representable range of numbers in the RNS domain is: $0 \leq X \leq M - 1$. Operations, such as addition and multiplication, are computed independently (in parallel) in each modulus m_i path

$$Z = X \text{ op } Y \xrightarrow{RNS} \begin{cases} Z_{m_1} = \langle X_{m_1} \text{ op } Y_{m_1} \rangle_{m_1} \\ Z_{m_2} = \langle X_{m_2} \text{ op } Y_{m_2} \rangle_{m_2} \\ \dots \\ Z_{m_P} = \langle X_{m_P} \text{ op } Y_{m_P} \rangle_{m_P} \end{cases} \quad (2)$$

Consequently, operations on large wordlengths are split into several modular operations executed in parallel and characterized by a reduced wordlength [1].

A. RNS Advantages

The RNS representation offers many advantages in the implementation of DSP systems.

A large dynamic range is split in RNS in slices of smaller dynamic ranges that are characterized by the absence of carry propagation among modular paths. Therefore, the ability of RNS to perform addition and multiplication independently on the different moduli makes it beneficial for many DSP applications, particularly when large word lengths and high throughput are required [4].

Another advantage of the RNS comes from a particular representation of complex numbers developed to limit the numbers of operations needed to execute a complex multiplication and called the Quadratic Residue Number System (QRNS) [6]. It is based on the definition of two distinct terms for each modulo such that complex operations never require cross products between them, but they are kept on being processed distinctly each in their own path.

Being the RNS based on the non-linear operation of modulo, independently on the input distribution, the modulo

operation cuts out any information tied to the bits position: the modulo brings any input distribution back to a uniformly distribution over the whole dynamic range of each modulo. This property is useful to predict the power consumption of a RNS design: the power consumption is completely insensitive to possible correlations and distribution of the input.

B. RNS Disadvantages

Besides the many advantages, the RNS presents also some drawbacks which may cause the RNS to be unappealing in some classes of applications.

The first disadvantage is the overhead introduced by the forward and reverse converters that, based on the complexity of the system to implement, may result too large for the RNS to be advantageous compared to the TCS.

The main limit of the RNS resides in the complexity behind the scaling operations: to reduce the dynamic range in the RNS domain is very onerous because it involves an action performed on the whole set of moduli and its overhead is comparable at least to the overhead of an output converter.

The complexity of scaling in RNS puts a hard limit on the classes of applications that can be efficiently implemented in RNS. This limitation practically excludes the RNS from implementing algorithms that presents some cascade multiplications because of the consequent growth in dynamic range. For these reasons, the use of RNS is practically limited to classes of applications with a limited increment of the dynamic range, such as linear combinations.

Another inefficiency of the RNS is the coding overhead [7]. That is efficiency loss in a RNS system due to the larger number of bits that are needed to represent the whole set of moduli when compared with the number of bits needed to represent the same dynamic range in a TCS system.

III. KEY FACTORS IN IMPLEMENTING DSPS IN RNS

A. Choice of the Moduli Set

The selection of the RNS base is of capital importance to obtain an efficient RNS implementation especially when high order/high dynamic range filters must be implemented. To cover a dynamic range of D bits we can choose two completely different approaches.

- 1) use of small sets of moduli (the dynamic range of each modulus is medium/large). Often in this case *special structure* moduli can be used [4].
- 2) use of medium/large sets of moduli (each modulus has a medium/small dynamic range). In this case, it is generally difficult, or impossible, to use *special structure* moduli, and, usually, the moduli set is chosen by selecting prime moduli to exploit the simplifications in the implementation of the multiplication by the isomorphism technique [8].

For example, for a 24 bit FIR filter by using the two approaches we have:

- 1) the coprime moduli set $2^8 - 1, 2^8, 2^8 + 1$ (slightly under the selected dynamic range) for an average number of 8 bits in the base;

- 2) the coprime moduli set $\{7, 11, 13, 17, 31, 32\}$ for an average number of 4.3 bits in the base.

The use of the second approach means that we have more degrees of freedom in the moduli selection so a better design space exploration can be done. An inefficiency may arise from the differences in the moduli bit-width causing the delays in the different paths be unbalanced.

Another important lesson learned in our research is that the selected RNS base requires extra bits with respect to the chosen dynamic range as a consequence that each modulus in the base is not a power of two.

This effect named *Coding Overhead (OH)* has been introduced in [7]. High OH means that the number of registers to memorize the internal variables and filter coefficients is significantly increased.

This factor is really important when high order parallel FIR filters must be implemented, or worse, when a frame based FIR filtering is required due to very high sample rate of the ADCs (several GSPS on two or more channels). In this case, very specialized Serializer/Deserializer (SER/DES) architectures are used and the internal processing in frame based.

In conclusion, in our RNS systems, we choose a RNS base characterized by a set of prime numbers and a power-of-two for the modulus with the higher dynamic range. The selection of the optimum RNS base must be implemented by minimizing a cost function. The costs are computed by a characterization of the modular units for different timing constraints, and by taking into account at the same time the impact of the OH. All the possible combinations that cover the dynamic range D are evaluated. For a given timing constraint T_C , if the delay of the modular unit is larger than T_C , the modulus is discarded. The final cost is the sum of the areas of the modular units that are faster than T_C . Finally, the RNS base with the minimum cost is selected.

B. Importance of RNS CAD tools

Another important learning from our research experience is that RNS has not been extensively used in the industry because of the less conventional arithmetic of modular operations. For this reason, it is important to have CAD tools for the automatic generation of RNS architectures (e.g., synthesizable VHDL blocks). CAD tools must optimize RNS bases to implement the final architecture hiding the complexity of the RNS to the designer. In our experience, CAD tools are also very important to handle the verification of RNS systems: testbenches, simulation scripts, etc.. Different CAD tools have been implemented such as the tool described in [9].

IV. RNS AND EVOLVING TECHNOLOGY

The RNS has been proposed as a technique useful to realise area efficient high speed digital systems.

Over the years the technological advances caused the birth of new devices, the continuous evolution of the design tools and the mutation of the most critical design figures of merit.

In this section, we investigate the impact of the technological evolutions on the RNS performance.

A. ASIC Platforms

In earlier days, the aim of the RNS was the implementation of very fast and small circuits [1]. However, in recent years, power consumption has emerged as an increasingly important design constraint, and has grown to become today the most important figure of merit in digital design for high performance and portable applications.

Because of the many technological and algorithmic advances, implementation results over the years and across many different tools and technologies have changed the tradeoffs, sometimes significantly.

The algorithmic advances in the implementation platforms affect differently the performances of different number systems. For example, the RNS is expected to be locally bounded and to have shorter datapaths. Although this feature belongs to the RNS by definition, in recent years it is completely hidden by the efforts in equalization of loads and delays during the synthesis and place-and-route processes. As a consequence, the wires are kept as short as possible and the wirelength differences between RNS and TCS implementations are concealed.

Especially timing driven placement algorithms are developed specifically targeting wires on critical paths rather than focusing on minimizing the total wirelength. The effects are evident in large TCS implementations characterized by unbalanced datapaths where the maximum delay can be significantly reduced. On the contrary, the effects on the smaller RNS moduli datapaths are less important, resulting in reduced advantages for the RNS.

Even if some of the benefits of the RNS seems to be reduced, the locality property of the RNS results in larger savings in terms of glitching power consumption: glitches mainly arise from delay mismatch at the input of the gates. Although the timing may be unbalanced among moduli, each modular path has highly equalized datapaths. Moreover, the timing in most of the moduli (the larger part of the system) is relaxed with the exception of the critical one that is however a small circuit compared with the TCS design.

These intrinsic features of the RNS result in the distribution of logic on a fewer number of levels of reduced drive strength cells. All these factors significantly affect the distribution of the delays in the circuit, and, consequently, the number of spurious transitions in the nodes. The locality of the RNS becomes evident in the smaller glitch percentage due to the reduced average delay mismatches at the gates inputs.

A technological technique developed to reduce the static power consumption consists in the utilization of multi-threshold voltage (multi- V_t , or MVT) cells libraries. The most recent algorithms map high- V_t (HVT) cells¹ in a MVT design flow trying to compensate the differential delay of input signals by changing standard- V_t cells in slower HVT cells. These algorithms are able to use the HVT cells in a RNS design to reduce both leakage and glitching power consumption by equalizing the small amount of delays unbalance and by filtering out glitches. On the contrary, in a TCS design the delays distribution is so unbalanced that these effects on glitching power are only marginal.

¹High- V_t cells are slower than standard- V_t cells, but consume significantly less power.

Design	2002, Virtex V600E [mW]	2012, Virtex-5 [mW]
TCS	387.9	2.01
RNS	198.7	3.15
Ratio	1.95	0.64

TABLE I. POWER CONSUMPTION OF TCS AND RNS FILTERS IMPLEMENTED ON FPGAS.

To show to the reader how the technological advances affected the RNS and TCS implementations over the years, we present the differences in the results of the implementation of the same architecture at the distance of a decade.

In 2000, we presented the RNS as an effective technique to lower the power consumption in DSP applications [5]: the RNS implementation of a FIR filter showed to be about half smaller and power hungry then the TCS filter when designed at the same timing constraint.

In 2012, we presented the implementation of a RNS FIR filter based on one of the most recent technology processes [10]. The RNS filter is still advantageous when realised at maximum speed, but the area and power savings are reduced to about a 30%. However, when the filter is realised with a relaxed timing constraint, the RNS is still advantageous in terms of power but not in area (it is larger than the TCS).

B. FPGA Platforms

The FPGA has been a very important platform to investigate RNS implementations from two different points of view:

- Higher speed and reduced logic resources by exploiting the look-up table (LUT) based architecture of the FPGAs.
- Lower power consumption.

The internal structure of the FPGAs, that is LUT based, is very suitable for the implementation of residual arithmetic. Moreover, the unavailability of embedded multipliers in early FPGAs was an important driver for the use of RNS in the implementation of parallel FIR filters.

In 2002, we showed that RNS implementations of FIR filters on FPGA were significantly advantageous [11] obtaining a 2.5 times less power consumption per tap and almost the same area than the TCS implementation when realised at the same timing constraint. The implementation was done on a Xilinx Virtex-E FPGA. The power saving ratios were higher also than those obtained for the standard cell implementation [5] where a power saving of 2 times was obtained per tap.

In recent years, the FPGA architecture has changed towards a hybrid architecture containing specialized full-custom processing elements, such as multipliers, or more complex and flexible processing elements (for example, the Xilinx DSP48 blocks). Consequently, the advantages in implementing isomorphic multipliers in general purpose FPGA resources, such as LUTs, are vanished.

In 2012, we compared the TCS and RNS filter implementations on a Xilinx Virtex-5 FPGA equipped with DSP48 arithmetic units [10]. The results show that the RNS is no more a competitive solution for the device: the distributed implementation of multipliers in LUTs cannot compete with the DSP48 based TCS implementation, which resulted to be more area and power efficient.

The comparison of the results between the implementations of 2002 and 2012 is illustrated (power dissipation only) in Table I.

There are some interesting classes of applications, such as, for example, space applications, in which the technology is naturally delayed, and where power savings and the lowering of the non recurring costs are of fundamental importance.

In the outer space there are needs for devices able to work in an environment characterized by particles and high-energy electromagnetic radiations. A programmable radiation-hardened device is a good solution to reduce the i fixed costs to fabricate an ASIC device.

Radiation-hardened components are based on their non-hardened equivalents and enhanced with extra circuitry to reduce the radiation damages. Because of the higher production costs and the lower market demand, radiation-hardened FPGAs tend not to incorporate the latest technological advances. For example, several families of FPGA space-certified still do not have embedded multipliers, or their number is small.

In this context, for high performance DSP requiring a large number of multipliers, such as a parallel high-order filter, the RNS implementation can still be advantageous [12].

V. RNS: WHAT WE SEE IN PERSPECTIVE

In the following, we point out our vision about RNS future perspectives.

First, we see an increase in parallelism: traditional microprocessors are evolving towards parallelism due to the impossibility to increase the clock rate. Therefore, we need to increase the number of cores to increase the throughput. This trend toward parallelism is also true for arithmetic cores used in general purpose processors: GPU based accelerators or vector processors (e.g., the NEON unit in ARM processors). We see an increase in parallelism also in DSP architectures: for example, to keep up the giga-sample rates of the latest ADCs, the samples are organized in frames requiring DSP blocks with memory to use matrix-structured implementations.

Second, we see the birth of DSP partitioning techniques at number system level in order to map a part of an algorithm to the most efficient number system for its implementation (for example the parts of the algorithm characterized by the absence of feedback and long linear combinations, such as high dynamic range and high order FIR filters, and implementations of transforms).

Third, we see a trend in the direction of fault-tolerant DSP systems. Due to the aggressive technology scaling, devices are so small that can be upset by radiations even at sea level. Because an increasing number of life critical applications (medical, automotive, avionics, etc.) are implemented in those devices, it is very important to design such system in a fault-tolerant and self-correcting manner. The RNS could be an enabling technology in this case due to its inherent parallelism that can be exploited to create the necessary redundancy.

Fourth, there is a general trend toward the implementation of DSP in floating-point to cope with the increased complexity of the DSP applications. In this context, the use of RNS is at disadvantage (scaling is required, etc.).

Fifth, the RNS can help the mitigation of hot-spot problems. A hot-spot is generated by high power consumption concentrated in a limited area of the die causing the temperature to rise. Because the RNS is characterized by a quite uniform distribution of node capacitance and RNS paths produce less glitches than TCS, the temperature profile of the die is expected to be rather flat.

VI. CONCLUSION

In this paper we analysed the advantages and the disadvantages of the RNS implementation on the most common digital platform for signal processing, namely FPGAs and ASICs. We investigated the impact on the RNS implementations of the technological advances over the years.

Although the technological advances and the modern CAD tool favor the implementation of the common case (TCS), the benefits of RNS in terms of area, delay and power consumption are still remarkable for systems implemented on ASICs.

As for FPGA platforms, RNS will still offer advantages in fault-tolerant systems and special devices used in special applications, such as space applications.

REFERENCES

- [1] N.S. Szabo and R.L. Tanaka, *Residue Arithmetic and its Applications to Computer Technology*. McGraw-Hill, 1967.
- [2] P.V. Ananda Mohan, *Residue Number Systems: Algorithms and Architectures*. Springer Science and Business Media, 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. Cardarilli, A. Nannarelli, and M. Re, *Reducing Power Dissipation in FIR filters Using the Residue Number System*, Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, 2000, vol. 1, pp. 320 - 323, 2000.
- [6] Jullien, G.A.; Krishnan, R.; Miller, W.C., *Complex digital signal processing over finite rings*, Circuits and Systems, IEEE Transactions on , vol.34, no.4, pp.365,377, Apr 1987
- [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 Conference on Signals, Systems, and Computers, pp. 1426-1429, Oct. 2007.
- [8] Jullien, G.A, *Implementation of Multiplication, Modulo a Prime Number, with Applications to Number Theoretic Transforms*, Computers, IEEE Transactions on , vol.C-29, no.10, pp.899,905, Oct. 1980
- [9] Del Re, A.; Nannarelli, A.; Re, M., *A tool for automatic generation of RTL-level VHDL description of RNS FIR filters*, Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings , vol.1, no., pp.686,687 Vol.1, 16-20 Feb. 2004
- [10] M. Petricca, P. Albicocco, G. C. Cardarilli, A. Nannarelli, and M. Re, *Power Efficient Design of Parallel/Serial FIR Filters in RNS*, in Proc. of 46th Asilomar Conference on Signals, Systems, and Computers. pp. 1015-1019, Nov. 2012.
- [11] G.C. Cardarilli, A. Del Re, A. Nannarelli and M. Re. *Power Characterization of Digital Filters Implemented on FPGA*, Proc. of 2002 IEEE International Symposium on Circuits and Systems (ISCAS), Vol. V, p. 801-804, Phoenix, Arizona (USA), May 2002
- [12] Pontarelli, S.; Cardarilli, G.C.; Re, M.; Salsano, A, *Totally Fault Tolerant RNS Based FIR Filters*, On-Line Testing Symposium, 2008. IOLTS '08. 14th IEEE International , vol., no., pp.192,194, 2008