

LOW-POWER IMPLEMENTATION OF POLYPHASE FILTERS IN QUADRATIC RESIDUE NUMBER SYSTEM

G. C. Cardarilli, A. Del Re, A. Nannarelli, and M. Re*

Dept. of Electrical Engineering, Univ. of Rome "Tor Vergata", Italy

* Informatics & Mathematical Modeling, Technical University, Denmark

ABSTRACT

The aim of this work is the reduction of the power dissipated in digital filters, while maintaining the timing unchanged. A polyphase filter bank in the Quadratic Residue Number System (QRNS) has been implemented and then compared, in terms of performance, area, and power dissipation to the implementation of a polyphase filter bank in the traditional two's complement system (TCS). The resulting implementations, designed to have the same clock rates, show that the QRNS filter is smaller and consumes less power than the TCS one.

1. INTRODUCTION

The polyphase decomposition is gaining increasing importance in digital telecommunication equipment because it allows the implementation of uniform filter banks with a reduced hardware complexity with respect to the classical solutions [1], [2]. Moreover, the sampling frequency is reduced early in the dataflow by decimation, resulting in relaxed timing constraints and reduced power consumption. Usually, these subsystems are implemented by using traditional number systems such as TCS or sign and magnitude, but, in recent years, the evolution of microelectronic technologies, gave to non traditional number systems a renewed importance. In particular the RNS arithmetic can be used to efficiently implement computational intensive signal processing blocks [3], [4], while the QRNS (Quadratic RNS) is convenient when dealing with complex numbers [5], [6]. In fact in QRNS a complex number is transformed into a pair of special integers, such that the complex multiplication requires only two integer multiplications.

Moreover the use of the QRNS (or RNS) allows the implementation of an operation with a given dynamic range into a set of modular operations with a smaller dynamic range implemented in parallel. The typical drawback presented by the QRNS is the overhead introduced by the input-output conversion from binary to QRNS and vice versa [7], [8], [9].

This work was partially supported by MIUR National Project: Optimization of Digital Signal Processing Structures.

In this work, the implementation of a polyphase filter bank based on the specifications of the Hot-Bird Satellite System for the Digital Video Broadcasting (DVB) is presented [10] [11]. The filter has been designed for the frequency demultiplexing of 8 channels coming from a RSAT-C type Ka-Band satellite transponder. As a first step, an error-free QRNS implementation of the complex filter is realized with two RNS structures, plus the input and output conversion blocks. The results obtained have been compared with both error-free (TCS) and truncated (TTCS) two's complement implementations, in terms of area occupation and power consumption. As a second step, the dynamic range of the QRNS is reduced, obtaining significant advantages over the traditional implementation. Such results confirm that using RNS/QRNS representations in computational intensive systems leads to better performance with reduced area occupation and power dissipation.

2. FILTER BANK SPECIFICATIONS

The designed system implements the demultiplexing unit required by the Ka-Band satellite transponder operating under RSAT-C signal specification [10], [11]. For that purpose, an eight channels polyphase filter bank has been designed. A 367 complex taps, Kaiser-Window pass-band prototype filter has been used, obtaining 43 dB out-band attenuation and 0.02 dB in-band ripple.

Four versions of the filter bank have been implemented:

1. Programmable \ Error-free Filter Bank.

The QRNS filter bank has been implemented without any truncation. Filter coefficients are loaded from the input port. An output dynamic range of 34 bit is required, 10 bits are needed for input samples, 12 bits to represent filter coefficients, 10 bits for the IDFT coefficients and 2 more bits related to the IDFT dynamic range extension. Synthesis results have been compared with those obtained for the TCS filter bank.

2. Fixed Coefficients \ Error-free Filter Bank.

In this implementation the coefficients are hardwired

resulting in a smaller silicon area and a lower power dissipation, but also in a reduced flexibility.

3. Programmable \ Truncated Filter Bank.

In common applications, the output dynamic range of digital FIR filters is reduced, by discarding the least significant bits. For that reason, a truncated version of the QRNS filter banks has been realized and the results have been compared with the TTCS implementation.

4. Fixed Coefficients \ Truncated Filter Bank.

As in item 2, but with truncation after the multiplication.

Details related to the implementations and the comparison results are given in the following sections.

3. RNS AND QRNS BASIC CONCEPTS

A Residue Number System (RNS) is defined by a set of relatively prime integers $\{m_1, m_2, \dots, m_P\}$. Its dynamic range is given by the product $M = m_1 \cdot m_2 \cdot \dots \cdot m_P$. Any integer $X \in \{0, 1, 2, \dots, M - 1\}$ has a unique RNS representation given by:

$$X \xrightarrow{RNS} (\langle X \rangle_{m_1}, \langle X \rangle_{m_2}, \dots, \langle X \rangle_{m_P})$$

where $\langle X \rangle_{m_i}$ denotes the operation $X \bmod m_i$ [3]. Operations on different m_i (moduli) are done in parallel

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

As a consequence, operations on large wordlengths can be split into several modular operations executed in parallel and with reduced wordlength.

In the complex arithmetic case, we can transform a complex number into a pair of integer numbers if the equation $q^2 + 1 = 0$ has two distinct roots q_1 and q_2 in the ring of integers modulo m_i (Z_{m_i}).

A complex number $x_R + jx_I = (x_R, x_I) \in Z_{m_i}$, with q root of $q^2 + 1 = 0$ in Z_{m_i} has a unique Quadratic Residue Number System representation given by

$$(x_R, x_I) \xrightarrow{QRNS} \begin{cases} (X_i, \hat{X}_i) & i = 0, 1, \dots, P \\ X_i = \langle x_R + q \cdot x_I \rangle_{m_i} \\ \hat{X}_i = \langle x_R - q \cdot x_I \rangle_{m_i} \end{cases}$$

The inverse QRNS transformation is given by

$$\begin{cases} x_R = \langle 2^{-1}(X_i + \hat{X}_i) \rangle_{m_i} \\ x_I = \langle 2^{-1} \cdot q^{-1}(X_i - \hat{X}_i) \rangle_{m_i} \end{cases}$$

where 2^{-1} and q^{-1} are the multiplicative inverses of 2 and q , respectively, modulo m_i :

$$\langle 2 \cdot 2^{-1} \rangle_{m_i} = 1 \quad \text{and} \quad \langle q \cdot q^{-1} \rangle_{m_i} = 1.$$

Therefore, the product of two complex numbers $x_R + jx_I$ and $y_R + jy_I$ is in QRNS

$$(x_R + jx_I)(y_R + jy_I) \xrightarrow{QRNS} (\langle X_i Y_i \rangle_{m_i}, \langle \hat{X}_i \hat{Y}_i \rangle_{m_i})$$

and it is realized by using two integers multiplications instead of four.

4. ERROR-FREE QRNS FILTER BANK IMPLEMENTATION

The architecture of a QRNS filter, shown in Fig. 1, is a direct consequence of the concepts of Section 3. The filter is divided into two structures (for X and \hat{X}) plus the input and output conversion blocks. Each structure is decomposed into P RNS paths, one for each modulus. In order to obtain the required dynamic range (34 bits), the following set of moduli has been chosen: $m_i = \{13, 17, 29, 37, 41, 53, 61\}$. Each RNS path is composed by 8 FIR filters and a block which computes the Inverse Discrete Fourier Transform (IDFT), as shown in the inset of Fig. 1.

In each tap of the FIR filters (implemented in direct form), the modular multiplication is computed by using the isomorphism technique: the product of the two residues is transformed into the sum of their indices, obtained by isomorphism [12]. Then, the products are added in a Wallace's tree structure.

The 8-point IDFT is implemented by the decimation in frequency (DIF) algorithm, which requires the implementation of $\frac{N}{2} \log(N) = 12$ butterflies in 3 stages [13]. Each butterfly has been implemented by using two look-up tables (LUTs) for the multiplication of signal samples by the IDFT coefficients. The internal dynamic range is increased as needed in order to avoid overflow.

The input and output conversions are implemented as described in [7] and [14]. Note that for the output, only one converter working at f_c (and mux/demux) is required to handle the conversion of the 8 channels at $f_c/8$. A total of 11 pipeline stages are required.

To evaluate the performance of the QRNS filter, a TCS filter based on the same specifications has been implemented. In the TCS polyphase filter, the FIR filters are also implemented in direct form. Each complex product is realized with 4 radix-4 Booth multipliers, and two Wallace's tree structures (for the real and imaginary parts) are used to sum up the products. The carry-save representation used in the trees is reduced to the conventional one in two's complement by a carry-look-ahead adder. Finally, the 8-point DIF IDFT is evaluated. The critical path has been broken into 7 pipeline stages.

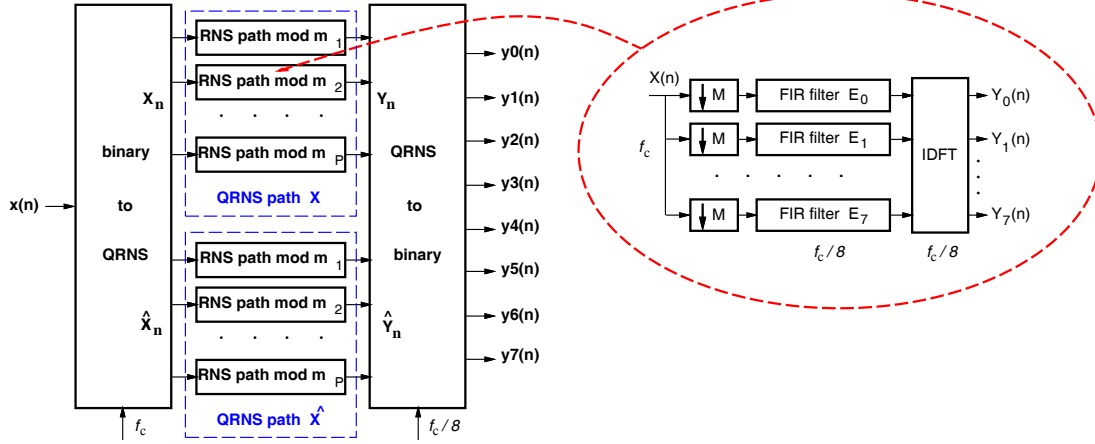


Fig. 1. QRNS polyphase filter bank.

For the programmable version of both TCS and QRNS filters, a state machine has been designed to load filter coefficients into the taps from the input port. When a load signal is given, the first 367 input samples are converted and loaded as the filter coefficients.

Both the TCS and the QRNS filters have been implemented in the AMS 0.35 μm library of standard cells. Delay, area and power dissipation have been determined by using Synopsys tools. The results for both programmable and wired coefficients filter banks are summarized in Table 1, where area is reported as number of NAND2 equivalent gates and power is computed at $f_c = 100$ MHz ($T = 10$ ns). Area and power dissipation do not take into account the contribution of interconnections.

The results show that the QRNS filter has higher latency (due to the input output conversions), but it can be clocked at the same rate of the traditional filter, and consequently, it can sustain the same throughput. Table 1 also shows that the conversion overhead of the QRNS filter does not offset the smaller area/power per processing elements, resulting in a saving of about 35% of silicon area and 38% of power dissipation for the programmable filter banks, and a saving of about 45% of area and about 50% of power dissipation for wired coefficients structures.

Progr. filters	T clock	latency	area	power
TCS	10 ns	7	2235K	2420 mW
QRNS	10 ns	11	1670K	1510 mW
Wired filters	T clock	latency	area	power
TCS	10 ns	7	775K	781 mW
QRNS	10 ns	11	425K	395 mW

Table 1. Results of the implementations.

5. TRUNCATED QRNS FILTER BANK IMPLEMENTATION

It is well known that non positional representations such as QRNS (and RNS) does not allow direct truncation starting from the residues. For this reason, we introduced two intermediate conversion stages (QRNS-binary and binary-QRNS) before the IDFT computation and perform truncation on the binary converted intermediate value (see Fig. 2). This introduced additional latency, but allowed the reduction of the dynamic range (and the number of moduli needed to cover it) from 34 bits to 22 bits in the filter banks, and to 27 bits in the IDFT datapaths, as shown in Fig. 2.

The truncated QRNS polyphase filter has been compared with a filter of the same characteristics implemented in the truncated two's complement system (TTCS). In the TTCS polyphase filter, the FIR sub-filters are implemented in direct form and the dynamic range is truncated to 15 bits after the sub-filter. As in the error-free version, each complex product is realized with 4 radix-4 Booth multipliers, and two Wallace's tree structures (for the real and imaginary parts) are used to sum the products. The critical path is broken into 7 pipeline stages.

The results for the truncated polyphase filters are summarized in Table 2.

Progr. filters	T clock	latency	area	power
TTCS	10 ns	7	2000K	2080 mW
QRNS (tr.)	10 ns	17	1050K	950 mW
Wired filters	T clock	latency	area	power
TTCS	10 ns	7	650K	630 mW
QRNS (tr.)	10 ns	17	260K	200 mW

Table 2. Results of implementations of truncated polyphase filters.

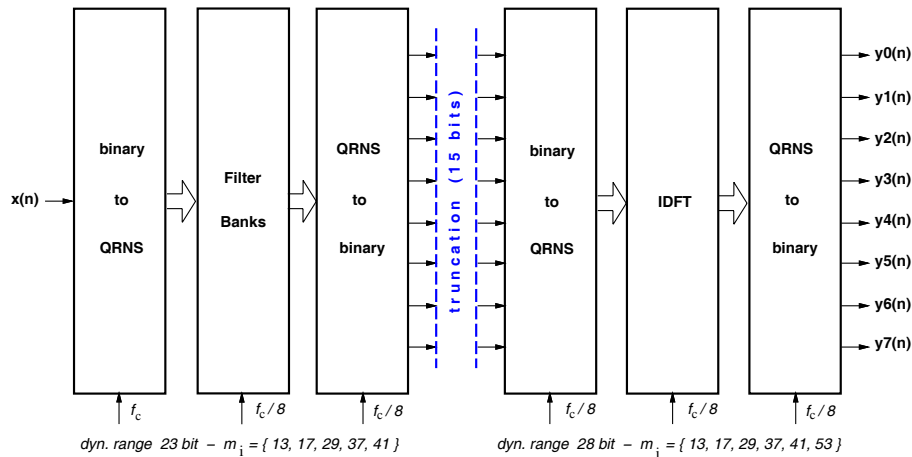


Fig. 2. Filter with truncated dynamic range.

Table 2 shows for the truncated QRNS programmable filter a reduction of about 50% of both area and power dissipation with respect to the programmable TTCS. For the truncated QRNS wired-coefficients filter the reduction in area is 60% and in power dissipation is 68% with respect to the wired-coefficients TTCS. Table 2 also shows that QRNS filters have higher latency because four more clock cycles are required for the QRNS/BIN and BIN/RNS conversions needed for truncation.

These results prove that the advantages in terms of area and power offered by the QRNS systems are not reduced by the introduction of two additional conversion stages needed for the truncation. Moreover, the reduction in the dynamic range of the filter banks (Fig. 2) is such that the overall saving in area and power with respect to the TCS are even larger for the truncated version of the filters.

6. CONCLUSIONS

In this work different architectures of a QRNS polyphase filter bank have been presented. The results obtained by the Synopsys synthesis tool showed that the conversions overhead of the QRNS filter does not offset the smaller area/power per processing elements, resulting in a saving of area and power dissipation both for the error-free and the truncated version of the filter bank. Greater benefits have been obtained by the truncated QRNS system, proving that advantages due to the shorter dynamic range are not reduced by the introduction of two more conversion stages.

7. REFERENCES

[1] P.P. Vaidyanathan, "Filter banks in digital communications," *IEEE Circuits and Systems Magazine*, vol. 1, pp. 4–25, 2001.
 [2] A. Del Re M. Re and G.C. Cardarilli, "Efficient implementation of a demultiplexer based on a multirate filter bank for the

skyplex satellites dvb system," *VLSI Design Journal, Taylor & Francis Ltd*, pp. 427–440, Vol. 15 (1) 2002.

[3] N.S. Szabo and R.I. Tanaka, *Residue Arithmetic and its Applications in Computer Technology*, McGraw-Hill, 1967.
 [4] M.A. Sodestrand, W.K. Jenkins, G. A. Jullien, and F. J. Taylor, *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*, IEEE Press, 1986.
 [5] F. J. Taylor, G. Papadourakis, A. Skavantzios, and A. Stouraitis, "A radix-4 FFT using complex RNS arithmetic," *IEEE Transactions on Computers*, pp. 573–576, June 1985.
 [6] M. Abdallah and A. Skavantzios, "On the binary quadratic residue system with noncoprime moduli," *IEEE Transactions on Signal Processing*, pp. 2085–2091, Aug. 1997.
 [7] G. Cardarilli, M. Re, R. Lojaco, and G. Ferri, "A new efficient architecture for binary to RNS conversion," *Proc. of European Conference on Circuit Theory and Design (ECCTD'99)*, vol. 2, pp. 1151–1154, 1999.
 [8] S. Piestrak, "A high-speed realization of a residue to binary number system converter," *IEEE Trans. Circuits Systems-II Analog and Digital Signal Processing*, vol. 42, pp. 661–663, Oct. 1995.
 [9] T. V. Vu, "Efficient implementation of the chinese remainder theorem for sign detection and residue decoding," *IEEE Trans. Circuits Systems-I*, vol. 45, pp. 667–669, June 1985.
 [10] Eutelsat, "Technical guide - annex d - skyplex," Jun. 1999.
 [11] Eutelsat, "Summary characteristic of the hot bird satellites," Jun. 1999.
 [12] I.M. Vinogradov, *An Introduction to the Theory of Numbers*, New York: Pergamon Press, 1955.
 [13] R.W. Schafer A.V. Oppenheim, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
 [14] G. Cardarilli, M. Re, and R. Lojaco, "A residue to binary conversion algorithm for signed numbers," *European Conference on Circuit Theory and Design (ECCTD'97)*, vol. 3, pp. 1456–1459, 1997.