

# A Hybrid RNS Adaptive Filter for Channel Equalization

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

Department of Electronics, University of Rome Tor Vergata, Rome, Italy

\*Dept. of Informatics & Math. Modelling, Technical University of Denmark, Kongens Lyngby, Denmark

**Abstract**—In this work a hybrid Residue Number System (RNS) implementation of an adaptive FIR filter is presented. The used adaptation algorithm is the Least Mean Squares (LMS). The filter has been designed to meet the constraints of specific class of applications. In fact, it is suitable for applications requiring a large number of taps where a serial updating of the filter coefficients is feasible (channel equalization or echo cancellation). In the literature, it has been shown that the RNS implementation of FIR filters grants earnings in area and power consumption due to the introduced arithmetic simplifications. Vice versa, the RNS implementation of the adaptation algorithm needs scaling circuits that are complex and expensive in RNS arithmetic. For this reason, a serial binary implementation of the adaptation algorithm is chosen. The advantages in terms of area and speed of the RNS adaptive filter with respect to the two's complement one have been evaluated for a standard cells implementation.

## I. INTRODUCTION

In recent years, it has been shown that the use of Residue Number Systems (RNS) in the implementation of application specific Digital Signal Processing (DSP) systems, and FIR filters in particular, grants great advantages in terms of area, speed and power consumption over their two's complement counterparts [1], [2], [3]. In particular, these advantages are related to the arithmetic simplifications offered by the Residue Number System in the implementation of the multipliers that are used in the parallel implementation of FIR filters. A multiplier is implemented in RNS by using the isomorphism technique [4], [5] requiring two small LUTs (its size is related to the moduli dynamic range) and a modular adder. Moreover the basic operations in RNS are based on LUTs and for this reason the RNS implementation of DSP algorithms is particularly suitable for a FPGA implementation [2]. These advantages have been proved also for RNS adaptive filters [6], [8]. From the application point of view, adaptive filters are often used for channel equalization in communication systems (for time-varying channels) and for echo cancellation in full-duplex communication systems [9], [10]. Different updating algorithms (LMS, RLS, FBLMS) are used, depending on the performance of the adaptation process such as, for example, steady state error and convergence speed. In certain class of applications requiring a relative small number of coefficients for the FIR filter and a fast adaptation rate, a fully parallel implementation of the updating algorithm is requested and it is feasible due the small number of filter taps (in fact, the LMS algorithm complexity measured as number of multipliers

grows with the filter taps). In applications requiring a large number of taps but characterized by a slow varying channel a serial updating of the filter coefficients can be used. In these cases, the complexity of the FIR filter can be reduced by using RNS while maintaining a serial implementation in binary arithmetic for the adaptation algorithm. In this paper, the RNS implementation of an adaptive LMS FIR filter is presented. The adaptive filter has been designed to match the requirements of low rate time variable channels characterized by high distortion in amplitude and phase, as in the case of satellite links, power line carrier communications and beam-forming networks. For this applications, a hybrid RNS-Binary architecture grants earnings in area and power consumption for the FIR filter implementation, but at the same time, a serial binary implementation of the adaptation algorithm avoids the use of complex and expensive scaling circuits in RNS. The paper is organized as follows: Section II presents background material on RNS computation and filtering, while in Section III and IV respectively the channel equalization/adaptive filtering background and channel equalizer design are shown. In Section V the RNS hybrid architecture is presented while in Section VI the conclusions are drawn.

## II. BACKGROUND ON RNS COMPUTATION AND FILTERING

A Residue Number System (RNS) is defined by a set of relatively prime integers  $\{m_1, m_2, \dots, m_P\}$  which identify the RNS base. 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  $X \bmod m_i$  [4]. Operations, such as addition and multiplication, on different  $m_i$  (moduli) are done in parallel

$$Z = X \text{ op } Y \xrightarrow{RNS} \begin{cases} Z_{m_1} = \langle X_{m_1} \text{ op } Y_{m_1} \rangle_{m_1} \\ Z_{m_2} = \langle X_{m_2} \text{ op } Y_{m_2} \rangle_{m_2} \\ \dots \quad \dots \quad \dots \\ Z_{m_P} = \langle X_{m_P} \text{ op } Y_{m_P} \rangle_{m_P} \end{cases} \quad (1)$$

As a consequence, operations on large wordlengths can be split into several modular operations executed in parallel and with reduced wordlength [4].

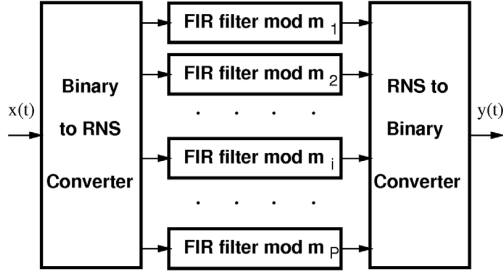


Fig. 1. FIR filter implemented in RNS.

A Finite Impulse Response (FIR) filter of order  $N$  is described by the expression

$$y(n) = \sum_{k=0}^{N-1} a_k x(n-k) \quad (2)$$

As a direct consequence of (1), (2) becomes in RNS

$$\begin{cases} Y_{m_1}(n) = \left\langle \sum_{k=1}^N \langle A_{m_1}(k) \cdot X_{m_1}(n-k) \rangle_{m_1} \right\rangle_{m_1} \\ Y_{m_2}(n) = \left\langle \sum_{k=1}^N \langle A_{m_2}(k) \cdot X_{m_2}(n-k) \rangle_{m_2} \right\rangle_{m_2} \\ \dots \\ Y_{m_P}(n) = \left\langle \sum_{k=1}^N \langle A_{m_P}(k) \cdot X_{m_P}(n-k) \rangle_{m_P} \right\rangle_{m_P} \end{cases}$$

and the filter can be implemented in RNS by decomposing it into  $P$  filters working in parallel, as shown in Fig. 1. Each of the  $P$  filters is implemented with additions and multiplications mod  $m_i$ .

A drawback of RNS is the need for converters from/to the conventional number system (two's complement or sign-and-magnitude). However, the conversion overhead has a small impact for high-order (large  $N$ ) filters such as the ones needed for adaptive filtering.

Moreover, a number of simple operations in binary arithmetic, such as sign-detection and truncation, are non trivial to implement in RNS.

Because of the complexity of modular multiplication in RNS, it is convenient to implement the product of residues by the isomorphism technique [7]. By using isomorphisms, the product of the two residues is transformed into the sum of their indices which are obtained by an isomorphic transformation. According to [7], if  $m$  is prime there exists a primitive radix  $q$  such that its powers modulo  $m$  cover the set  $[1, m-1]$ :

$$n = \langle q^w \rangle_m \quad \text{with } n \in [1, m-1] \text{ and } w \in [0, m-2].$$

The transformations  $n \rightarrow w$  and  $w \rightarrow n$  can be implemented with  $m-1$  entries look-up tables, if the moduli are not too large (for  $m_i \leq 251$  the tables have less than  $2^8$  entries).

Therefore, the product  $A(k)X(n-k)$  in the filter modulo  $m$  can be obtained as:

$$\langle A \cdot X \rangle_m = \langle q^W \rangle_m$$

where

$$W = \langle \hat{A} + \hat{X} \rangle_{m-1} \quad \text{with } A = \langle q^{\hat{A}} \rangle_m \text{ and } X = \langle q^{\hat{X}} \rangle_m$$

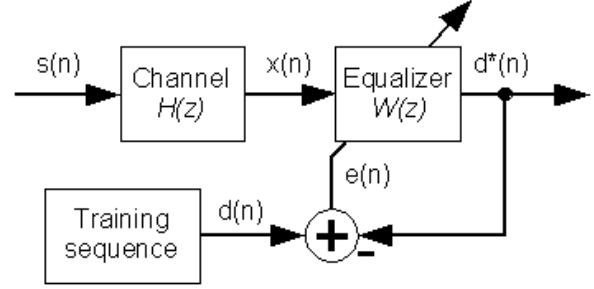


Fig. 2. Details of a baseband data transmission system equipped with an adaptive channel equalizer

Summarizing, to implement the modular multiplication the following operations are performed:

- i) Two direct isomorphic transformations (DIT) to obtain  $\hat{A}$  and  $\hat{X}$ ;
- ii) One modulo  $m-1$  addition  $\langle \hat{A} + \hat{X} \rangle_{m-1}$ ;
- iii) One inverse isomorphic transformations (IIT) to obtain the product.

By using isomorphic transformations, the modular multipliers are replaced by modular adders which are simpler and faster. On the other hand, the use of isomorphism limits the selection of the moduli to only those which are prime numbers.

### III. CHANNEL EQUALIZATION AND ADAPTIVE FILTERING BACKGROUND

In Fig. 2, the block diagram of a typical equalization system is shown. The equalizer is used to recover the distortion introduced by the channel while minimizing its effects. Given a channel  $H(z)$ , an equalizer works perfectly when its transfer function is  $W(z) = 1/H(z)$ . Consequently, the transfer function of the overall channel will be  $H(z)W(z) = 1$ , and the transmitted signal  $s(n)$  goes through the channel without any distortions. Given the input  $d(n)$  the signal  $x(n)$  is expressed as

$$x(n) = \mathbf{h}_c^H \mathbf{d}_n. \quad (3)$$

The variable filter is a  $p$  order FIR with coefficients

$$\mathbf{w}_n = [w_n(0), w_n(1), \dots, w_n(p)]^T. \quad (4)$$

The error signal is

$$e(n) = d(n) - d^*(n) \quad (5)$$

The variable filter estimates the desired signal by the convolution of the input signal with its impulse response. In vector notation this is expressed as

$$d^*(n) = \mathbf{w}_n^T \mathbf{x}(n) \quad (6)$$

where

$$\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-p)]^T \quad (7)$$

TABLE I  
SYSTEM REQUIREMENTS

Description	Before equalization	After Equalization
Amplitude distortion	5 dB <sub>pp</sub>	0.5 dB <sub>pp</sub>
Phase distortion	10 degree <sub>pp</sub>	2 degree <sub>pp</sub>
Input sampling rate	160 Msps	

is the input vector to the adaptive filter. The updated coefficients are

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta \mathbf{w}_n. \quad (8)$$

Different algorithms can be used for the coefficient updating, depending on system requirements. Among the most used there are: Least Mean Square (LMS), Block LMS, Fast Block LMS and the Transform Domain LMS [13].

#### IV. CHANNEL EQUALIZER DESIGN

In order to evaluate the benefits coming from the proposed hybrid architecture, some requirements related to the channel equalization system have been fixed. In Table I, the amplitude and phase distortions before and after the equalizer are summarized. Moreover, a quasi-stationary channel is assumed, and a training sequence is used to update the filter coefficients. The above assumptions allow for a serial updating of the variable filter coefficients, resulting in area and power consumption savings.

The LMS algorithm has been chosen for the coefficient updating, since it is not very expensive in terms of hardware requirements with respect to other algorithms, and results sufficiently fast in tracking. The coefficient update vector  $\Delta \mathbf{w}_n$  using the LMS algorithm is

$$\Delta \mathbf{w}_n = \mu \cdot \mathbf{x}(n) \cdot e(n). \quad (9)$$

where  $\mu$  is a scaling factor chosen to normalize the error with respect to the input signal power.

Floating Point (FLP) simulations have been carried out to define the minimum number of the taps of the equalizer, obtaining  $N = 192$ . The number of bits used to represent the input samples  $x(n)$  and the coefficients of the variable filter  $a_k$  is 12, resulting in a dynamic range for the FIR filter of 32 bits. The scaling factor  $\mu$  is chosen to be a power of two, so that only one multiplication is needed for the serial implementation of the LMS algorithm.

In Fig. 3, the block diagram of the proposed channel equalizer is sketched. When the system is processing the user data, the serial LMS block is frozen (the control unit is not shown). Once the training sequence is detected, the variable filter output signal  $d^*(n)$  is used to calculate the error signal  $e(n)$  and the new coefficients are computed one at a time by means of eq. 8. When all the coefficients have been updated, an iteration of the LMS algorithm is completed and the variable filter coefficients are updated with the new ones. The main drawback of this solution is the fact that a single iteration of

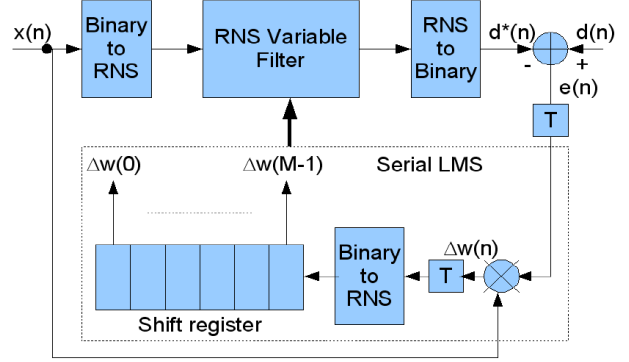


Fig. 3. Proposed Adaptive Filter Block Diagram

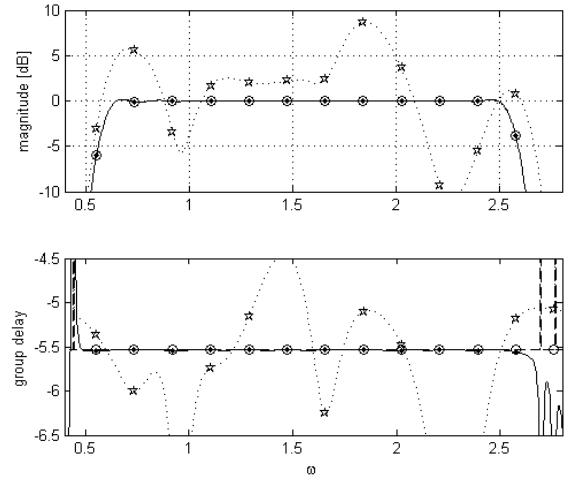


Fig. 4. Magnitude and phase response of signals in passband (simulation time:  $5.5 \times 10^6 T_{clk}$ ).  $\star \cdots$  channel to be recovered;  $\circ - -$  Reference channel;  $\bullet - -$  equalized channel.

eq. 8 is carried out each time the training sequence is detected, meaning that the LMS algorithm converges in a longer time.

In Fig. 4 the results of the fixed point simulations are shown. The magnitude and group delay of the disequalized channel response (star marker), the reference channel (circle marker) and the output of the adaptive filter (bullet marker) show a good matching with the desired response both in magnitude and phase.

#### V. HYBRID RNS ARCHITECTURE AND IMPLEMENTATION

The straightforward implementation of the proposed equalizer in RNS would require a dynamic range extension to 44 bits, because the scaling operation is not trivial in RNS. In fact, the variable filter is a 192 taps filter, 12 bits for input samples and 12 bits for coefficients, resulting in an output dynamic range of 32 bits. So the error  $e(n)$  would be represented with 32 bits and the multiplication by  $x(n)$  required by the LMS algorithm adds 12 extra bits. In order to limit the dynamic range of the RNS implementation and the size of the filter, the multiplication needed for the serial

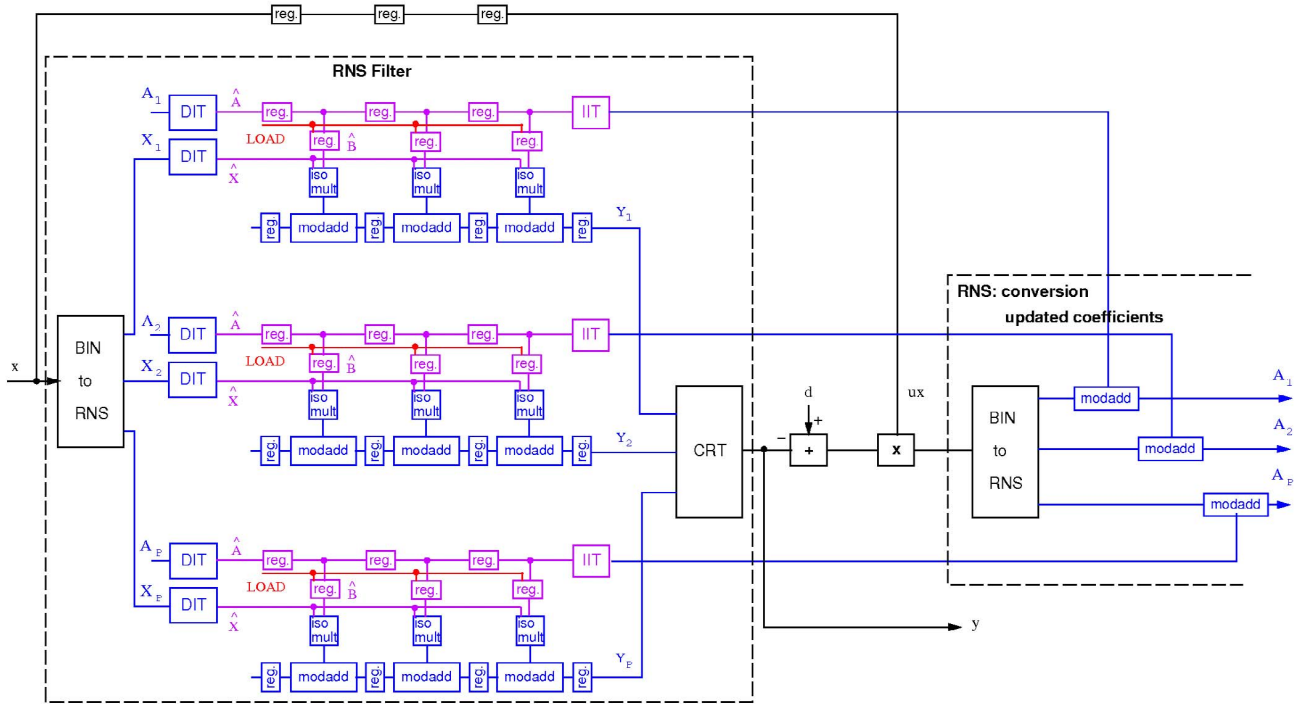


Fig. 5. Hybrid Adaptive Filter Architecture.

LMS algorithm implementation has been carried out in binary arithmetic. In this way, the adaptive filter is partly implemented in RNS (the FIR filter) and partly in two's complement (the LMS algorithm). In this hybrid architecture, we take advantage of the better characteristics of RNS filters concerning reduced area and power dissipation, and we perform the scaling to obtain the new filters coefficients in two's complement by a simple truncation.

The resulting system block diagram is shown in Fig. 5 (3 RNS channels,  $P = 3$ ). In the figure, the parts implemented in RNS are displayed inside dotted boxes. In addition to the hardware for the programmable RNS FIR filter, the variable filter of Fig. 5 requires an extra binary-RNS converter (right hand side of the figure) to convert the updated coefficients in RNS and an extra bank of registers to store the new coefficients while the update is in progress.

The characteristics in terms of speed and area of the RNS filter depends on the choice of the moduli set used to match the given dynamic range. In order to select the RNS base for the best delay/area/power tradeoffs, the tool described in [11] has been used. It implements a selection of the moduli set based on the characterization of all the RNS components of the filter (modular adders and multipliers, converters, etc.) under several timing conditions.

For the 192-taps and 32-bit dynamic range filter proposed in this paper, the selected moduli set is  $\{ 5, 7, 11, 13, 17, 19, 23, 128 \}$ . The hybrid RNS adaptive filter has been compared with a filter implemented in two's complement (TCS). Table II reports the characteristics of the variable filter implemented in

TABLE II  
RESULTS OF FILTERS IMPLEMENTATIONS.

	clock freq. [MHz]	Area [mm <sup>2</sup> ]	ratio	Power [mW]	ratio
TCS	200	3.1	1.00	215	1.00
RNS	200	1.7	0.55	125	0.58

RNS and in TCS. Both filters were synthesized by Synopsys Design Compiler by using the STM 90 nm library of standard cells [12].

Table II shows that the implementation of the hybrid RNS implementation of the LMS adaptive filter offers savings in area and power dissipation of about 50%.

## VI. CONCLUSION

In this work, a hybrid Residue Number System (RNS) implementation of an adaptive FIR filter based on the Least Mean Squares (LMS) adaptation algorithm is presented. The filter has been designed to meet the constraints of applications requiring a large number of taps where a serial updating of the filter coefficients is feasible such as channel equalization or echo cancellation. The advantages in terms of area and speed of the hybrid RNS adaptive filter with respect to the two's complement one have been evaluated for a standard cells implementation showing savings in area and power dissipation of about 50%.

## REFERENCES

- [1] G.C. Cardarilli, A. Del Re, A. Nannarelli, M. Re, "Low Power and Low Leakage Implementation of RNS FIR Filters", *Conference Record of the*

- Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, October 28 – November 1, 2005, Page(s):1620–1624.
- [2] G.C. Cardarilli, A. Del Re, A. Nannarelli, M. Re, “Power characterization of digital filters implemented on FPGA”, *IEEE International Symposium on Circuits and Systems, ISCAS2002*, Volume 5, 26–29 May 2002, Pages V. 801-804.
  - [3] A. Del Re, A. Nannarelli, M. Re, “Implementation of digital filters in carry-save residue number system”, *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, Volume 2, 4–7 Nov. 2001 Pages 1309-1313.
  - [4] N.S. Szabo and R.I. Tanaka, “Residue Arithmetic and its Applications in Computer Technology”, *New York: McGraw-Hill*, 1967. October 28 – November 1, 2005, Page(s):1620–1624.
  - [5] M.A. Sodestrand, W.K. Jenkins, G. A. Jullien, F. J. Taylor, “Residue Number System Arithmetic: Modern Applications in Digital Signal Processing”, *New York: IEEE Press*, 1986.
  - [6] W.K. Jenkins, B.A. Schnauffer, “Fault Tolerant Adaptive Filters Based on the Block LMS Algorithm”, *IEEE International Symposium on Circuits and Systems*, 3-6 May 1993, Page(s):862 - 865.
  - [7] I.M. Vinogradov. *An Introduction to the Theory of Numbers*. New York: Pergamon Press, 1955.
  - [8] J. Ramirez, U. Meyer-Baese, A. Garcia and A. Lloris, “Design and Implementation of RNS-based Adaptive Filters”, *Proceedings of the 13th International Conference on Field Programmable Logic and Applications*, Lisbon – Portugal, September 1–3, 2003, pp. 1135–1138.
  - [9] Y. Tsuda, T. Shimamura, “An improved NLMS algorithm for channel equalization”, *IEEE International Symposium on Circuits and Systems, ISCAS2002*, Volume 5, 26–29 May 2002, Pages V. 353356.
  - [10] J.V. Lopez, J.C. Sanchez, H.P. Meana, “Adaptive echo canceller using a modified LMS algorithm” *Proceedings of the 2nd International Conference on Electrical and Electronics Engineering*, 7–9 Sept. 2005 Pages 93–96.
  - [11] A. Nannarelli, A. Del Re and M. Re, “A Tool for Automatic Generation of RTL-Level VHDL Description of RNS FIR Filters”, *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, 16–20 Feb. 2004, Vol. 1, Pages 686-687.
  - [12] STMicroelectronics. 90nm CMOS090 Design Platform. [Online]. Available: <http://www.st.com/stonline/prodpres/dedicate/soc/asic/90plat.htm>
  - [13] B. Farang-Boroujeny “Adaptive Filters - Theory and Applications”, John Wiley and Sons, ISBN 0-471-98337-3.