Scaling algorithm designed to reduce circuit costs for modular digital filters

I A Kalmykov¹, N S Sidorov¹, K T Tyncherov², A A Vorohov³

¹ Federal State Autonomous Educational Institution for Higher Education "North-Caucasus Federal University", 1, Pushkin St., Stavropol, 355009, Russian Federation
² Ufa State Petroleum Technological University, Branch of the University in the City of Oktyabrsky, 54a, Devonskaya St., Oktyabrsky, Republic of Bashkortostan, 452607, Russian Federation
³ Novosibirsk State Technical University, 20, Karl Marks ave., Novosibirsk, 630073, Russian Federation

E-mail: kia762@yandex.ru

Abstract. Currently, there is a tendency to increase the speed and accuracy of data processing in computer networks (CN). It is possible to meet these requirements through the use of new technologies in CN telecommunication devices that rely on residue class codes (RNS). In these codes, input values are presented as sets of remainder moduli of the RNS base selected. As a result, arithmetic operations are performed in parallel, which provides an increased speed of signal processing. Therefore, RNS codes are used in high-speed modular digital filters (MDF). To advance the accuracy of digital signal processing in MDF, the number of RNS bases is increased, which leads to greater circuit costs. It is possible to remedy this shortcoming by scaling an MDF response. Hence, it is crucial to develop a scaling algorithm to reduce circuit costs for MDF implementation.

1. Introduction

The effectiveness of modern computer networks is largely determined by the technologies exploited in CN telecommunication devices. Digital signal processing (DSP) is widely used to ensure high speed transmission and signal processing in these devices. Digital signal processing is a technology designed to solve the tasks set by high quality real-time signal processing [1, 2]. The main algorithm that implements DSP is the digital filtering algorithm. Digital filtering in telecommunication devices is used in pre-processing noise-control units. A digital filter (DF) processes a sequence of input reports in order to suppress certain frequency bands [3, 4]. Digital filtering is based on arithmetic operations of multiplying input samples by filtering coefficients, followed by summing the resulting products. Since the time for calculating a single DF response should not exceed the sampling period of a signal, digital filters should use parallel calculation methods. One of the promising solutions to this problem is the use of RNS codes. Due to parallel and independent processing of residues, RNS codes provide parallelization at the level of arithmetic operations [5, 6]. This provides maximum performance of modular digital filter (MDF).

However, MDF efficiency largely depends on the capacity of filtering coefficients. Due to a limited bit depth, MDF frequency response and output signal are distorted. To remedy this
shortcoming, the capacity of digital batteries and multipliers is improved several times in positional digital filters when undistorted intermediate results are obtained. With RNS codes, this result is achieved by increasing the number of code bases. However, this leads to a significant increase in circuit costs required for performing inverse transformations from RNS to positional code. Hence, the paper aims to reduce circuit costs by scaling an MDF response. Therefore, it is crucial to develop a scaling algorithm to reduce circuit costs incurred for MDF implementation.

2. Materials and methods
The use of digital filtering in telecommunication devices allows efficient signal primary processing in the presence of interference. Currently, filters with a finite impulse response (FIR) and filters with an infinite impulse response (IIR) are known [2, 3]. The most common are FIR filters, the algorithm of which is determined by the expression

$$y(n) = \sum_{k=0}^{N-1} h(n)x(n-k),$$

where $h(n)$ is a DF impulse response; $x(n)$ is an input vector; $y(n)$ is output sequences (filter response); $N$ is a filter order.

The impulse response is determined by the filter coefficients $h_0$, $h_1$, ..., $h_{N-1}$. The capacity of these coefficients is the main factor that determines the type of filter frequency response. With this increase in the order of $N$ and the bit depth of filter coefficients, a wordlength dynamic range should be significantly extended, which affects the speed of digital filtering. That is why, it is difficult to deal with high-frequency signals.

Since the time for calculating a single DF response should not exceed the sampling period of a signal, digital filters should use parallel computation methods. This problem can be solved by RNS codes. According to [5, 6], co-prime integers $m_1, m_2, ..., m_s$ are used to obtain an RNC code. Then, the remainders modulo $W$ of the bases $w_i \equiv w \mod m_i, i = 1, 2, ..., s$ are calculated. Then we get

$$W = (w_1, w_2, ..., w_s),$$

With RNS codes, the following arithmetic operations are performed [7,8]:

$$W + X = ((w_1 + x_1) \mod m_1, (w_2 + x_2) \mod m_2, ..., (w_s + x_s) \mod m_s),$$

$$W + X = ((w_1 + x_1) \mod m_1, (w_2 + x_2) \mod m_2, ..., (w_s + x_s) \mod m_s),$$

$$W + X = ((w_1 + x_1) \mod m_1, (w_2 + x_2) \mod m_2, ..., (w_s + x_s) \mod m_s),$$

where $X = x_1 \mod m_1; i = 1, 2, ..., s$.

MDF functioning is organized by properly selecting the RNS bases $m_1, m_2, ..., m_s$ so that the condition is

$$D_{RNS}^* > D,$$

where $D = 2^L$; $L$ is an input DF wordlength; $D_{RNS}^* = \prod_{i=1}^{s} m_i$ is RNS range.

If the filter coefficients are represented as integers $h^*(n) = Kh(n)$, where $K$ is the scaling factor, then digital filtering in the RNS code looks like

$$y^*_i(n) = \sum_{k=0}^{N-1} h^*_i(n)x_i(n-k),$$

where $x(n-k) \equiv x_i(n-k) \mod m_i; h^*(n) \equiv h_i(n) \mod m_i; y^*_i(n) \equiv y^*_i(n) \mod m_i; i = 1, 2, ..., s.$
To restore the proper MDF response, the result obtained in the RNS code is translated into a positional code, and subsequently divided by the scaling factor $K$.

Since the type of frequency response and the output result depend on the capacity of intermediate data, the number of bases $m_1, \ldots, m_s, m_{s+1}, \ldots, m_{s+v}$ is increased in the MDF, which entails the extension of the range to a value $D^{s+v}_{\text{RNS}} = \prod_{i=1}^{s+v} m_i$. However, this increases the circuit costs required for performing inverse transformations from the RNS to the positional code. According to [6–10], this is allowed by the Chinese remainder theorem

$$Y = \sum_{i=1}^{s+v} y_i B_i \mod D^{s+v}_{\text{RNS}},$$

where $B_i$ is an orthogonal basis for RNS code; $B_i = \text{g}_i D^{s+v}_{\text{RNS}} m_i^{-1}$; $\text{g}_i$ is the weight of the orthogonal basis, the value of which is selected from the condition $\text{g}_i \left(D^{s+v}_{\text{RNS}} m_i^{-1}\right) \equiv 1 \mod m_i$.

To build this converter, $U_1 = s + v$ of LUT tables is necessary to implement the operation $Y_i = y_i B_i \mod D^{s+v}_{\text{RNS}}$, as well as $U_2 = \left\lfloor \log_2 (s + v) \right\rfloor$ of LUT tables to implement the operation $Y = \sum_{j=1}^{U_1} Y_j \mod D^{s+v}_{\text{RNS}}$. Then the circuit costs are determined by the number of LUT tables according to

$$U = U_1 + U_2 = \sum_{i=1}^{s+v} \left(s + v + \left\lfloor \log_2 (s + v) \right\rfloor \right).$$

To reduce the circuit costs, an algorithm was developed for scaling the results. It involves the following steps.

Step 1. From the MDF response presented as $y(n) = (y_1(n), y_2(n), \ldots, y_{s+v}(n))$, the senior remainder $y_{s+v}$ is subtracted. Then we have

$$A^{s+v-1}(n) = \begin{pmatrix} y_1(n) - y_{s+v}(n) \\ \vdots \\ y_{s+v-1}(n) - y_{s+v}(n) \end{pmatrix} = \begin{pmatrix} A_1^{s+v-1}(n), \ldots, A_{s+v-1}^{s+v-1}(n) \end{pmatrix}.$$  \hfill (10)

The resulting residues are respectively multiplied by a constant $m_i^{-1}$, $i = 1, 2, \ldots, s + v - 1$.

$$y^{s+v-1}(n) = \begin{pmatrix} A_1^{s+v-1}(n) m_1^{-1} \\ \vdots \\ A_{s+v-1}^{s+v-1}(n) m_{s+v-1}^{-1} \end{pmatrix} = \begin{pmatrix} y_1^{s+v-1}(n), y_2^{s+v-1}(n), \ldots, y_{s+v-1}^{s+v-1}(n) \end{pmatrix}.$$  \hfill (11)

Step 2. The residue $y_{s+v-1}^{s+v-1}$ is subtracted from the result $y_{s+v-1}^{s+v-1}(n)$. Then

$$A^{s+v-2}(n) = \begin{pmatrix} y_1^{s+v-1}(n) - y_{s+v-1}^{s+v-1}(n) \\ \vdots \\ y_{s+v-2}^{s+v-1}(n) - y_{s+v-1}^{s+v-1}(n) \end{pmatrix} = \begin{pmatrix} A_1^{s+v-2}(n), \ldots, A_{s+v-2}^{s+v-2}(n) \end{pmatrix}.$$  \hfill (12)

The resulting residues are respectively multiplied by a constant $m_i^{-1}$, $i = 1, 2, \ldots, s + v - 2$. As a result, we have an RNS combination consisting of the residues $s + v - 2$

$$y^{s+v-2}(n) = \begin{pmatrix} A_1^{s+v-2}(n) m_1^{-1} \\ \vdots \\ A_{s+v-2}^{s+v-2}(n) m_{s+v-2}^{-1} \end{pmatrix} = \begin{pmatrix} y_1^{s+v-2}(n), y_2^{s+v-2}(n), \ldots, y_{s+v-2}^{s+v-2}(n) \end{pmatrix}.$$  \hfill (13)

The algorithm comprises $v$ stages. At the last stage, a combination $\tilde{y}(n) = (\tilde{y}_1(n), \tilde{y}_2(n), \ldots, \tilde{y}_s(n))$ will be less than the source by $K' = m_{s+1} \cdots m_{s+v-1} \cdot m_{s+v}$ times. Moreover, the use of this algorithm will reduce MDF circuit costs.
3. The results and discussion

Let the modular digital filter process 16-bit input values and 16-bit filter coefficients. To fulfill the condition (6), the selected RNS bases are $m_1 = 3, m_2 = 5, m_3 = 7, m_4 = 11, m_5 = 13, m_6 = 19$. Then, the RNS range is $D_{RNS}^X = 282585$. The multiplication requires 32-bit output. For this, the bases added are $m_7 = 23, m_8 = 29, m_9 = 31, m_{10} = 37$. Since the MDF impulse response has negative coefficients, they are represented in the RNS code as

$$-h(n) = \left( |D_{RNS}^{++} - h(n)|_{m_1}, |D_{RNS}^{++} - h(n)|_{m_2}, \ldots, |D_{RNS}^{++} - h(n)|_{m_{10}} \right)$$

(14)

Thus, the MDF positive coefficients will be in the first half of the range $D_{RNS}^{++} = 218257003965$, and the negative coefficients in the RNC code will be represented by the digits of the second half of the range. The methods for presenting positive and negative values in RNS are provided in [5,6,8] in greater detail.

The study was conducted in an interactive environment for developing algorithms and data analysis – a Matlab 2017 software package. A prototype was a 15-order digital filter with coefficients $h_0 = h_{14} = -0.00174, h_{1} = h_{13} = -0.003295, h_2 = h_{12} = 0.016274, h_3 = h_{11} = -0.01031, h_4 = h_{10} = -0.06751, h_5 = h_9 = 0.018207, h_6 = h_8 = 0.303234, h_7 = 0.478305.$

The 15-order modular digital filters were synthesized with and without the formulated scaling algorithm. To obtain integer values, the coefficient $K = 10^5$ was chosen. Then the MDF coefficients were $h_0 = h_{14} = -1.74, h_1 = h_{13} = 329, h_2 = h_{12} = 1627, h_3 = h_{11} = -1031, h_4 = h_{10} = -6751, h_5 = h_9 = 1820, h_6 = h_8 = 30323, h_7 = 47830.$ Once entering the MDF, the input sequence $x(n) = (60000, 61001, 56089, 56789, 46002, 35678, 47578, 35456, 60231, 59789, 47578, 36546, 24367, 12567),$ was converted to the RNS code. Thus, $x(0) = 60000 = (0, 0, 3, 1, 5, 17, 16, 28, 15, 23), \ x(1) = 61001 = (2, 1, 3, 6, 5, 11, 5, 14, 24, 25).$ In this case, the coefficients of the digital filter were also presented in the RNS code. Then we had $h(0) = -174 = (0, 1, 1, 2, 8, 16, 10, 0, 12, 21)$ and $h(1) = 329 = (2, 4, 0, 10, 4, 6, 7, 10, 19, 33).$ According to (7), a zero sample of the output sequence was calculated in the RNS as

$$y^*(0) = y^*(0)x(0) = (0, 3, 1, 1, 6, 22, 0, 25, 31).$$

The first sample of the output sequence was calculated in the RNS as

$$y^*(0) = h^*(0)x(0) = (0, 1, 3, 6, 8, 12, 1, 19, 15, 35).$$

Similarly, the remaining samples of the MDF response were calculated. Let us scale the zero sample of the output sequence.

Step 1. Subtract the senior remainder $y_{nev} = 31$ from the response

$$y^*(0) = (0, 0, 3, 1, 1, 6, 22, 0, 25, 31), \quad A^9(0) = \left( |0 - 31|^R_{m_1}, |0 - 31|^R_{m_2}, |3 - 31|^R_{m_3}, |0 - 31|^R_{m_4}, |6 - 31|^R_{m_5}, |0 - 31|^R_{m_6}, |22 - 31|^R_{m_7}, |0 - 31|^R_{m_8}, |25 - 31|^R_{m_9}, 0 \right) = (2, 4, 0, 3, 9, 13, 14, 27, 25, 0).$$

The resulting residues are respectively multiplied by a constant $37^{-1}|m_{10}$. Then we have

$$y^*(0) = \left( 2, 4, 0, 3, 9, 13, 14, 27, 1, 22, 0, 25, 31 \right) =$$

$$= (2, 2, 0, 9, 2, 6, 1, 7, 30).$$

Step 2. Subtract $y_0^* = 30$ from the output $y^*(0) = (2, 2, 0, 9, 2, 6, 1, 7, 30) \quad y_0^* = 30.$ We have

$$A^8(0) = (2, 2, 5, 11, 14, 17, 6, 0).$$

4
The resulting residues are respectively multiplied by a constant $31^{-1}$ $m_h$. Then we have

$$y^8(0) = (2, 2, 4, 5, 10, 17, 5, 3).$$

Step 3. Subtract $y^8 = 3$ from the output $y^8(0) = (2, 2, 4, 5, 10, 17, 5, 3)$. We have

$$A^7(0) = \{[2 - 3]_{13}, [2 - 3]_{15}, [4 - 3]_{17}, [5 - 3]_{19}, [10 - 3]_{13}, [17 - 3]_{19}, [5 - 3]_{23}, 0\} = (2, 4, 1, 2, 7, 14, 2, 0).$$

The resulting residues are respectively multiplied by a constant $29^{-1}$ $m_h$. Then we have

$$y^7(0) = \{[2 \cdot 2]_{13}, [4 \cdot 4]_{15}, [1 \cdot 1]_{17}, [3 \cdot 4]_{19}, [1 \cdot 2]_{23}\} = (1, 1, 5, 11, 9, 8).$$

Step 4. Subtract $y^7 = 8$ from the output $y^7(0) = (1, 1, 5, 11, 9, 8)$. We have

$$A^6(0) = \{[1 - 8]_{13}, [1 - 8]_{15}, [5 - 8]_{17}, [1 - 8]_{19}, [9 - 8]_{13}, [8 - 8]_{19}\} = (2, 3, 0, 8, 3, 1, 0).$$

The resulting residues are respectively multiplied by a constant $23^{-1}$ $m_h$. Then we have

$$y^8(0) = y^* = \{[2 \cdot 2]_{13}, [3 \cdot 2]_{15}, [1 \cdot 4]_{17}, [3 \cdot 4]_{19}, [1 \cdot 5]_{23}\} = (1, 1, 0, 8, 12, 5).$$

This completes the scaling process. Let us perform the inverse transformation

$$Y(0) = \sum_{i=1}^{6} y^*_i(0) B_i \mod D^{y^*_{\text{RNS}}} = (1 \cdot 95095 + 1 \cdot 171171 + 0 \cdot 40755 + 8 \cdot 181545 + 12 \cdot 21945 +
+ 5 \cdot 60060) \mod 285285 = 285271$$

Since the resulting number belongs to the second part $D^{y^*}_{\text{RNS}} = 285285$, the zero response is $Y^*(0) = -14$. To restore the true result, it is necessary to calculate

$$Y = Y^*(0) \cdot K^{-1} (K^{-1})^{-1} = -14 \cdot 7.65049 = -107.10686.$$

Other output samples are scaled in a similar fashion.

The studies have shown that the above scaling algorithm reduced circuit costs. Thus, 19 LUT-tables are required to ensure the implementation of the inverse converter without scaling for the example provided. With the formulated scaling algorithm, the circuit costs will be 11 LUT-tables, which is 1.72 times less. The formulated scaling method leads to a slightly increased error in calculating the responses of the digital filters. In the example considered, it did not exceed 2.3% in comparison with the positional DF.

4. Conclusion

The paper presents an algorithm developed for scaling output samples of the MDF. To evaluate the effectiveness of this algorithm, two modular 15 order digital filters were synthesized in Matlab 2017. A comparative analysis showed that the use of the developed scaling algorithm for the given example reduced the circuit costs by 1.72 times in comparison with the MDF employed without this algorithm. A further direction towards the improvement of the scaling algorithm is to reduce the calculation error.

5. Acknowledgments

This work was supported by the Russian Foundation for Basic Research, project No. 18–07-01020.

References

[1] Antonio A 2016 Digital Signal Processing (New York: McGraw-Hill)

[2] Paulo S R Dinis 2010 Digital Signal Processing. System Analysis and Design (New York: Cambridge University Press)

[3] Saeed V V 2010 Advanced Digital Signal Processing and Noise Reduction. (London: John Wiley&Sons.Ltd)
[4] Shao P, Wu Z, Zhou X 2017 *Soft Comput* 21 pp2631–2642
[5] Omondi A, Premkumar B 2007 *Residue Number Systems: Theory and Implementation* (London: Imperial College Press)
[6] Mohan P V 2016 *Residue Number Systems: Theory and Applications* (New York: Springer)
[7] Tyncherov K T, Chervyakov N I, Selivanova M V, Kalmykov I A 2018 *Bulletin of the Tomsk Polytechnic University. Geo Assets Engineering* 329(3) pp36-43
[8] Kuleshova L S, Mukhametshin V V and Safiullina A R 2019 Applying information technologies in identifying the features of deposit identification under conditions of different oil-and gas provinces *Journal of Physics: Conference Series (ITBI 2019 – International Conference "Information Technologies in Business and Industry")* 1333(7) 1-5 DOI: 10.1088/1742-6596/1333/7/072012
[9] Andreev V E, Chizhov A P, Kotenev Yu A, Sultanov Sh Kh, Mukhametshin V Sh and Chibisov A V 2019 Prediction of Gas Exposure in the Conditions of the Oil Fields of the Volga-Urals *Atlantis Highlights in Material Sciences and Technology (ISEES 2019 – International Symposium "Engineering and Earth Sciences: Applied and Fundamental Research" dedicated to the 85th anniversary of H.I. Ibragimov)* 1 666–669 DOI: 10.2991/isees-19.2019.132
[10] Kaplun D I, Klionskiy D M, Bogaevskiy D V 2017 *Automatic Control and Computer Sciences* 51(3) pp167-173