IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 February 2023, accepted 13 March 2023, date of publication 17 March 2023, date of current version 22 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3258689

RESEARCH ARTICLE

Digital Filter Architecture Based on Modified Winograd Method F(2 × 2, 5 × 5) and Residue Number System

GEORGII VALUEV[1], MARIA VALUEVA[1], (Graduate Student Member, IEEE), MIKHAIL BABENKO[1], AND ANDREI TCHERNYKH[2,3], (Member, IEEE)

[1]North-Caucasus Center for Mathematical Research, North-Caucasus Federal University, 355017 Stavropol, Russia
[2]Control/Management and Applied Mathematics, Ivannikov Institute for System Programming, 109004 Moscow, Russia
[3]CICESE Research Center, Computer Science Department, Ensenada 22800, Mexico

Corresponding author: Georgii Valuev (mail@gvvaluev.ru)

This work was supported by the Russian Science Foundation under Grant 19-71-10033.

ABSTRACT Improving the characteristics of digital signal processing devices is an important task in many practical problems. The paper proposes the architecture of a two-dimensional digital filter with a 5 × 5 mask, in which calculations are performed according to the Winograd method in the Residue Number System (RNS) with moduli of a special type. Theoretical analysis and hardware Field-Programmable Gate Array simulation are presented. The results show that the fragment throughput fr/s (number of fragments per second) of the device is 29.6% – 724.7% higher than state-of-the-art solutions. This is achieved by the combination of the Winograd method, which reduces the number of multiplications, with the RNS arithmetic, which performs addition and multiplication under smaller operands in parallel. However, our experiments showed that the proposed method requires up to 2.54% – 11.01% more Look-Up Tables and 3.58% – 19.83% higher power consumption compared to known analogues.

INDEX TERMS Residue number system, winograd method, field-programmable gate array, digital filter, digital signal processing.

I. INTRODUCTION

Digital filters are widely used as components of complex digital signal processing and analysis systems. These systems are used in practical tasks such as medicine [1], [2], [3], [4], geolocation [5], [6], video surveillance systems [7], product quality control in production [8], and many other areas. In these practical problems, performance plays a main role. Therefore, development of high-speed digital signal processing devices is an important problem [9], [10].

Operations parallelization is a common approach to increase the performance of a device. However, in many cases this method leads to an increase in hardware resources [11]. One of the approaches to reduce hardware resources is the Common subexpression elimination (CSE) technique to minimize logical operators and reduce the logical depth [12], [13].

Although the calculation of the filter coefficients according to the given parameters within the device is making the filter more versatile, it requires additional hardware resources [14]. Therefore, it is advisable to calculate the filter coefficients in advance and store them in the device memory. Moreover, when the form of the filter coefficients is known in advance, this allows to optimize device architecture [15].

The main computational load during filtering is the repeated execution of the multiplication operation. It is to reduce the number of multiplications to increase the performance. In [16], the Winograd filtering method was proposed, which reduces the number of multiplications in the filtering process by increasing the number of additions. Another approach is parallel computations. The Residue

The associate editor coordinating the review of this manuscript and approving it for publication was Manuel Rosa-Zurera.

Number System (RNS) is a non-positional number system where numbers are represented are represented as set of reminders by independent co-prime moduli and arithmetic operations can be performed in parallel [17]. The authors of [18] propose a method of constructing digital filters in RNS to automate the device design process and provide an effective ratio of performance and energy efficiency. The article [19] presents a digital filter architecture based on the Winograd method and RNS for a 2 × 2 filter mask. Unfortunately, the case of a 2 × 2 mask considered in the article is rarely used in practice.

Using RNS in real applications faces the problem of implementing computationally complex operations, such as forward and inverse conversion to RNS from positional representation, sign detection, comparison of numbers, and division. Despite the listed problematic operations, RNS allows to increase the speed of calculations, for example, as it shown for convolutional neural networks in [20]. To use advantage of non-positional nature of RNS the Winograd method should be modified accordingly. In this paper, we propose a new approach to the design of the device of digital filters based on RNS and modified Winograd method. Our contribution is summarized in the following list:

- A new modified Winograd method is proposed to increase the performance of two-dimensional digital filters with a 5×5 mask.
- Winograd method 3-modulus RNS with moduli of a special form 2^α and $2^\alpha - 1$ has been merged to increase the performance of the digital filter.
- The architecture of a digital filter device with a 5 × 5 mask has been developed based on the proposed modified Winograd method.
- Performance of the digital filter is theoretically evaluated based on the unit-gate model [21] was made. Theoretical evaluation showed the performance advantage of the proposed architecture compared to known analogues.
- The results of our FPGA simulation show that the proposed filter architecture has a higher fragment throughput by 29.6% - 724.7% compared to analogues.

The proposed device is designed to filter a 2D signal depending on the filter mask being used. It can perform various functions, such as smoothing, noise removal (impulse, Gaussian), sharpening, and edge detection. The main target application of the proposed filter is the hardware accelerators design of convolutional neural networks (CNN), since the 5 × 5 mask is often used in CNN architectures [22], [23].

The rest of the paper is organized as follows. The second section presents the features of digital filtering in RNS. The third section consists of the known Winograd method for two-dimensional filtering. Forth section proposes modification of Winograd method using RNS with moduli 2^α and $2^\alpha - 1$ for design digital filters. Fifth section contains results of theoretical analysis and simulation. Analysis of the research results are presented in the sixth section. The conclusions are presented in the seventh section.

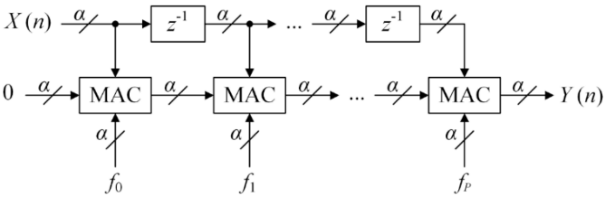


FIGURE 1. Circuit of the α-bit FIR device of order P.

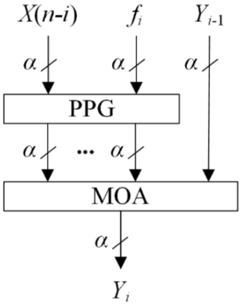


FIGURE 2. Circuit of the α- bit MAC device.

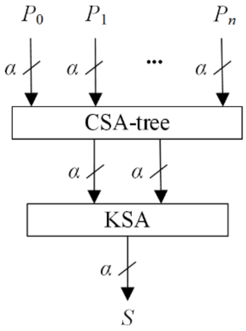


FIGURE 3. Circuit α—bit MOA device.

II. APPLICATION OF RNS FOR DIGITAL FILTERING

The tool for implementing digital signal filtering are digital filters, which are usually divided into filters with a finite impulse response (FIR) and filters with an infinite impulse response (IIR). Based on a sequence of signal samples $X(n)$ a signal $Y(n)$ is formed at the output of the FIR filter defined by the formula:

$$Y(n) = \sum_{i=0}^{P-1} f_i X(n-1), \tag{1}$$

where f_i are filter coefficients and P is a filter order.

Figure 1 shows the FIR filter architecture. The device input receives a sequence of signal samples $X(n)$ and filter coefficients f_i, and the output is a signal $Y(n)$. The multiplication with accumulation operation according to the equation (1) is performed using multiply-accumulate units (MAC) shown in Figure 2. The MAC device consists of a partial product generator (PPG) unit, which is formed from an array of AND gates [24] and a multi-operand adder (MOA).

MOA units can be implemented using a tree of various adders. In this paper we use carry-save adder (CSA) [24], which convert the addition operation of three numbers to

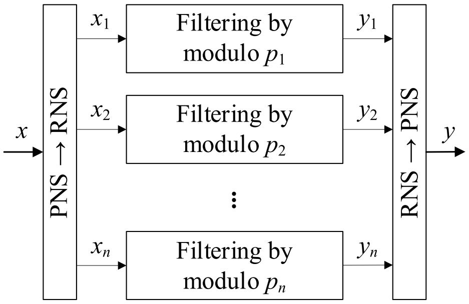

x
PNS → RNS
x_1
x_2
x_n
Filtering by modulo p_1
Filtering by modulo p_2
Filtering by modulo p_n
y_1
y_2
y_n
RNS → PNS
y

FIGURE 4. Digital filtering in RNS.

addition of two numbers. The result of the CSA-tree is added using a Kogge-Stone parallel-prefix adder (KSA) [25].

The MOA device architecture is shown in Figure 3. A sequence of terms $\{P_i\}$ is fed to the input of the device, where $0 \leq i \leq n$ and the output is the sum S.

We use RNS as one of the ways to accelerate computations using parallelism. Any integer $0 \leq A < P$ can be uniquely represented in RNS as residues from division into system modules $A = \{a_1, a_2, \ldots, a_n\}$, where P – is the RNS dynamic range equal to the multiplication of coprime modules $\{p_1, p_2, \ldots, p_n\}$.

Digital filtering according to formula (1) in RNS is performed in several stages (Figure 4). First, it is necessary to convert data from the positional number system (PNS) to RNS. Then, filtering is performed in parallel on several computational channels, which correspond to the RNS moduli. Next, the inverse conversion from RNS to PNS is performed.

The type of RNS moduli affects the performance of calculations. Therefore, their choice is an important problem when designing application systems that use RNS arithmetic. On the one hand, the moduli set must provide a sufficient dynamic range of the system for unambiguous numbers representation in RNS. On the other hand, the moduli must be balanced in such a way that the execution time for each channel is approximately the same and doesn't cause long system downtime for any computing channel. Finally, the RNS moduli of the special form 2^α and $2^\alpha - 1, \alpha \in \mathbb{N}$, where $\mathbb{N}$ stands for the set of natural numbers, make it possible to avoid the resource-consuming operation of modulo division.

We propose to use the modified Winograd method with calculations in RNS with modules of a special form 2^α and $2^\alpha - 1$ to implement digital filtering.

III. WINOGRAD FILTERING METHOD F (2 x 2, 5 x 5)

One-dimensional filtering by the Winograd method can be represented in matrix form as:

$$z = A^T \left((Gw) \odot \left(B^T d \right) \right), \tag{2}$$

where operator $\odot$ denotes element-wise matrix multiplication, A, G and B are transformation matrices, w is one-dimensional filter mask, d is data vector, z is filtering result [26]. The algorithm of one-dimensional filtering according to the Winograd method is usually denoted $F(n, k)$, where n is a vector'sz size, and k is filter mask w size.

Two-dimensional filtering by the Winograd method in matrix form is [26]:

$$Z = A^T \left(\left(GWG^T \right) \odot \left(B^T DB \right) \right) A, \tag{3}$$

where W is a two-dimensional filter mask, D is a two-dimensional data array, and Z is a two-dimensional array of filter result. The two-dimensional filtering algorithm according to the Winograd method is usually denoted $F(n \times n, k \times k)$.

Consider one-dimensional filtering by the Winograd method using the example of the case $F(2, 5)$. We represent the vectors w, d and z as polynomials

$$\begin{aligned} z(x) &= z_1 x + z_0, \\ w(x) &= w_4 x^4 + w_3 x^3 + w_2 x^2 + w_1 x + w_0, \\ d(x) &= d_5 x^5 + d_4 x^4 + d_3 x^3 + d_2 x^2 + d_1 x + d_0. \end{aligned} \tag{4}$$

Then the filtering can be represented as a product of polynomials

$$d(x) = w(x) z(x), \tag{5}$$

Let us introduce a polynomial $m(x)$ of degree 6, and represent $d(x)$ as the remainder modulo $m(x)$

$$d(x) = w(x) z(x) \, mod m(x), \tag{6}$$

If we replace $m(x)$ of degree 6 with a polynomial of degree 5, then

$$d(x) = w(x) z(x) \, mod m(x) + R_{m(x)}[d(x)], \tag{7}$$

where $R_{m(x)}[d(x)]$ is remainder of the division $d(x)$ by $m(x)$.

Let's choose a polynomial $m(x) = m^{(0)}(x) \cdot m^{(1)}(x) \cdot m^{(2)}(x) \cdot m^{(3)}(x) \cdot m^{(4)}(x) \cdot m^{(5)}(x) \cdot m^{(6)}(x) = x(x-1)(x+1)(x-2)(x+2)(x - \frac{1}{2})(x-\infty)$, where $(x-\infty)$ corresponds $R_{m(x)}[d(x)]$. Then the remainder after division $w(x)$ by $m^{(i)}(x)$ are

$$\begin{aligned} w^{(0)}(x) &= w_0, \\ w^{(1)}(x) &= w_0 + w_1 + w_2 + w_3 + w_4, \\ w^{(2)}(x) &= w_0 - w_1 + w_2 - w_3 + w_4, \\ w^{(3)}(x) &= w_0 + 2w_1 + 4w_2 + 8w_3 + 16w_4, \\ w^{(4)}(x) &= w_0 - 2w_1 + 4w_2 - 8w_3 + 16w_4, \\ \mathrm{w}^{(5)}(\mathrm{x}) &= \mathrm{w}_4. \end{aligned} \tag{8}$$

And the remainder from division $z(x)$ by $m^{(i)}(x)$ are

$$\begin{aligned} \mathrm{z}^{(0)}(\mathrm{x}) &= \mathrm{z}_0, \\ \mathrm{z}^{(1)}(\mathrm{x}) &= \mathrm{z}_0 + \mathrm{z}_1, \\ \mathrm{z}^{(2)}(\mathrm{x}) &= \mathrm{z}_0 - \mathrm{z}_1, \\ \mathrm{z}^{(3)}(\mathrm{x}) &= \mathrm{z}_0 + 2\mathrm{z}_1, \\ \mathrm{z}^{(4)}(\mathrm{x}) &= \mathrm{z}_0 - 2\mathrm{z}_1, \\ \mathrm{z}^{(5)}(\mathrm{x}) &= \mathrm{z}_1. \end{aligned} \tag{9}$$

The transformation matrix A is composed of coefficients with remainders after division $z(x)$ by $m^{(i)}(x)$ and has the following form

$$A=\begin{bmatrix}1&0\\1&1\\1&-1\\1&2\\1&-2\\0&1\end{bmatrix}. \tag{10}$$

Let $M^{(i)}(x)=\frac{m(x)}{m^{(i)}(x)}$, then

$$\begin{aligned}\mathrm{m}\left(\mathrm{x}\right)&=\mathrm{x}^5-5\mathrm{x}^3+4\mathrm{x},\\ \mathrm{M}^{(0)}\left(\mathrm{x}\right)&=\mathrm{x}^4-5\mathrm{x}^2+4,\\ \mathrm{M}^{(1)}\left(\mathrm{x}\right)&=\mathrm{x}^4+\mathrm{x}^3-4\mathrm{x}^2-4\mathrm{x},\\ \mathrm{M}^{(2)}\left(\mathrm{x}\right)&=\mathrm{x}^4-\mathrm{x}^3-4\mathrm{x}^2+4\mathrm{x},\\ \mathrm{M}^{(3)}\left(\mathrm{x}\right)&=\mathrm{x}^4+2\mathrm{x}^3-\mathrm{x}^2-2\mathrm{x},\\ \mathrm{M}^{(4)}\left(\mathrm{x}\right)&=\mathrm{x}^4-2\mathrm{x}^3-\mathrm{x}^2+2\mathrm{x}.\end{aligned} \tag{11}$$

The transformation matrix B is composed of the coefficients of the polynomials $M^{(i)}(x)$ and $m(x)$, and coefficients $M^{(i)}(x)$ correspond to the i-th column of the matrix B

$$B=\begin{bmatrix}4&0&0&0&0&0\\0&-4&4&-2&2&4\\-5&-4&-4&-1&-1&0\\0&1&-1&2&-2&-5\\1&1&1&1&1&0\\0&0&0&0&0&1\end{bmatrix}. \tag{12}$$

Using the extended Euclidean algorithm, we compute $h^{(i)}(x)$ and $H^{(i)}(x)$ such that $h^{(i)}\left(x\right)m^{(i)}\left(x\right)+H^{(i)}\left(x\right)M^{(i)}\left(x\right)=1$ [16]:

$$\begin{aligned}&\mathrm{h}^{(0)}\left(\mathrm{x}\right)=-\frac{1}{4}\mathrm{x}^3+\frac{5}{4}\mathrm{x},\quad \mathrm{H}^{(0)}\left(\mathrm{x}\right)=\frac{1}{4};\\ &\mathrm{h}^{(1)}\left(\mathrm{x}\right)=\frac{1}{6}\mathrm{x}^3+\frac{1}{3}\mathrm{x}^2-\frac{1}{3}\mathrm{x}-1,\quad \mathrm{H}^{(1)}\left(\mathrm{x}\right)=-\frac{1}{6};\\ &\mathrm{h}^{(2)}\left(\mathrm{x}\right)=\frac{1}{6}\mathrm{x}^3-\frac{1}{3}\mathrm{x}^2-\frac{1}{3}\mathrm{x}+1,\quad \mathrm{H}^{(2)}\left(\mathrm{x}\right)=-\frac{1}{6};\\ &\mathrm{h}^{(3)}\left(\mathrm{x}\right)=-\frac{1}{24}\mathrm{x}^3-\frac{1}{6}\mathrm{x}^2-\frac{7}{24}\mathrm{x}-\frac{1}{2},\quad \mathrm{H}^{(3)}\left(\mathrm{x}\right)=\frac{1}{24};\\ &\mathrm{h}^{(4)}\left(\mathrm{x}\right)=-\frac{1}{24}\mathrm{x}^3+\frac{1}{6}\mathrm{x}^2-\frac{7}{24}\mathrm{x}+\frac{1}{2},\quad \mathrm{H}^{(4)}\left(\mathrm{x}\right)=\frac{1}{24}.\end{aligned} \tag{13}$$

The transformation matrix G is composed of the coefficients of the division residues $w^{(i)}(x)$ (x) multiplied by $H^{(i)}(x)$

$$G=\begin{bmatrix}\frac{1}{4}&0&0&0&0\\-\frac{1}{6}&-\frac{1}{6}&-\frac{1}{6}&\frac{1}{6}&-\frac{1}{6}\\-\frac{1}{6}&-\frac{1}{6}&-\frac{1}{6}&\frac{1}{6}&-\frac{1}{6}\\\frac{1}{24}&\frac{1}{12}&\frac{1}{6}&\frac{1}{3}&\frac{2}{3}\\\frac{1}{24}&-\frac{1}{12}&\frac{1}{6}&-\frac{1}{3}&\frac{2}{3}\\0&0&0&0&1\end{bmatrix}. \tag{14}$$

For two-dimensional filtering $F\left(2\times 2,5\times 5\right)$ calculations are made according to formula (3). Next, the device architectures of two-dimensional filtering by the Winograd method $F\left(2\times 2,5\times 5\right)$ with calculations in RNS are presented.

IV. THE FILTER ARCHITECTURE ACCORDING TO THE MODIFIED WINOGRAD METHOD F (2 × 2, 5 × 5) IN THE RESIDUE NUMBER SYSTEM

A new filtering method based on the Winograd method based on RNS with moduli of a special form 2^α and $2^\alpha-1$ is proposed to increase the performance of digital filtering.

Let's divide a two-dimensional signal into fragments D with size $m\times m$, $m>k$. Each fragment is processed by a $k\times k$ filter w using the Winograd method $F\left(n\times n,k\times k\right)$ with step n for each dimension. In the case of $F\left(2\times 2,5\times 5\right)$ the two-dimensional signal is divided into 6×6 fragments and the processing is performed with a step of 2, the result of filtering one fragment D is a filtered 2×2 fragment Z. Figure 5a shows filtering process of a 256 × 256 2D signal with a 5×5 filter mask using Winograd method $F\left(2\times 2,5\times 5\right)$.

Performing filtering with $k\times k$ mask in traditional way requires k^2 multiplications. Then n^2k^2 multiplications are required to form an $n\times n$ filtered fragment. Winograd method $F\left(n\times n,k\times k\right)$ requires $(n+k-1)^2$ multiplications [26]. Then to filter a 6×6 signal fragment with a 5×5 filter mask it is necessary to perform 900 multiplication operations. Using Winograd method $F\left(2\times 2,5\times 5\right)$ allows to reduce number of multiplications to 36, that is, the computational complexity is reduced by 25 times.

The procedure of two-dimensional filtering according to the Winograd method described by formula (3), processes the signal in several stages. Let's denote the result of the filter mask transformation is denoted as $U=GWG^T$. Since the filter coefficients are constants, this transformation can be performed once in advance, which means it does not carry a computational load. Let's designate the transformation result of input data D as $V=B^TDB$, and the result of element-wise matrix multiplication as $M=U\odot V$. Then, considering the introduced notations, formula (3) becomes $Z=A^TMA$. Figure 5b shows filtering process of signal fragment D according to the Winograd method using the case $F\left(2\times 2,5\times 5\right)$ as an example.

The addition of several numbers modulo 2^α and $2^\alpha-1$ is proposed to be performed using a multi-operand modulo adder, denote them as $MOMA_{2^\alpha}$ and $MOMA_{2^\alpha-1}$ respectively (Figure 6). These devices consist of a CSA tree and a KSA. The vector $\mathrm{P}=\left\{\mathrm{P}_0,\mathrm{P}_1,\ldots,\mathrm{P}_\beta\right\}$, comes to the input of the devices and the sum S is formed at the output. For calculations modulo $2^\alpha-1$ End-Around-Carry (EAC) technique is used [27].

Data transformations modulo 2^α are performed using the devices DTE_{2^α} (data transform element), shown in Figure 7a. The device input is the vector $\{P_i\}$, where $0\leq i<l$. Since negative numbers modulo 2^α are represented in a two's complement code, a correction constant C, is introduced, equal to the number of vector $\{P_i\}$ negative elements. SL (shift left) blocks perform a left shift by n bits, which corresponds to a multiplication by 2^n. Next, addition is performed using the CSA adder tree. Data conversion modulo $2^\alpha-1$ is performed using the device $DTE_{2^\alpha-1}$ (Figure 7b) differs in that the technique of cyclic transfer of EAC high bits is used, and SLA devices (shift left around) perform cyclic shift by n bits.

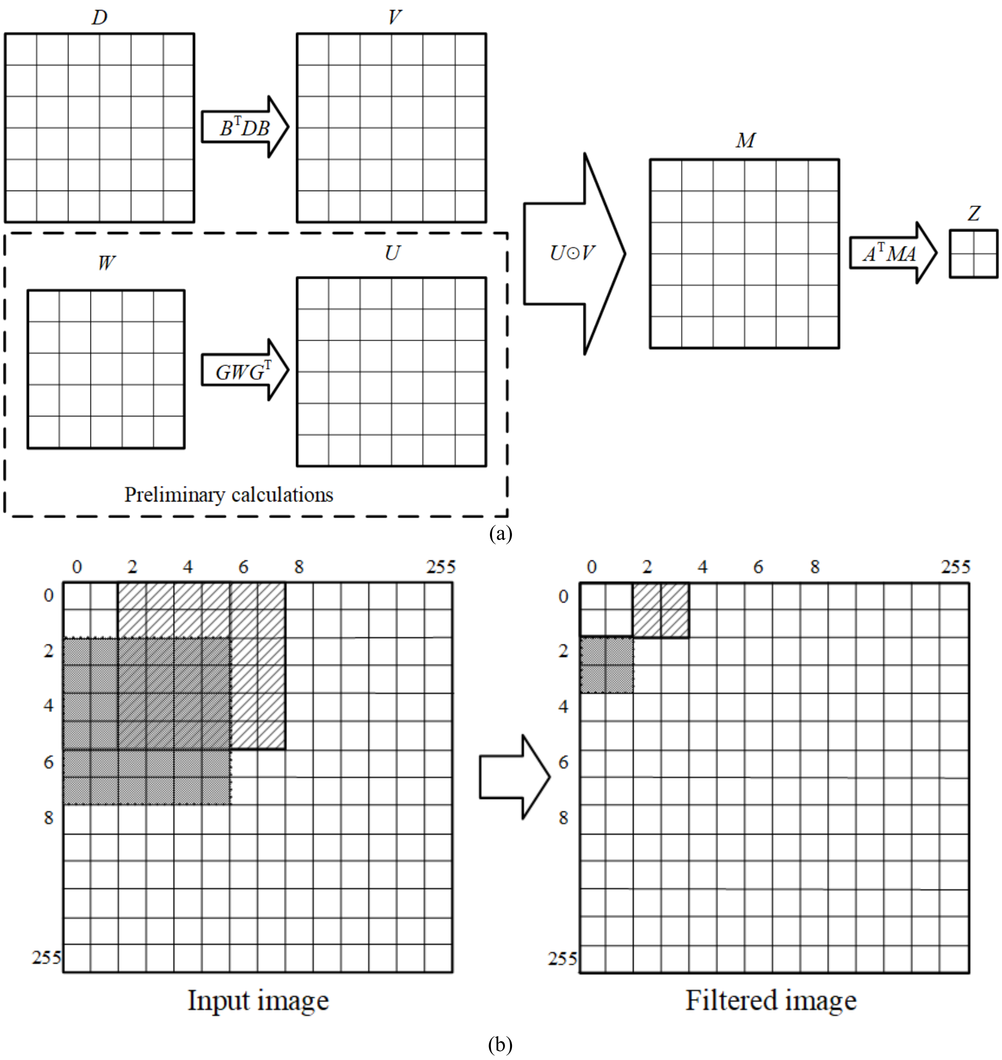


FIGURE 5. Filtering scheme according to the Winograd method F (2×2, 5×5) of a 256 × 256 2D signal (a) a scheme for dividing a 256 × 256 2D signal into 6 × 6 fragments D with a step of 2; (b) filtering scheme of 6 × 6 fragment D according to the Winograd F (2×2, 5×5) method.

Since negative numbers modulo $2^\alpha - 1$ are represented in the one's complement code, then adding a correcting constant is not required.

The calculation of one matrix V row elements modulo 2^α is performed by the DTR_{2^α} device (Figure 8). This device performs data transformation from matrix D and generates elements $V_{i,j}$, $0 \leq i \leq 5$, $0 \leq j \leq 5$. To calculate the elements of the i-th row of the matrix V modulo 2^α, the input DTR_{2^α} is supplied with the data vector $D^i = \{D_0^i, D_1^i, \ldots, D_{23}^i\}$, correction coefficient $C = 2$ and shift vector $N^i = \{N_0^i, N_1^i, N_2^i\}$. For example, to calculate the elements of the row $V_{0,j}$, data vector is supplied to the input $D^0 = \{d_{0,0}, \overline{d_{2,0}}, \overline{d_{2,0}}, d_{4,0}, d_{0,1}, \overline{d_{2,1}}, \overline{d_{2,1}}, d_{4,1}, d_{0,2}, \overline{d_{2,2}}, \overline{d_{2,2}}, d_{4,2}, d_{0,3}, \overline{d_{2,3}}, d_{4,3}, d_{0,4}, \overline{d_{2,4}}, \overline{d_{2,4}}, d_{4,4}, d_{0,5}, \overline{d_{2,5}}, \overline{d_{2,5}}, d_{4,5}\}$, correction coefficient $C = 2$ shift vector $N^0 = \{1, 2, 2\}$. The input data goes to the DTE_{2^α} data conversion devices, the result is added using $MOMA_{2^\alpha}$ adders. Thus, the data conversion device modulo 2^α (let's denote it as DT_{2^α}) consists of 6 DTR_{2^α} devices.

Calculation of the matrix V elements modulo $2^\alpha - 1$ requires the representation of negative numbers in the one's complement code, that is, the inversion of the number, therefore, the correction constants are not involved in the calculations. Therefore, the device for data transformation modulo $2^\alpha - 1$ (let's denote it as $DT_{2^\alpha-1}$) consists of 6 $DTR_{2^\alpha-1}$ devices (shown in Figure 8) the inputs of which are vectors D^i and N^i.

Element-wise multiplication of matrices U and V is performed using devices EWM_{2^α} and $EWM_{2^\alpha-1}$, consisting of 36 parallel multipliers of two numbers MUL_{2^α} and $MUL_{2^\alpha-1}$ respectively, and shown in Figure 10. Elements of the matrices U and V are supplied to the input of the device. The multiplier MUL_{2^α} consists of a partial product generator modulo 2^α PPG_{2^α}, which is formed from an array of AND gates [21] and $MOMA_{2^\alpha}$. The $MUL_{2^\alpha-1}$ device consists of a partial product generator modulo $2^\alpha - 1$ $PPG_{2^\alpha-1}$, using the EAC technique, and $MOMA_{2^\alpha-1}$. Thus, a 6 × 6 matrix M is formed.

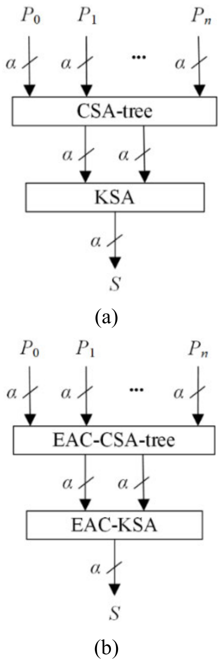


FIGURE 6. Circuit of multi-operand modulo adder MOMA: (a) modulo 2^α ($MOMA_{2^\alpha}$); (b) modulo $2^\alpha - 1$ ($MOMA_{2^\alpha-1}$).

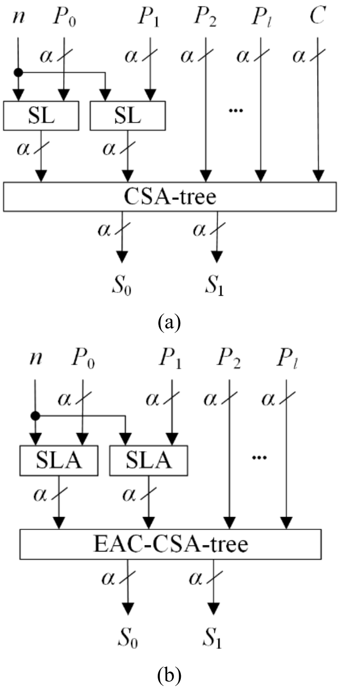


FIGURE 7. DTE data conversion device: (a) modulo 2^α; (b) modulo $2^\alpha - 1$.

The calculation of the matrix Z one row elements modulo 2^α is performed by the FTR_{2^α} deivce (Figure 11). This device performs the final data transformation from the matrix M and generates elements $Z_{i,j}$, $0 \leq i \leq 1$, $0 \leq j \leq 1$. To calculate the elements of the i-th row of the matrix Z modulo 2^α, to the input of FTR_{2^α} supplied data vector $R^i = \{R_0^i, R_1^i, \ldots, R_{29}^i\}$, correction coefficients $C^i = \{C_0^i, C_1^i, C_2^i\}$ and offset vector $N^i = \{N_0^i, N_1^i, N_2^i\}$. For example, to calculate the elements of the row $Z_{0,j}$, to the input supplied data vector $R^0 = \{M_{0,0}, M_{1,0}, M_{2,0}, M_{3,0}, M_{4,0}, M_{0,1}, M_{1,3}, M_{1,1}, M_{2,1}, M_{3,1}, M_{4,1}, M_{0,2}, M_{1,2}, M_{2,2}, M_{3,2}, M_{4,2}, M_{0,3}, M_{2,3}, M_{3,3}, M_{4,3}, M_{0,4}, M_{1,4}, M_{2,4}, M_{3,4}, M_{4,4}, M_{0,5}, M_{1,5}, M_{2,5}, M_{3,5}, M_{4,5}\}$, correction coefficients $C^0 = \{0, 0, 2\}$ and offset vector $N^0 = \{1, 0, 0\}$. The input data goes to the DTE_{2^α} data conversion devices, the result is added using $MOMA_{2^\alpha}$ adders. Thus, the device for data transformation modulo 2^α (let's denote it as FT_{2^α}) consists of 2 FTR_{2^α} devices.

Calculation of the matrix Z elements modulo $2^\alpha - 1$ requires the representation of negative numbers in the inverse code, therefore, the correction constants do not participate in the calculations. Therefore, the device for data transformation modulo $2^\alpha - 1$ (let's denote it as $FT_{2^\alpha-1}$) consists of 2 $FTR_{2^\alpha-1}$ devices (shown in Figure 8) the inputs of which are vectors R^i and N^i.

Figure 13 shows the proposed filtering device $F\,(2 \times 2, 5 \times 5)_{2^\alpha}$ modulo 2^α. The $U = GWG^T$ filter mask transformation is done preliminarily, and the result is stored in the device memory. Since operations with negative numbers require their presentation in two's complement code, the correction constants are also stored in the device memory. Figure 14 shows a circuit of the proposed filtering device $F\,(2 \times 2, 5 \times 5)_{2^\alpha-1}$ by modulo $2^\alpha - 1$. Only the converted filter mask is stored in the memory of this device.

V. EVALUATION

For a theoretical assessment of the delay and area of a digital device, we use the "unit-gate" model [21]. The proposed filters based on the Winograd method $F\,(2 \times 2, 5 \times 5)$ and RNS with moduli of a special type consist of the DTR, MUL and FTR devices described above. The DTR device consists of DTE and MOMA blocks. Device parameters DTE_{2^α} (Figure 7a) are calculated by the formula:

$$U_{delay}\,(DTE_{2^\alpha}) = 6,8log_2\,(l+1)\,,$$
$$U_{area}\,(DTE_{2^\alpha}) = 7\alpha l - 7\alpha, \qquad (15)$$

and device $DTE_{2^\alpha-1}$ (Figure 7b) by the formula:

$$U_{delay}\,(DTE_{2^\alpha-1}) = 6,8log_2 l,$$
$$U_{area}\,(DTE_{2^\alpha-1}) = 7\alpha l - 14\alpha \qquad (16)$$

The parameters of the adder $MOMA_{2^\alpha}$ have the form, where N – is the number of terms [28]:

$$U_{delay}\,(MOMA_{2^\alpha}) = 6,8log_2 N + 2log_2\alpha + 4,$$
$$U_{area}\,(MOMA_{2^\alpha}) = 3\alpha log_2\alpha + 7\alpha N - 11\alpha + 1. \qquad (17)$$

Similarly, the parameters of the adder $MOMA_{2^\alpha-1}$ can be represented as [28]:

$$U_{delay}\,(MOMA_{2^\alpha-1}) = 6,8log_2 N + 2log_2\alpha + 4,$$
$$U_{area}\,(MOMA_{2^\alpha-1}) = 3\alpha log_2\alpha + 7\alpha N - 8\alpha. \qquad (18)$$

The MUL multiplier consists of a partial product generator PPG and a MOMA block with α inputs. The partial product generator PPG_{2^α} has the following delay and area parameters:

$$U_{delay}\,(PPG_{2^\alpha}) = 0,5\alpha^2{+}0,5\alpha,$$
$$U_{area}\,(PPG_{2^\alpha}) = 0,5\alpha^2 + 0,5\alpha. \qquad (19)$$

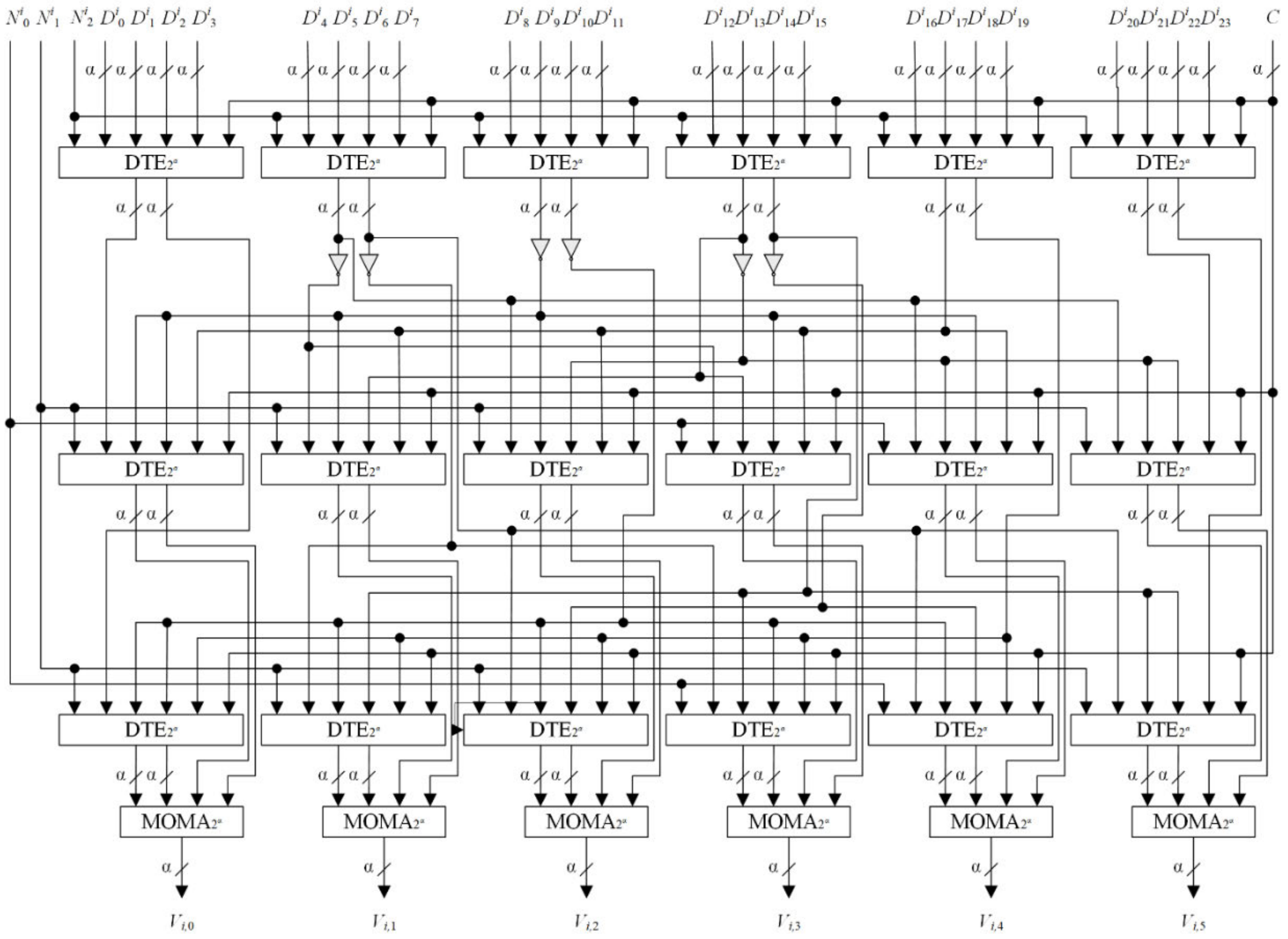

N^i_0 N^i_1 N^i_2 D^i_0 D^i_1 D^i_2 D^i_3
D^i_4 D^i_5 D^i_6 D^i_7
D^i_8 D^i_9 D^i_{10} D^i_{11}
D^i_{12} D^i_{13} D^i_{14} D^i_{15}
D^i_{16} D^i_{17} D^i_{18} D^i_{19}
D^i_{20} D^i_{21} D^i_{22} D^i_{23}
C
α
DTE_{2^α}
$MOMA_{2^\alpha}$
$V_{i,0}$
$V_{i,1}$
$V_{i,2}$
$V_{i,3}$
$V_{i,4}$
$V_{i,5}$

FIGURE 8. ***DTR_{2^α}*** **device for calculating the elements of the *i*-th row of the matrix *V* modulo2^α.**

The partial product generator $PPG_{2^\alpha-1}$ has the following parameters:

$$U_{delay}\left(PPG_{2^\alpha-1}\right) = \alpha^2,$$
$$U_{area}\left(PPG_{2^\alpha-1}\right) = \alpha^2. \qquad (20)$$

Then, considering (17) and (19), the multiplier MUL_{2^α} has the following parameters:

$$U_{delay}\left(MUL_{2^\alpha}\right) = 8, 8log_2\alpha + 0, 5\alpha^2 + 0, 5\alpha + 4,$$
$$U_{area}\left(MUL_{2^\alpha}\right) = 3\alpha log_2\alpha + 7, 5\alpha^2 - 10, 5\alpha + 1. \qquad (21)$$

And the multiplier $MUL_{2^\alpha-1}$, based on (18) and (20) has the following parameters:

$$U_{delay}\left(MUL_{2^\alpha-1}\right) = 8, 8log_2\alpha + \alpha^2 + 4,$$
$$U_{area}\left(MUL_{2^\alpha-1}\right) = 3\alpha log_2\alpha + 8\alpha^2 - 8\alpha. \qquad (22)$$

The DTR_{2^α} device parameters (Figure 8) can be calculated using the formulas:

$$\begin{aligned}U_{delay}\left(DTR_{2^\alpha}\right) &= 2U_{delay}\left(DTE_{2^\alpha}\right)\\ &\quad + U_{delay}\left(MOMA_{2^\alpha}\right)\\ &= 2log_2\alpha + 58, 4,\end{aligned}$$

$$\begin{aligned}U_{area}\left(DTR_{2^\alpha}\right) &= 18U_{area}\left(DTE_{2^\alpha}\right)\\ &\quad + 6U_{area}\left(MOMA_{2^\alpha}\right)\\ &= 18\alpha \log_2 \alpha + 480\alpha + 6, \qquad (23)\end{aligned}$$

and device $DTR_{2^\alpha-1}$ by formulas

$$\begin{aligned}U_{delay}\left(DTR_{2^\alpha-1}\right) &= 2U_{delay}\left(DTE_{2^\alpha-1}\right)\\ &\quad + U_{delay}\left(MOMA_{2^\alpha-1}\right) = 2log_2\alpha + 44, 8,\\ U_{area}\left(DTR_{2^\alpha-1}\right) &= 18U_{area}\left(DTE_{2^\alpha-1}\right)\\ &\quad + 6U_{area}\left(MOMA_{2^\alpha-1}\right)\\ &= 18\alpha \log_2 \alpha + 372\alpha. \qquad (24)\end{aligned}$$

The parameters of the MUL device for numbers multiplication are calculated using formulas (21) and (22). FTR device parameters modulo 2^α are

$$\begin{aligned}U_{delay}\left(FTR_{2^\alpha}\right) &= 2U_{delay}\left(DTE_{2^\alpha}\right)\\ &\quad + U_{delay}\left(MOMA_{2^\alpha}\right)\\ &= 2log_2\alpha + 58, 4,\\ U_{area}\left(FTR_{2^\alpha}\right) &= 10U_{area}\left(DTE_{2^\alpha}\right)\\ &\quad + 2U_{area}\left(MOMA_{2^\alpha}\right)\\ &= 6\alpha \log_2 \alpha + 297\alpha + 2, \qquad (25)\end{aligned}$$

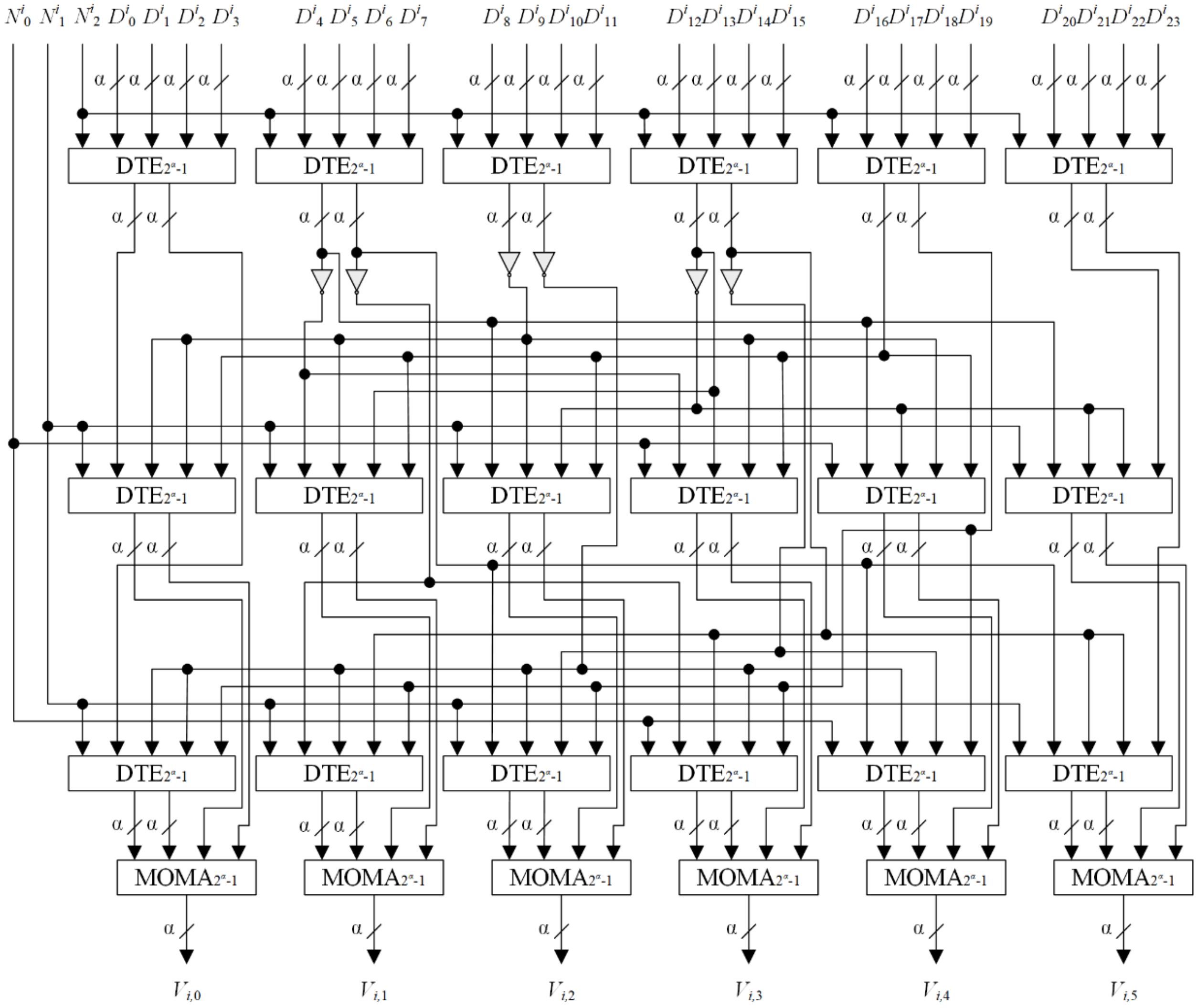

N^i_0 N^i_1 N^i_2 D^i_0 D^i_1 D^i_2 D^i_3
D^i_4 D^i_5 D^i_6 D^i_7
D^i_8 D^i_9 D^i_{10} D^i_{11}
D^i_{12} D^i_{13} D^i_{14} D^i_{15}
D^i_{16} D^i_{17} D^i_{18} D^i_{19}
D^i_{20} D^i_{21} D^i_{22} D^i_{23}
α
$DTE_{2^\alpha-1}$
$MOMA_{2^\alpha-1}$
$V_{i,0}$
$V_{i,1}$
$V_{i,2}$
$V_{i,3}$
$V_{i,4}$
$V_{i,5}$

FIGURE 9. $DTR_{2^\alpha-1}$ **device for calculating the elements of the *i*-th row of the matrix *V* modulo $2^\alpha - 1$.**

and modulo $2^\alpha - 1$ are

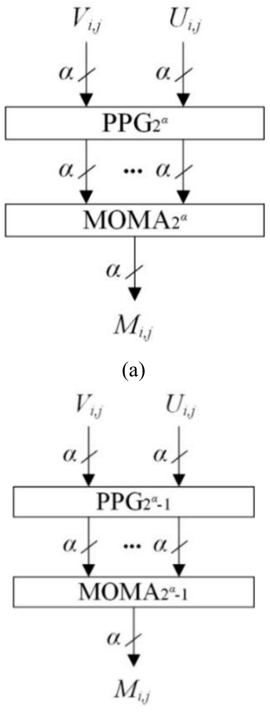
$$\begin{aligned} U_{delay}\left(FTR_{2^\alpha-1}\right) &= 2U_{delay}\left(DTE_{2^\alpha-1}\right) \\ &\quad + U_{delay}\left(MOMA_{2^\alpha-1}\right) \\ &= 2log_2\alpha + 58,4, \\ U_{area}\left(FTR_{2^\alpha-1}\right) &= 10U_{area}\left(DTE_{2^\alpha-1}\right) \\ &\quad + 2U_{area}\left(MOMA_{2^\alpha-1}\right) \\ &= 6\alpha\log_2\alpha + 250\alpha. \end{aligned} \tag{26}$$

The proposed Winograd filtering devices $F(2 \times 2, 5 \times 5)$ consist of 6 DTR devices, 36 MUL devices and 2 FTR devices. Thus, the parameters of the proposed device modulo 2^α, based on (21), (23) and (25), are

$$\begin{aligned} U_{delay}\left(F\left(2\times 2, 5\times 5\right)_{2^\alpha}\right) &= U_{delay}\left(DTR_{2^\alpha}\right) \\ &\quad + U_{delay}\left(MUL_{2^\alpha}\right) \\ &\quad + U_{delay}\left(FTR_{2^\alpha}\right) \\ &= 12,8log_2\alpha + 0,5\alpha^2 \\ &\quad + 0,5\alpha + 120,8, \\ U_{area}\left(F\left(2\times 2, 5\times 5\right)_{2^\alpha}\right) &= 6U_{area}\left(DTR_{2^\alpha}\right) \\ &\quad + 36U_{area}\left(MUL_{2^\alpha}\right) \\ &\quad + 2U_{area}\left(FTR_{2^\alpha}\right) \\ &= 228\alpha\log_2\alpha + 270\alpha^2 \\ &\quad + 3096\alpha + 76, \end{aligned} \tag{27}$$

and modulo $2^\alpha - 1$, based on (22), (24) and (26) are

$$\begin{aligned} U_{delay}\left(F\left(2\times 2, 5\times 5\right)_{2^\alpha-1}\right) &= U_{delay}\left(DTR_{2^\alpha-1}\right) \\ &\quad + U_{delay}\left(MUL_{2^\alpha-1}\right) \\ &\quad + U_{delay}\left(FTR_{2^\alpha-1}\right) \\ &= 12,8log_2\alpha + \alpha^2 + 107,2, \\ U_{area}\left(F\left(2\times 2, 5\times 5\right)_{2^\alpha-1}\right) &= 6U_{area}\left(DTR_{2^\alpha-1}\right) \\ &\quad + 36U_{area}\left(MUL_{2^\alpha-1}\right) \\ &\quad + 2U_{area}\left(FTR_{2^\alpha-1}\right) \\ &= 228\alpha\log_2\alpha \\ &\quad + 288\alpha^2 + 2552\alpha. \end{aligned} \tag{28}$$



(a)

(b)

FIGURE 10. MUL multiplier circuit: (a) modulo 2^{α} ($MUL_{2^{\alpha}}$); (b) modulo $2^{\alpha}-1$ ($MUL_{2^{\alpha}-1}$).

TABLE 1. Units for magnetic properties.

Filter bit depth, bit	RNS moduli set	RNS dynamic range
8	$\{2^5, 2^3-1, 2^2-1\}$	672
16	$\{2^8, 2^5-1, 2^4-1\}$	119040
32	$\{2^{12}, 2^{11}-1, 2^{10}-1\}$	8577355776

The proposed signal filtering device was compared with a device for filtering by the Winograd method without using RNS [29]. In addition, a comparison was made with a filtering device consisting of multiply-accumulation units (MAC) [30], and a device consisting of truncated multiply-accumulation units (TMAC) without RNS arithmetic [31] and with RNS [28]. To implement calculations in RNS, moduli set of a special type were used, presented in Table 1. We consider only 3-modulus RNS cases, although the proposed method can be applied to RNS with more modules. The choice of RNS modules depends on the specific practical task, while considering required dynamic range of the system and the fact that bigger number of moduli leads to higher complexity of the inverse conversion to PNS.

FIR filters based on MAC units [30] denotes as $FIR\,(MAC)$, then for a filter of order P, the device area and delay parameters are calculated as follows:

$$U_{delay}\,(FIR\,(MAC)) = 8,8P\log_2\alpha{+}8,\,8\log_2\alpha + 5P{+}5,$$

$$\begin{aligned} U_{area}\,(FIR\,(MAC)) &= 3\alpha P log_2\alpha + 3\alpha log_2\alpha \\ &\quad + 8\alpha^2 P + 8\alpha^2 - 20\alpha P - 4\alpha + P + 1. \end{aligned} \tag{29}$$

For filters based on TMAC blocks $FIR\,(TMAC)$, the filter of order parameters P are calculated as follows:

$$\begin{aligned} U_{delay}\,(FIR\,(TMAC)) &= 6,8Plog_2\alpha{+}8,\,8log_2\alpha + P{+}5, \\ U_{area}\,(FIR\,(TMAC)) &= 3\alpha log_2\alpha + 8\alpha^2 P + 8\alpha^2 + 3\alpha + 1. \end{aligned} \tag{30}$$

The parameters of a device based on TMAC blocks in RNS that performs calculations modulo 2^{α}, are calculated by (30) and for devices modulo $2^{\alpha}-1$ as follows [28]:

$$\begin{aligned} U_{delay}\left(FIR\,(TMAC)_{2^{\alpha}-1}\right) &= 6,8Plog_2\alpha \\ &\quad + 8,\,8log_2\alpha + P + 5, \\ U_{area}\left(FIR\,(TMAC)_{2^{\alpha}-1}\right) &= 3\alpha log_2\alpha + 8\alpha^2 P \\ &\quad + 8\alpha^2 + 6\alpha. \end{aligned} \tag{31}$$

Table 2 presents the results of the theoretical evaluation of the area and delay parameters of the proposed and known filters based on the "unit-gate" model. The processing time for a frame sized 256 × 256 was also estimated.

Hardware simulation on FPGA was carried out in the Xilinx Vivado 2018.3 CAD environment for the Virtex UltraScale xcvu440-flgb2377-3-e target board with the Flow_PerfOptimized_high optimization strategy. The proposed architecture is not tied to a specific board and can be synthesized on other target devices. Calculations are made in fixed point format. 8-, 16-, and 32-bit filters were considered. For devices with calculations in RNS with modules of a special type, the capacity of each computing channel corresponds to the degree of the module.

The hardware simulation results are presented in Table 3. To evaluate devices, parameters such as clock frequency, number of LUTs, power consumption, and performance were used, which were obtained as a result of simulation in the design environment. Device fragment throughput refers to the number of processed frames with size of 256 × 256 pixels per second.

VI. DISCUSSION

The theoretical analysis of the parameters of the proposed and known two-dimensional filters with a 5 × 5 mask showed that the use of the proposed approach based on the Winograd method and RNS with moduli of a special type reduces the device delay by 15.3% – 81.3%, and the signal processing time also decreases by 15.3% – 95.3%, compared with known approaches. In addition, the device based on the proposed method has a 9.66% – 46.76% smaller area compared to the device based on the Winograd method [29]. Nevertheless, the application of the proposed approach increases the area of the device by 2.7% – 437% in comparison with other considered known methods.

The results of hardware simulation showed that the proposed method of constructing filters based on the Winograd method and RNS allows to increase the clock frequency of 16-bit and 32-bit devices by 29.63% and 38.24%, respectively, compared to the filter based on the Winograd method [29] without using RNS arithmetic. But for 8-bit devices, the filter clock based on the proposed method is

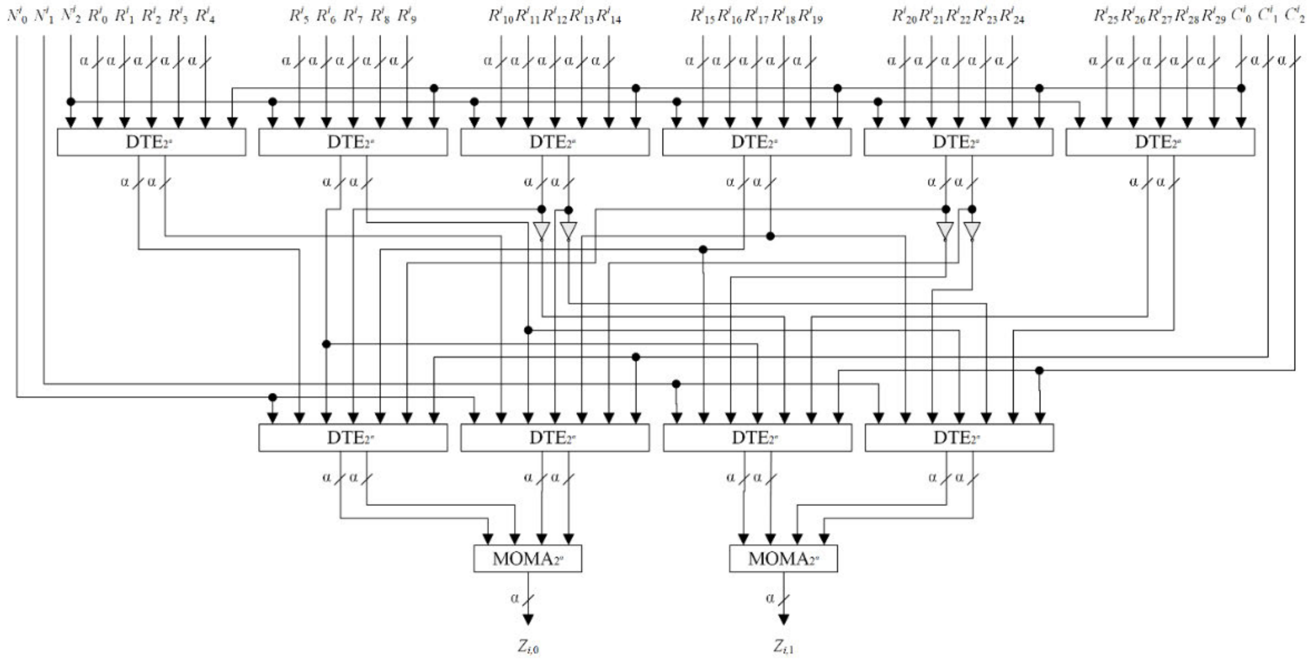


FIGURE 11. $FTR_{2\alpha}$ **device for calculating the elements of the i-th row of the matrix** Z **modulo** 2^{α}**.**

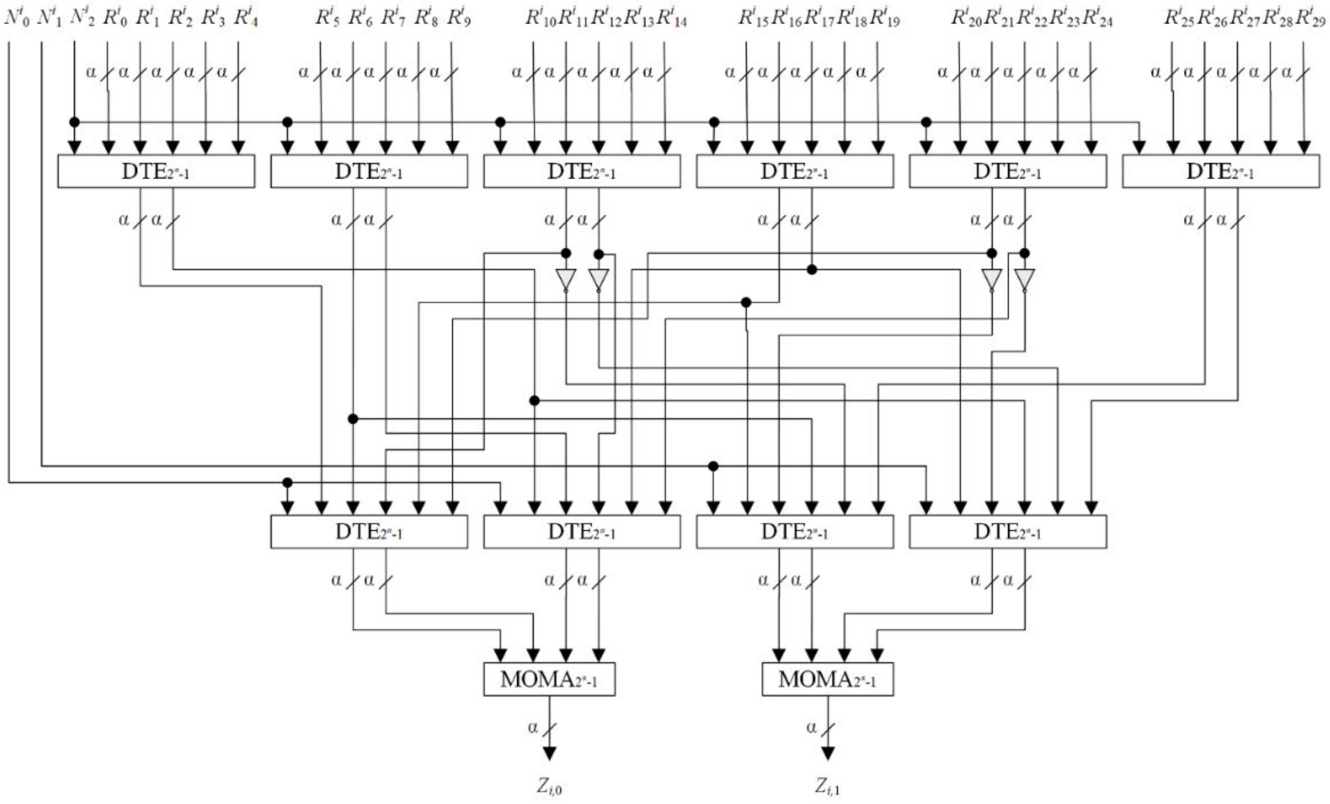


FIGURE 12. $FTR_{2\alpha-1}$ **device for calculating the elements of the i-th row of the matrix** Z **modulo** $2^{\alpha}-1$**.**

3.23% lower. In addition, the proposed method increases the clock frequency of the device by 7.14% - 105.88% compared to methods based on FIR filters with MAC and TMAC blocks [28], [30], [31].

The combined use of RNS and the Winograd method allows to reduce the number of LUTs by 9.50% - 28.17%, and energy consumption by 0.49% - 4.14%, compared with the Winograd method in the PNS. However,

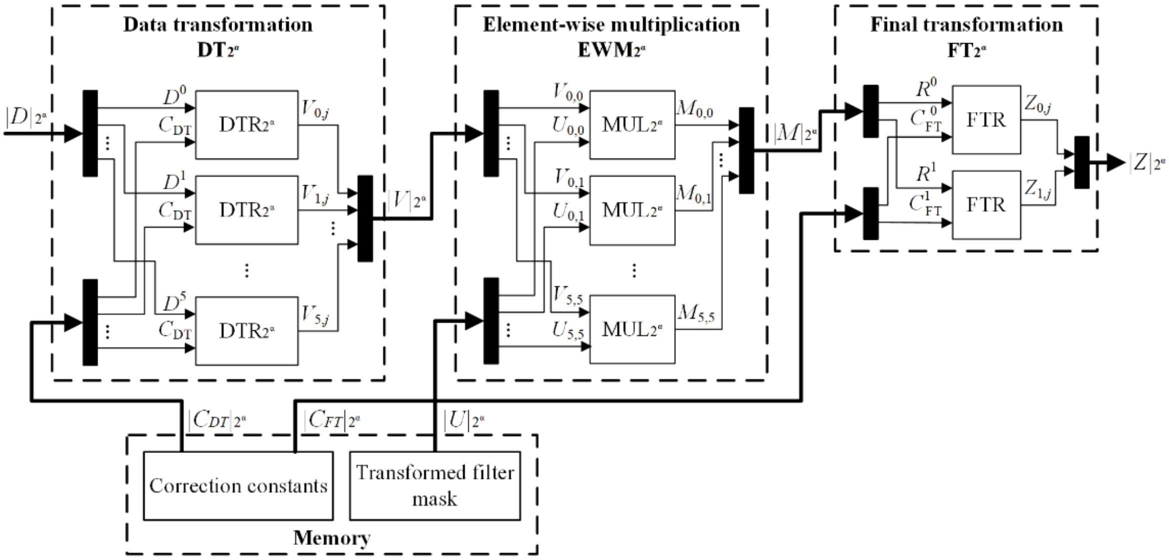


FIGURE 13. Two-dimensional filtering device $F\ (2\times2,\ 5\times5)_{2^\alpha}$ by the Winograd method with calculations by modulo 2^α.

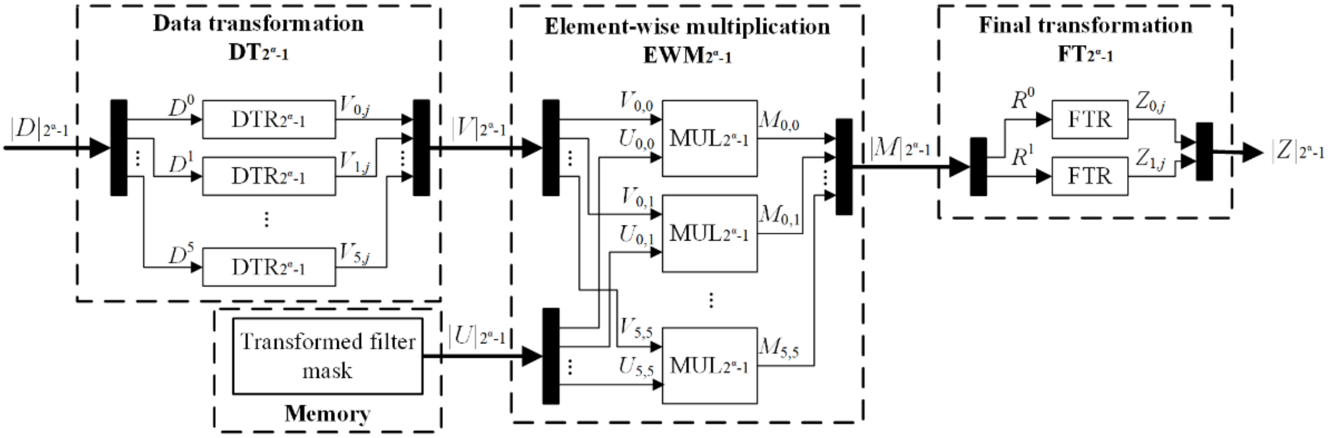


FIGURE 14. Two-dimensional filtering device $F\ (2\times2,\ 5\times5)_{2^\alpha-1}$ by the Winograd method with calculations by modulo $2^\alpha-1$.

TABLE 2. Theoretical parameters of two-dimensional filters with 5 × 5 mask.

Parameter	Filter bit width, bit	Proposed architecture	Methods			
			[28]	[29]	[30]	[31]
Delay	8	**166**	446	196	817	567
	16	**196**	567	308	1046	746
	32	**273**	671	713	1274	924
Area	8	42998	**8006**	47596	11178	13409
	16	86843	**22050**	133324	50202	53489
	32	219408	**76428**	412108	209370	213569
Processing time	8	**2719744**	29229056	3211264	53542912	37158912
	16	**3211264**	38465280	5046272	68550656	48889856
	32	**4472832**	43974656	11681792	83492864	60555264

devices designed according to the proposed method use 2.54% - 11.01% more LUTs and have 3.58% - 19.83% higher power consumption compared to devices based on methods [28], [30], [31].

Hardware simulation showed that the use of the proposed method based on the Winograd and RNS method increases the filter fragment throughput by 29.6% - 724.7% compared to filters based on the considered known methods. However, the 8-bit device based on the Winograd method and PNS has a performance improvement of 3.22% compared to the device based on the proposed method. The slight difference between the results of theoretical analysis and the results of hardware simulation is explained by the peculiarity of the "unit-gate" model, which does not consider the load capacity of the device output, as well as the involved memory and the time of accessing it.

TABLE 3. Parameters of hardware simulation of two-dimensional filters with a 5 × 5 mask.

Parameter	Filter bit width, bit	Proposed architecture	Methods			
			[28]	[29]	[30]	[31]
Clock frequency, MHz	8	60	60	**62**	49	56
	16	**47**	38	34	29	37
	32	**35**	30	27	17	26
Number of LUTs	8	5199	**472**	5745	510	589
	16	13456	**1278**	18144	2916	2417
	32	45920	**6044**	63927	18059	13872
Power consumption, W	8	3.041	**2.928**	3.056	2.936	2.930
	16	3.160	2.949	**3.168**	2.948	2.951
	32	3.589	**2.977**	3.744	2.995	3.012
Fragment throughput, fr/s	8	3662	915	**3784**	747	854
	16	**2868**	579	2075	442	564
	32	**2136**	457	1647	259	396

As the experiment results showed, the proposed filter architecture can be applied in digital signal processing systems where high performance is required. In systems with limited hardware resources, it is better to use the filter architecture proposed in [28], although this leads to performance decrease.

The proposed filter architectures can be applied to digital filters for edge detection [32], [33] and smoothing [34], discrete wavelet transform [35], and to implement the convolution operation in the convolutional layer of the convolutional neural network [36].

VII. CONCLUSION

The paper proposes a digital filter architecture with 5×5 mask based on the modified Winograd method using RNS with moduli of special type 2^α and $2^\alpha - 1$. A theoretical analysis and its hardware implementation on FPGA were performed.

Comparison with known digital filter architectures shows that the proposed method allows to:

- increase the clock frequency of 16-bit and 32-bit devices by
  - 29.63%-38.24%, compared to the filter based on the Winograd method without RNS [29],
  - 7.14% - 105.88% compared to methods based on FIR filters with MAC and TMAC blocks [28], [30], [31].
- reduce the number of occupied LUTs by 9.50% - 28.17%, and power consumption by 0.49% - 4.14%, compared with the Winograd method without RNS.
- increase filter fragment throughput (fr/s) by 29.6% - 724.7% compared to filters based on the known methods.

The research results can be efficiently used in the design of digital signal processing systems, for example, neural networks, machine vision, and many others.

ACKNOWLEDGMENT

This work was supported by the Russian Science Foundation 19-71-10033, https://rscf.ru/project/19-71-10033.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

[1] N. T. Bui, D. T. Phan, T. P. Nguyen, G. Hoang, J. Choi, Q. C. Bui, and J. Oh, "Real-time filtering and ECG signal processing based on dual-core digital signal controller system," *IEEE Sensors J.*, vol. 20, no. 12, pp. 6492–6503, Jun. 2020, doi: 10.1109/JSEN.2020.2975006.

[2] Z. Deng, P. Xu, L. Xie, K.-S. Choi, and S. Wang, "Transductive joint-knowledge-transfer TSK FS for recognition of epileptic EEG signals," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 8, pp. 1481–1494, Aug. 2018, doi: 10.1109/TNSRE.2018.2850308.

[3] R. Fu, Y.-F. Chen, Y. Huang, S. Chen, F. Duan, J. Li, J. Wu, D. Jiang, J. Gao, J. Gu, M. Zhang, and C. Chang, "Symmetric convolutional and adversarial neural network enables improved mental stress classification from EEG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 1384–1400, 2022, doi: 10.1109/TNSRE.2022.3174821.

[4] T. Neycheva, D. Dobrev, and V. Krasteva, "Common-mode driven synchronous filtering of the powerline interference in ECG," *Appl. Sci.*, vol. 12, no. 22, p. 11328, Nov. 2022, doi: 10.3390/app122211328.

[5] K. Kamirul, W. Hasbi, P. R. Hakim, and A. H. Syafrudin, "Automatic ship recognition chain on satellite multispectral imagery," *IEEE Access*, vol. 8, pp. 221918–221931, 2020, doi: 10.1109/ACCESS.2020.3042702.

[6] P. Singh, H. Jeon, S. Yun, B. Wook Kim, and S.-Y. Jung, "Vehicle positioning based on optical camera communication in V2I environments," *Comput., Mater. Continua*, vol. 72, no. 2, pp. 2927–2945, 2022, doi: 10.32604/cmc.2022.024180.

[7] G. Wang, B. Li, Y. Zhang, and J. Yang, "Background modeling and referencing for moving cameras-captured surveillance video coding in HEVC," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 2921–2934, Nov. 2018, doi: 10.1109/TMM.2018.2829163.

[8] L. M. B. Alonzo, F. B. Chioson, H. S. Co, N. T. Bugtai, and R. G. Baldovino, "A machine learning approach for coconut sugar quality assessment and prediction," in *Proc. IEEE 10th Int. Conf. Hum., Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Nov. 2018, pp. 1–4, doi: 10.1109/HNICEM.2018.8666315.

[9] J. Lu, L. Zhao, K. Chen, P. Deng, B. Li, S. Liu, and Q. An, "Real-time FPGA-based digital signal processing and correction for a small animal PET," *IEEE Trans. Nucl. Sci.*, vol. 66, no. 7, pp. 1287–1295, Jul. 2019, doi: 10.1109/TNS.2019.2908220.

[10] Q. Song, J. Zhang, L. Sun, and G. Jin, "Design and implementation of convolutional neural networks accelerator based on multidie," *IEEE Access*, vol. 10, pp. 91497–91508, 2022, doi: 10.1109/ACCESS.2022.3199441.

[11] S. M. Cho, P. K. Meher, L. T. Nhat Trung, H. J. Cho, and S. Y. Park, "Design of very high-speed pipeline FIR filter through precise critical path analysis," *IEEE Access*, vol. 9, pp. 34722–34735, 2021, doi: 10.1109/ACCESS.2021.3061759.

[12] S. Bose, A. De, and I. Chakrabarti, "Area-delay-power efficient VLSI architecture of FIR filter for processing seismic signal," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 11, pp. 3451–3455, Nov. 2021, doi: 10.1109/TCSII.2021.3081257.

[13] J. Chen, C.-H. Chang, J. Ding, R. Qiao, and M. Faust, "Tap delay-and-accumulate cost aware coefficient synthesis algorithm for the design of area-power efficient FIR filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 2, pp. 712–722, Feb. 2018, doi: 10.1109/TCSI.2017.2725916.

[14] J. Goldsmith, L. H. Crockett, and R. W. Stewart, "A natively fixed-point run-time reconfigurable FIR filter design method for FPGA hardware," *IEEE Open J. Circuits Syst.*, vol. 3, pp. 25–37, 2022, doi: 10.1109/OJCAS.2022.3152399.

[15] D. Datta, S. Akhtar, and H. S. Dutta, "FPGA implementation of symmetric systolic FIR filter using multi-channel technique," in *Proc. IEEE VLSI Device Circuit Syst. (VLSI DCS)*, Kolkata, India, Jul. 2020, pp. 225–228, doi: 10.1109/VLSIDCS47293.2020.9179926.

[16] S. Winograd, *Arithmetic Complexity of Computations*. Philadelphia, PA, USA: SIAM, 1980, p. 33, doi: 10.1137/1.9781611970364.

[17] A. Omondi and B. Premkumar, *Residue Number Systems: Theory and Implementation*. London, U.K.: Imperial College Press, 2007, p. 296.

[18] G. C. Cardarilli, L. D. Nunzio, R. Fazzolari, A. Nannarelli, M. Petricca, and M. Re, "Design space exploration based methodology for residue number system digital filters implementation," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 1, pp. 186–198, Jan. 2022, doi: 10.1109/TETC.2020.2997067.

[19] M. Valueva, P. Lyakhov, G. Valuev, and N. Nagornov, "Digital filter architecture with calculations in the residue number system by Winograd method F (2 × 2, 2 × 2)," *IEEE Access*, vol. 9, pp. 143331–143340, 2021, doi: 10.1109/ACCESS.2021.3121520.

[20] M. V. Valueva, N. N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Math. Comput. Simul.*, vol. 177, pp. 232–243, Nov. 2020.

[21] R. Zimmermann, *Binary Adder Architectures for Cell-based VLSI and Their Synthesis*. Konstanz, Germany: Hartung-Gorre, 1998.

[22] D. Zhang, J. Lv, and Z. Cheng, "An approach focusing on the convolutional layer characteristics of the VGG network for vehicle tracking," *IEEE Access*, vol. 8, pp. 112827–112839, 2020, doi: 10.1109/ACCESS.2020.3003229.

[23] X. Fu, N. Cai, K. Huang, H. Wang, P. Wang, C. Liu, and H. Wang, "M-Net: A novel U-Net with multi-stream feature fusion and multi-scale dilated convolutions for bile ducts and hepatolith segmentation," *IEEE Access*, vol. 7, pp. 148645–148657, 2019, doi: 10.1109/ACCESS.2019.2946582.

[24] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*. London, U.K.: Oxford Univ. Press, 2010, p. 492.

[25] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Trans. Comput.*, vol. C-22, no. 8, pp. 786–793, Aug. 1973, doi: 10.1109/TC.1973.5009159.

[26] A. Lavin and S. Gray, "Fast algorithms for convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4013–4021, doi: 10.1109/CVPR.2016.435.

[27] H. T. Vergos and G. Dimitrakopoulos, "On modulo $2^n + 1$ adder design," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 173–186, Feb. 2012, doi: 10.1109/TC.2010.261.

[28] P. Lyakhov, M. Valueva, G. Valuev, and N. Nagornov, "High-performance digital filtering on truncated multiply-accumulate units in the residue number system," *IEEE Access*, vol. 8, pp. 209181–209190, 2020, doi: 10.1109/ACCESS.2020.3038496.

[29] J. Yepez and S.-B. Ko, "Stride 2 1-D, 2-D, and 3-D Winograd for convolutional neural networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 4, pp. 853–863, Apr. 2020, doi: 10.1109/TVLSI.2019.2961602.

[30] C.-W. Tung and S.-H. Huang, "A high-performance multiply-accumulate unit by integrating additions and accumulations into partial product reduction process," *IEEE Access*, vol. 8, pp. 87367–87377, 2020, doi: 10.1109/ACCESS.2020.2992286.

[31] P. Lyakhov, M. Valueva, G. Valuev, and N. Nagornov, "A method of increasing digital filter performance based on truncated multiply-accumulate units," *Appl. Sci.*, vol. 10, no. 24, p. 9052, Dec. 2020, doi: 10.3390/app10249052.

[32] Y. Sumiya, T. Otsuka, Y. Maeda, and N. Fukushima, "Gaussian Fourier pyramid for local Laplacian filter," *IEEE Signal Process. Lett.*, vol. 29, pp. 11–15, 2022, doi: 10.1109/LSP.2021.3121198.

[33] R. Zhang, Q. Xiao, Y. Du, and X. Zuo, "DSPI filtering evaluation method based on Sobel operator and image entropy," *IEEE Photon. J.*, vol. 13, no. 6, pp. 1–10, Dec. 2021, doi: 10.1109/JPHOT.2021.3118924.

[34] Y. Yamak, M. Kilicaslan, and R. Demirci, "Image filter with logistic functions," in *Proc. 4th Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, İstanbul, Turkey, Oct. 2020, pp. 1–5, doi: 10.1109/ISMSIT50672.2020.9255100.

[35] L. Wang and H. Ji, "A watermarking optimization method based on matrix decomposition and DWT for multi-size images," *Electronics*, vol. 11, no. 13, p. 2027, Jun. 2022, doi: 10.3390/electronics11132027.

[36] X. Sui, Q. Lv, Y. Bai, B. Zhu, L. Zhi, Y. Yang, and Z. Tan, "A hardware-friendly low-bit power-of-two quantization method for CNNs and its FPGA implementation," *Sensors*, vol. 22, no. 17, p. 6618, Sep. 2022, doi: 10.3390/s22176618.

GEORGII VALUEV received the master's degree in applied computer science from North-Caucasus Federal University (NCFU), in 2020, where he is currently pursuing the Ph.D. degree. He works with the North-Caucasus Center for Mathematical Research as an Intern Researcher. His research interests include high-performance computing and digital image processing.

MARIA VALUEVA (Graduate Student Member, IEEE) received the bachelor's and master's degrees in applied mathematics and computer science from North-Caucasus Federal University (NCFU), in 2016 and 2018, respectively, where she is currently pursuing the Ph.D. degree. She works with the North-Caucasus Center for Mathematical Research as a Junior Researcher. Her research interests include high-performance computing, modular arithmetic, artificial neural networks, and digital image processing.

MIKHAIL BABENKO graduated in mathematics from Stavropol State University (SSU), in 2007, and the Ph.D. degree in mathematics from SSU, in 2011. He is currently the Head of the Department of Computational Mathematics and Cybernetics with North-Caucasus Federal University. He is the author of over 77 publications and five patents. His research interests include cloud computing, high-performance computing, residue number systems, neural networks, and cryptography.

ANDREI TCHERNYKH (Member, IEEE) received the Ph.D. degree from the Institute of Precise Mechanics and Computer Technology, Russian Academy of Sciences (RAS), Russia, in 1986. He is currently a Full Professor with the CICESE Research Center, Computer Science Department, Ensenada, Baja California, Mexico, and an Adjunct Professor with the Institute for System Programming, RAS, Russia. He is also the Head of the Parallel Computing Laboratory, CICESE, and the Laboratory of Problem-Oriented Cloud Computing, South Ural State University, Russia. His main research interests include resource optimization techniques, adaptive resource provisioning, multiobjective optimization, computational intelligence, incomplete information processing, cloud computing, and security. He is a member of the National System of Researchers of Mexico (SNI), Level II, and leads several national and international research projects.