UDC 004.31:681.5
DOI: 10.15587/1729-4061.2020.194342

SYNTHESIS OF FAST-OPERATING DEVICES FOR DIGITAL SIGNAL PROCESSING BASED ON THE NUMBER-THEORETIC TRANSFORMS

A. Ivashko
PhD, Professor*
E-mail: ivashkoauts@gmail.com
I. Liberg
PhD, Professor*
E-mail: i_liberg@ukr.net
D. Lunin
Senior Lecturer*
E-mail: lunindenis77@gmail.com
*Department of Automation and
Control in Technical Systems
National Technical University
«Kharkiv Polytechnic Institute»
Kyrpychova str., 2, Kharkiv, Ukraine, 61002

Запропоновано вибір модулів спеціального виду і відповідних їм первісних коренів, які допускають спрощену структуру арифметичних пристроїв, із застосуванням теоретико-числових перетворень. Розроблено метод визначення модулів, що забезпечує мінімальне число арифметичних операцій при виконанні операцій додавання і множення за модулем. Розроблено і промодельовано структури суматорів за модулями спеціального виду, що дозволяють максимально швидко виконувати операцію складання. Синтезовані і протестовані суматори за модулями чисел Ферма, Мерсенна і Голомба, які можна застосувати в арифметичних блоках швидкодіючих кореляторів і фільтрів.

Обчислення кореляцій і згорток в реальному часі стає досить трудомістким завданням у разі довгих вхідних послідовностей. Для вирішення цього завдання доцільно застосувати так звані швидкі алгоритми. Однак це вимагає високої продуктивності обчислювача згорток і кореляцій, які часто перевищують можливості сучасної обчислювальної техніки. Тому запропонована методика визначення модуля і розроблені структурні схеми суматорів за модулями спеціального виду дозволяють прискорити обчислення кореляцій і згорток з використанням теоретико-числових перетворень.

Так як операція множення за модулем виконується за допомогою операцій додавання і зсуву, то трудомісткість розрахунку теоретико-числових перетворень в значній мірі залежить від кількості одиниць в двійковому поданні ступенів первісного кореня. Операція множення, як правило, зводиться до багаторазового складання чисел, то складність і швидкодія арифметичних пристроїв для теоретико-числових перетворень визначається характеристиками суматорів за модулем.

Запропонований метод проектування обчислювальних модулів для цифрових пристроїв обчислення кореляції і згортки на основі швидких теоретико-числових перетворень забезпечує спрощену апаратну і програмну реалізацію цих структур, що призводить до високошвидкісної обробки сигналів і зображень

Ключові слова: автокореляційна функція, кореляція, згортка, програмовані логічні інтегральні схеми, дискретне перетворення Фур'є

Received date 23.12.2019
Accepted date 31.01.2020
Published date 29.02.2020

1. Introduction

The tasks of estimating correlation functions and convolution computing c arise in various fields of digital signal processing. The calculation of autocorrelation functions (ACF) and cross-correlation functions (CCF) may be necessary when processing images, in radar or sonar systems for ranging and direction finding, when calculating the spectrum of signals and in many other areas.

Real-time correlation and convolution calculation become a rather time-consuming task in the case of long input sequences. To solve this task, it is advisable to use the so-called fast algorithms. However, this requires the high-performance calculator of convolutions and correlations, which often exceed the capabilities of current computer equipment.

This task can be solved by the hardware implementation based on programmable logic integrated circuits (PLIC). There are known examples of the hardware implementation of correlation and convolution processors on PLICs [1]. However, they do not fully use the capabilities of mathematical methods for accelerating computations.

To ensure the maximum performance speed of digital signal processing processors, it is necessary, on the one hand, to optimize the convolution and correlation computing algorithms, and on the other hand, to design the structures of high-speed arithmetic units for such processors.

2. Literature review and problem statement

The calculation of convolution and correlation functions requires the number of additions and multiplications proportional to N^2, where N is the length of the processed sequence. Such a volume can be unacceptably large when processing signals in real time, therefore, faster algorithms for calculating convolution and correlation were proposed. One of these methods [2] implies calculating the calculation of a discrete Fourier transform (DFT) of the sequences of multiplication of the coefficients of the obtained DFT and the calculation of the inverse DFT of the obtained sequence. However, when calculating the DFT of sequences, the emergence of complex irrational values is inevitable, which complicates the computations, since it becomes necessary to perform calculations involving both the real and the imaginary part of the number.

Another approach is based on number-theoretic transforms (NTT) [3], which are similar to DFT and have the property of convolution, but whose coefficients accept only integer values that do not exceed a certain maximum value. The NTT of sequence x_i, i=0...N–1 is determined in the following way:

$$X_k = \sum_{i=0}^{N-1} x_i \cdot g^{ik} \left(\operatorname{mod} p\right), \tag{1}$$

where the p module and the length of the sequence N do not share the multipliers, while g is chosen to meet the condition:

$$g^N = 1\left(\operatorname{mod}\ p\right). \tag{2}$$

Specifically, for any simple p, there is a variant when N=p–1 and g is the so-called primitive root.

An analysis of the methods for finding convolutions and ACF with the help of NTT shows that the main arithmetic operations that need to be performed are addition and multiplication based on the selected simple module, that is, calculating the remainder of the result of arithmetic operations. For an arbitrary module, these operations are rather laborious; therefore, some moduli of a special form are of interest for which the modulo addition operation is much simpler [4].

As simple moduli p, in particular, the so-called Mersenne numbers in the form p=2^n–1, where n is a prime, are used.

The arithmetic modulo of Mersenne numbers is described in [5, 6]. However, these works give only the theoretical foundations of modulo arithmetic and there are no adder and multiplier schemes on the basis of which specialized PLICs and large scale integrated circuits (LSI) can be built.

Also convenient in terms of reducing hardware costs are moduli in the form $p = 2^{2^t} + 1,$ known as the Fermat numbers [7]. Similar to the procedure for Mersenne numbers, a bit with a weight of 2^n represents a value comparable to –1 modulo p. Therefore, when adding, the bit of the carry with weight 2^n is subtracted from the low order bits [8]. Thus, it is interesting to find simple p moduli for NTT that are not the Mersenne and Fermat numbers and which simplify the computation of these transforms.

Note that little attention is paid to the hardware implementation of such moduli. Examples of the implementation of adders and multipliers for some moduli are given in [9, 10]; however, the issue of synthesizing the structures of moduli applicable to all types of NTT that are quite easily implemented on PLIC has remained unresolved.

3. The aim and objectives of the study

The aim of this work is to construct methods for selecting the NTT parameters and for synthesizing the arithmetic units that would provide accelerated hardware and software implementation.

To accomplish the aim, the following tasks have been set:

– to explore the possibilities for choosing moduli of a special form and their corresponding primitive roots in order to conduct fast NTT that permit a simplified structure of arithmetic devices;

– to develop a method for determining moduli that ensure the minimum number of arithmetic operations when performing addition and multiplication operations;

– to develop adder structures for moduli of a special form that would allow the addition operation to be carried out as quickly as possible, and to evaluate the effectiveness of their implementation by modeling.

4. Selecting special-form moduli for NTT

When selecting such NTT parameters (1) as the dimensionality of transform N, the prime modulus p and the primitive root g, there are a series of hard-to-compatible requirements. Thus, for the application of the so-called fast algorithms for calculating NTT, it is advisable that the dimensionality of the transformation should be a power of two N=2^n. To simplify addition and multiplication, the module must have a special form, for example, 2^n±1.

In addition to the above-considered Fermat and Mersenne numbers, whose quantity is small, the following moduli can be considered

$$p=p_1 \cdot p_2+1=(2^a-1)\cdot 2^b+1, \tag{3}$$

which also allow simple hardware and software implementation of NTT whose dimensionality is 2^n.

A special case of the moduli $p=p_1 \cdot p_2+1=(2^a-1)\cdot 2^b+1$ are the $p=3\cdot 2^n+1$ moduli, known as the Golomb numbers [11, 12].

Using the developed software, we analyzed in the MATLAB environment all moduli in the form (3) for a=2..15, b=8..18. Composite moduli were excluded from consideration, and for each of the simple ones, a primitive root was searched for and checked whether it satisfies condition (2).

Simple moduli in the form $p=p_1 \cdot p_2+1=(2^a-1)\cdot 2^b+1$, identified in this way, as well as the corresponding sequence lengths N and primitive roots g, are summarized in Table 1.

Table 1

Simple moduli in the form $p=(2^a-1)\cdot 2^b+1$

N	a	b	p_1	p_2	p	g
262,144	2	18	3	262,144	786,433	5
65,536	1	16	1	65,536	65,537	3
16,384	3	14	7	16,384	114,689	15
16,384	6	14	63	16,384	1,032,193	94
4,096	2	12	3	4,096	12,289	41
4,096	4	12	15	4,096	61,441	19
4,096	7	12	127	4,096	520,193	71
1,024	4	10	15	1,024	15,361	84
1,024	6	10	63	1,024	64,513	21
512	4	9	15	512	7,681	62
256	1	8	1	256	257	3
256	2	8	3	256	769	7
256	5	8	31	256	7,937	71

It should be noted that the derived moduli can be used to calculate the NTT of not only the dimensionality N, specified in Table 1, but also of lower powers of twos. For example, based on modulo 61,441, one can calculate the NTT not only of dimensionality 4,096 but also 2,048, 1,024, 512, etc.

In addition to the known numbers by Fermat 65,537, by Mersenne 8,191, 131,071, 524,287, and by Golomb 12,289, 786,433, we additionally propose to use the moduli 114,689; 1,032,193; 61,441; 52,019; 15,361; 64,513.

5. Method for determining moduli that ensure a minimum number of arithmetic operations

Since the operation of modulo multiplication is performed using the operations of addition and shift, the complexity of calculating the NTT largely depends on the number of unities in the binary representation of the degrees of the primitive root *g*. For a series of values for the dimensionality of transformations *N*, various simple moduli *p* and the corresponding primitive roots *g* were sorted out from Table 1. For each of the possible moduli, the complexity of calculating the NTT from formula (1) was estimated by counting the unities in the binary representation of the degrees of the primitive root *g*; those moduli and primitive root were selected for which the total number of unities is minimal.

Results from computing the dimensionality N=1,024, frequently used in signal processing, are given in Table 2.

Table 2

Simple moduli and their corresponding number of additions in NTT

Module p	12,289	15,361	61,441	64,513	114,689
Number of additions A	29,785	28,968	32,555	33,067	33,799

An analysis of Table 2 shows that the minimum amount of computations in the calculation of fast NTT is provided by the values for module p=15,361. Having analyzed Table 1, we shall determine the primitive root as g=84. In this case, the wrong choice of a module may require 17 % more computations. Similarly, one can find a minimum number of NTT additions for any dimensionality *N* and module *p*.

6. Structure of modulo adders for NTT

For the quick calculation of correlations and convolutions using NTT, it is necessary to develop the basic "building blocks" of the NTT processors – adders and modulators. Since the operation of multiplication is typically reduced to the multiple addition of numbers, it can be argued that the complexity and performance speed of arithmetic devices for NTT is determined by the characteristics of the modulo adders.

Fig. 1 shows the proposed modulo adder circuit for numbers in the form 2^m–1, for example, the Mersenne numbers.

The scheme provides the summation of seven-digit binary numbers modulo 127.

In the scheme, FA unit is a complete adder with three inputs: a, b and p_{-1} – the input of the carry from the previous cascade, as well as with two output: s – sum and p – output carry; HA unit is a half-adder with two inputs, a and b, and two outputs: s is the sum and p is the output carry [13].

The scheme provides a summation of the carry bit with a weight of 2^n and a lower significant bit. There is also a possibility to correct the result if the sum of input values is equal to the module.

The modulo adder circuit for numbers in the form 2^m+1, for example, the Fermat numbers, is shown in Fig. 2. The adder shown ensures a summation of nine-digit binary numbers modulo 257.

Similar to the procedure described above, a bit with a weight of 2^n represents a value comparable to –1 modulo *p*. Therefore, when adding, the carry bit with a weight of 2^n is subtracted from the least significant bits.

Of greatest interest is the development of adders for moduli $p=p_1 \cdot p_2+1=(2^a–1)\cdot 2^b+1$, which provide for a wider choice of NTT dimensionalities and the ranges of operand values and results.

Consider, in particular, an adder for the Golomb numbers modulo, for which a=2, $p=3\cdot 2^n+1$.

The results of arithmetic operations for modulo $p=3\cdot 2^n+1$ can easily be reduced to residuals, given that $4\cdot 2^n$ is comparable to $(2^n–1)$ modulo $3\cdot 2^n+1$. Therefore, when added, the carry bit with a weight of $4\cdot 2^n$ forms a value consisting of *n* unit bits. Next, this formed value must be added to *n* – the least significant bits of the resulting sum. The result is the sum modulo $p=3\cdot 2^n+1$.

For the case when, at adding, one obtains a carry bit equal to zero, it is necessary to perform additional processing of the computation result. To this end, one should sum $(n–1)$ – the least significant bits of the resulting sum, and then multiply them by two bits with a weight of 2^n and 2^{n+1} and logically add it to the overflow bit, which weights 2^{n+2}. The derived bit is then summed with each of the *n*-least significant bits of the received sum.

The block diagram of the modulo adder for the Golomb numbers, based on the above reasoning, is shown in Fig. 3.

All the above structures were modeled and tested in the Active-HDL ver.9.1 environment, which proved their operability and the possibility of constructing NTT processors based on PLIC or specialized LSI.

Fig. 4 shows an adder of the VHDL model for the Mersenne number $8{,}191=2^{13}–1$ modulo; Fig. 5 – time diagrams of the adder operation.

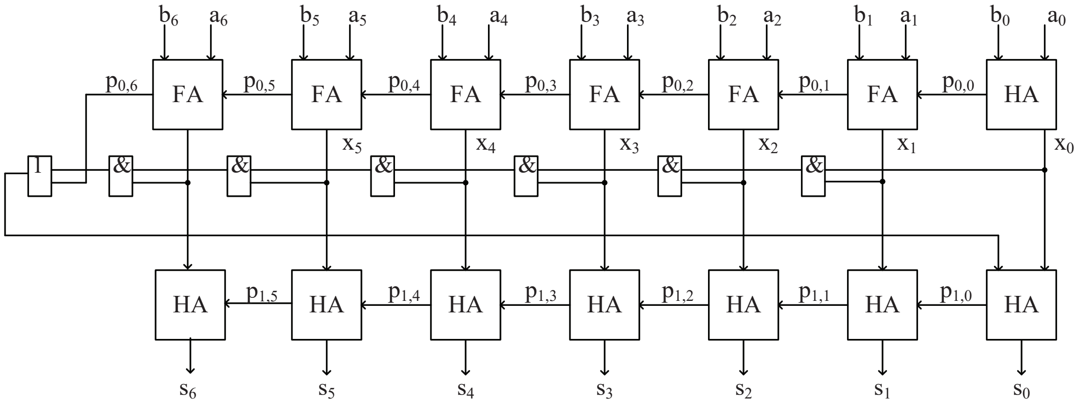


Fig. 1. Block diagram of the modulo 127 adder

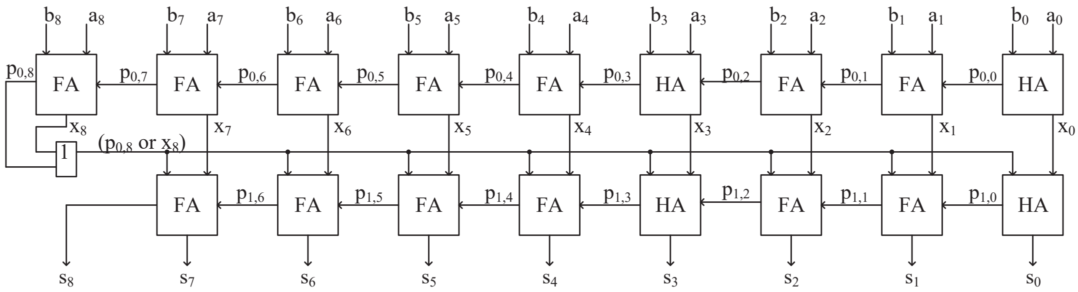

b8 a8 b7 a7 b6 a6 b5 a5 b4 a4 b3 a3 b2 a2 b1 a1 b0 a0
FA FA FA FA FA HA FA FA HA
$p_{0,8}$ $p_{0,7}$ $p_{0,6}$ $p_{0,5}$ $p_{0,4}$ $p_{0,3}$ $p_{0,2}$ $p_{0,1}$ $p_{0,0}$
x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0
1
($p_{0,8}$ or x_8)
FA FA FA FA HA FA FA HA
$p_{1,6}$ $p_{1,5}$ $p_{1,4}$ $p_{1,3}$ $p_{1,2}$ $p_{1,1}$ $p_{1,0}$
s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 s_0

Fig. 2. Block diagram of the modulo 257 adder

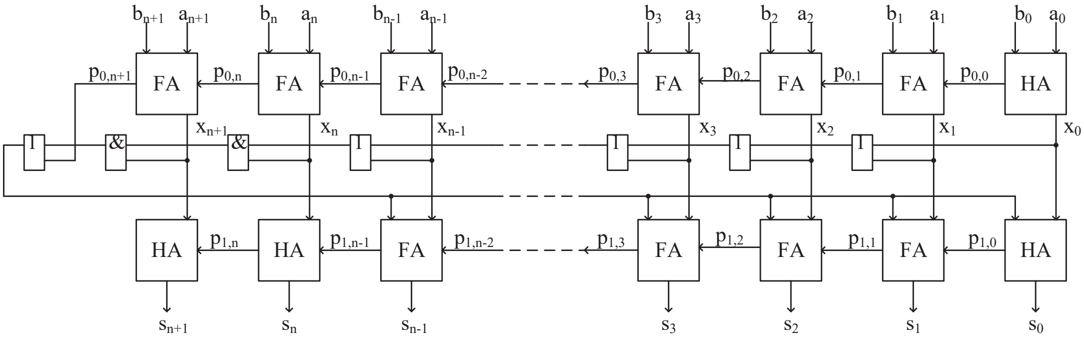

b_{n+1} a_{n+1} b_n a_n b_{n-1} a_{n-1} b_3 a_3 b_2 a_2 b_1 a_1 b_0 a_0
FA FA FA FA FA FA HA
$p_{0,n+1}$ $p_{0,n}$ $p_{0,n-1}$ $p_{0,n-2}$ $p_{0,3}$ $p_{0,2}$ $p_{0,1}$ $p_{0,0}$
x_{n+1} x_n x_{n-1} x_3 x_2 x_1 x_0
1 & & 1 1 1 1
HA HA FA FA FA FA HA
$p_{1,n}$ $p_{1,n-1}$ $p_{1,n-2}$ $p_{1,3}$ $p_{1,2}$ $p_{1,1}$ $p_{1,0}$
s_{n+1} s_n s_{n-1} s_3 s_2 s_1 s_0

Fig. 3. Block diagram of the modulo p=3·2^n+1 adder

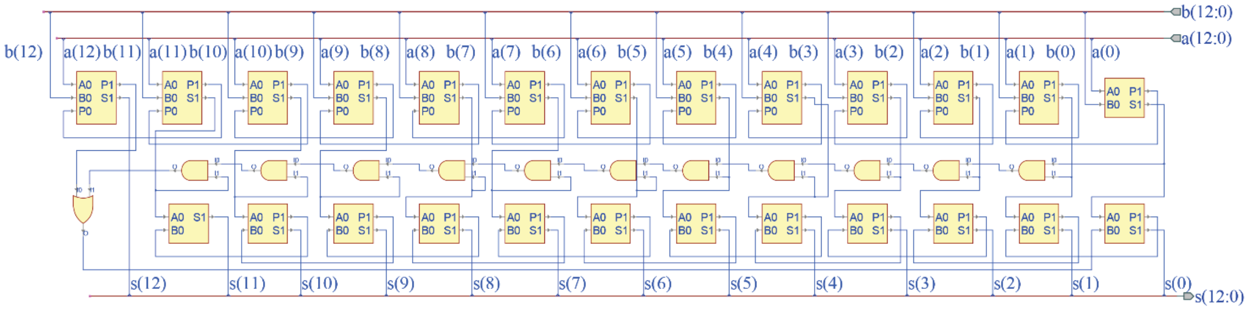
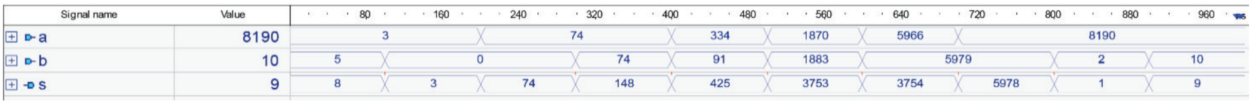

b(12:0)
a(12:0)
b(12) a(12) b(11) a(11) b(10) a(10) b(9) a(9) b(8) a(8) b(7) a(7) b(6) a(6) b(5) a(5) b(4) a(4) b(3) a(3) b(2) a(2) b(1) a(1) b(0) a(0)
A0 P1 B0 S1 P0
A0 S1 B0
A0 P1 B0 S1
s(12) s(11) s(10) s(9) s(8) s(7) s(6) s(5) s(4) s(3) s(2) s(1) s(0)
s(12:0)

Fig. 4. Model of the modulo 8,191 adder



Signal name	Value
a	8190
b	10
s	9

Fig. 5. Results from modeling the modulo 8,191 adder

7. Discussion of results of studying the methods for NTT acceleration

Known works consider only the moduli in the form of the Fermat, Mersenne, and Golomb numbers. By applying methods from the theory of numbers, this paper has demonstrated a two-fold and larger increase in the number of moduli for theoretical-numerical transformations (Table 1). Such an expansion of the range of moduli makes it possible, on the one hand, to better adjust the parameters of a NTT processor to specific requirements for accuracy and speed, on the other hand, to accelerate the computation of additions for the module.

In addition, known papers paid little attention to the form of the primitive root. Our analysis of possible moduli and primitive roots has made it possible to choose those that ensure the simplification of exponentiation and multiplication operations (Table 2).

It has also become possible to design circuits for the fast-acting modulo adders, easily implemented on PLIC and specialized LSI, for a wide range of moduli not reported in available studies.

Thus, this research has allowed us to increase the speed performance of devices for computing NTT due to the use, on the one hand, the number theory methods, and, on the other hand, modern methods of synthesis and modeling of digital devices.

At the same time, the selection of modules proceeded to some extent empirically. Better results could be probably achieved by applying such an algebraic apparatus as the Galois fields and group characters when choosing moduli.

In addition, this work does not include the adder schemes for p=(2^a–1)·2^b+1 modulo for the case a>2, that is, modulo generalized Golomb numbers.

In the future, it is advisable to synthesize such schemes and, if possible, develop software for the synthesis of adders and multipliers for all possible moduli in a special form. This task, however, may require a volume of computations beyond the capabilities of modern computer equipment.

Theoretical and experimental estimates should also be made of the performance speed of the proposed structures depending on the element base used.

8. Conclusions

1. A procedure has been proposed for choosing the optimal moduli for calculating NTT, which could make it possible to increase by two times, and larger, the number of moduli for NTT, as well as to better adjust the NTT processor parameters for specific requirements for accuracy and performance speed.

2. A procedure has been proposed for analyzing the NTT parameters and choosing the primitive roots, which allows 15–20 % faster computation of correlations and convolutions when using NTT. A procedure has been proposed and the algorithms and programs have been developed for analyzing the NTT primitive roots, which make it possible to accelerate the computation of correlations and convolutions when using NTT.

3. We have synthesized and tested high-speed adders for the proposed moduli, which could serve as the basis for arithmetic units in high-speed correlators and filters.

References

1. Ortega Cisneros, S., Rivera D., J., Moreno Villalobos, P., Torres, C., C. A., Hernandez-Hector, H., Raygoza, P., J. J. (2015). An image processor for convolution and correlation of binary images implemented in FPGA. 2015 12th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). doi: https://doi.org/10.1109/iceee.2015.7357987
2. Ito, I. (2013). A Computing Method for Linear Convolution and Linear Correlation in the DCT Domain. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E96.A (7), 1518–1525. doi: https://doi.org/10.1587/transfun.e96.a.1518
3. Valencia, F., Khalid, A., O'Sullivan, E., Regazzoni, F. (2017). The design space of the number theoretic transform: A survey. 2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS). doi: https://doi.org/10.1109/samos.2017.8344640
4. Zhang, J., Li, S. (2009). Data-recovery algorithm and circuit for cyclic convolution based on FNT. IEICE Electronics Express, 6 (14), 1019–1024. doi: https://doi.org/10.1587/elex.6.1019
5. Boussakta, S., Hamood, M. T., Rutter, N. (2012). Generalized New Mersenne Number Transforms. IEEE Transactions on Signal Processing, 60 (5), 2640–2647. doi: https://doi.org/10.1109/tsp.2012.2186131
6. Campbell, K., Lin, C.-H., Chen, D. (2018). Low-cost hardware architectures for mersenne modulo functional units. 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC). doi: https://doi.org/10.1109/aspdac.2018.8297388
7. Toivonen, T., Heikkila, J. (2006). Video filtering with Fermat number theoretic transforms using residue number system. IEEE Transactions on Circuits and Systems for Video Technology, 16 (1), 92–101. doi: https://doi.org/10.1109/tcsvt.2005.858612
8. Zhang, J., Li, S. (2009). High Speed Parallel Architecture for Cyclic Convolution Based on FNT. 2009 IEEE Computer Society Annual Symposium on VLSI. doi: https://doi.org/10.1109/isvlsi.2009.10
9. Kumar, S., Chang, C.-H. (2017). A Scaling-Assisted Signed Integer Comparator for the Balanced Five-Moduli Set RNS {2^n–1, 2^n, 2^n+1, 2^{n+1}–1, 2^{n-1}–1}. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25 (12), 3521–3533. doi: https://doi.org/10.1109/tvlsi.2017.2748984
10. Efstathiou, C., Vergos, H. T., Nikolos, D. (2003). Modulo 2n±1 adder design using select-prefix blocks. IEEE Transactions on Computers, 52 (11), 1399–1406. doi: https://doi.org/10.1109/tc.2003.1244938
11. Golomb, S. W., Reed, I. S., Truong, T. K. (1977). Integer Convolution over Finite Field GF($3 \cdot 2^{n+1}$). SIAM Journal on Applied Mathematics, 32 (2), 356–365. doi: https://doi.org/10.1137/0132029
12. Chernov, V. M., Korepanov, A. O. (2006). Teoretiko-chislovye preobrazovaniya v zadachah tsifrovoy obrabotki signalov. Samara, 112.
13. Baozhou, Z., Ahmed, N., Peltenburg, J., Bertels, K., Al-Ars, Z. (2019). Diminished-1 Fermat Number Transform for Integer Convolutional Neural Networks. 2019 IEEE 4th International Conference on Big Data Analytics (ICBDA). doi: https://doi.org/10.1109/icbda.2019.8713250