The Procedure for Implementing the Operation of Multiplying Two Matrices Using the Residual Number System

Victor Krasnobayev
Electronics and Control Systems Department
V. N. Karazin Kharkiv National University
Kharkiv, Ukraine
v.a.krasnobaev@gmail.com

Alexandr Kuznetsov
Department of Information Systems and Technologies Security V. N. Karazin Kharkiv National University,
Kharkiv, Ukraine
kuznetsov@karazin.ua

Alina Yanko
Department of Computer and Information Technologies and Systems
National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine
al9_yanko@ukr.net

Tetiana Kuznetsova
Department of Information Systems and Technologies Security V. N. Karazin Kharkiv National University,
Kharkiv, Ukraine
kuznetsova.tatiana17@gmail.com

***Abstract*—The report considers solution to the problem of improving the speed implementation of the operation of multiplying two square matrices of the same dimension. To carry out calculations and comparative analysis of the speed of the multiplication operation, we consider a computer system (CS) in the positional binary number system (PNS) and in the non-positional number system in the residual classes (the residual number system - RNS). A comparative analysis of the performance of the CS was carried out with the same characteristics of the computing system : equal lengths of bit grids, the same command systems, the same methods of addressing operands and instructions, the same clock speed of the processor, the equal number of program commands, etc. When calculating the speed of the matrix multiplication operation, the fastest data processing method in RNS was used, based on the tabular principle.**

Keywords—algorithm for multiplying matrices, performance evaluation, positional number system, residual number system.

I. Introduction

One of the most important characteristics of a computing system is its speed and performance [1]–[4]. Computer systems (CS) performance is determined by the amount of computational work performed per unit of time. The most important factors affecting the performance of the CS are primarily factors such as: the clock speed of the CS processor, the number of program commands (tasks, algorithms) and the number of clock cycles for executing one command (average execution time of one command). In turn, the command consists of a sequence of arithmetic and other operations. In the general case, it can be said that quantitatively the performance of the CS depends on the clock frequency of the processor and on the time of implementation of the arithmetic and other operations that make up the command [1], [2], [5].

It is known that the use of residual number system (RNS) in the CS can significantly increase the speed of implementation of integer arithmetic operations [6]–[9]. At the same time, practical confirmation of the effectiveness of the use of RNS is required to increase the performance of the CS in the processing of integer data. Therefore, the task of calculating and comparative analysis of the performance of the CS functioning in the RNS and in the usual binary positional number system for specific computational algorithms is relevant and practically important.

As noted above, under the performance of the CS refers to the performance of the processor of the CS [1], [2]. This is due to the fact that the number system adopted in this CS primarily affects the structure and principles of operation of the processor of a computing system (CoS). A comparative analysis of the performance of the CS was carried out with the same characteristics of the CoS: equal lengths of bit grids, the same command systems, the same methods of addressing operands and instructions, the same clock speed of the processor, the equal number of program commands, etc. To conduct a comparative analysis (performance evaluation) of the performance of the CS, the analytical relationship is used, where $T_M^{(RNS)}$ – the time it takes the CS in the PNS to solve one specific problem; $T_M^{(SRC)}$ – the time of solving the same task by the CS in the RNS. In this case, the quantitative value of the efficiency coefficient $K_{eff.}^{(l)}$ depends only on the time of implementation of arithmetic and other operations that are part of the commands of the problem being solved in the PNS and in the RNS [10]–[12].

II. An Algorithm Basic Operation Matrix Multiplication Operation

Consider the problem of multiplying two square matrices A and B of equal size $N \times N$. The nodes of the matrix contain integers. Matrices A and B are represented respectively in the form [13]–[16]:

This work was supported in part by the National Research Foundation of Ukraine under Grant 2020.01/0351.

978-x-xxxx-xxxx-x/20/$31.00 ©2020 IEEE

$$A=\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{vmatrix}$$

and

$$B=\begin{vmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ b_{21} & b_{22} & \cdots & b_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NN} \end{vmatrix}$$

The result C of the multiplication of two matrices A and B will be represented as [17]–[20]:

$$C=\begin{vmatrix} c_{11} & c_{12} & \cdots & c_{1N} \\ c_{21} & c_{22} & \cdots & c_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ c_{N1} & c_{N2} & \cdots & c_{NN} \end{vmatrix}.$$

It is known that the basic operation (BO) in calculating the product of two matrices is an operation of the form:

$$c_{ij}=\sum\nolimits_{i,j}^{N}\left(a_{ik}\cdot b_{kj}\right)=a_{i1}\cdot b_{1j}+a_{i2}\cdot b_{2j}+a_{i3}\cdot b_{3j}+$$

$$+\ldots+a_{iN}\cdot b_{Nj}\ \left(i,j=\overline{1,N}\right),$$

where c_{ij} – corresponding element lying at the intersection of the i – row and j – column of the matrix C. As can be seen from the BO algorithm, the number of multiplications and additions is equal to N and $N-1$ respectively. The total number of BOs in the algorithm for multiplying two matrices, i.e. when determining the value $C=A\times B$, it is equal to the number N^2 of matrix C with size $(N\times N)$ elements $c_{ij}\left(i,j=\overline{1,N}\right)$.

In this case, the time of multiplication of two matrices in the PNS is represented by the expression:

$$T_M^{(PNS)}=N^2\cdot\left[N\cdot t_{mult.}+\left(N-1\right)\cdot t_{add.}\right], \quad (1)$$

It is known that the time of addition $t_{add.}$ and the time of multiplication $t_{mult.}$ of the ρ-bit numbers in the PNS is determined respectively by the formulas:

$$t_{mult.}=2\cdot\tau\cdot\rho^2 \text{ and } t_{add.}=\tau\cdot(2\rho-1),$$

where

$$\tau_T=\tau_{AND}=\tau_{OR}=\tau/2, \quad (2)$$

The time of multiplication in the RNS of two matrices is represented by the expression:

$$T_M^{(RNS)}=N^2\cdot\left[N\cdot 2+(N-1)\cdot 2\right]\cdot\tau/2,$$

where τ_m and τ_a are the times of the operation, respectively, of the multiplication and addition of the ρ-bit numbers in the RNS.

Given the choice of the tabular method for implementing the arithmetic operation, the previous (2) expression will be presented in the form:

$$T_M^{(RNS)}=N^2\cdot\left[N\cdot\tau_T+(N-1)\cdot\tau_T\right],$$

where $\tau_m=\tau_a=\tau_T=\tau_{AND}=\tau_{OR}=\tau/2.$

Based on the notation introduced, the previous expression $T_M^{(RNS)}$ is represented as:

$$T_M^{(RNS)}=N^2\cdot\left[N\cdot 2+(N-1)\cdot 2\right]\cdot\tau/2 \quad (3)$$

Expressions (1) and (3) are working formulas for determining the time of multiplication of two matrices. Based on these formulas, we will carry out a calculation and a comparative analysis of the efficiency $K_{eff.}^{(l)}=\frac{T_M^{(PNS)}}{T_M^{(RNS)}}$ in terms of the implementation time (productivity) of the operation of multiplying two matrices for PNS and RNS for various values of the l-byte grid of the CS:

$$K_{eff.}^{(l)}=\frac{N\cdot 2\cdot\tau\cdot\rho^2+(N-1)\cdot\tau\cdot(2\rho-1)}{(4\cdot N-2)\cdot\frac{\tau}{2}}=$$

$$=\frac{2\cdot\left[N\cdot 2\cdot\rho^2+(N-1)\cdot(2\rho-1)\right]\cdot\tau}{(4\cdot N-2)\tau}=$$

$$=\frac{2\cdot\left[N\cdot 2\cdot\rho^2+(N-1)\cdot(2\rho-1)\right]}{4\cdot N-2}. \quad (4)$$

In accordance with formula (4), we carry out a calculation and comparative analysis of the efficiency $K_{eff.}^{(l)}=\frac{T_M^{(PNS)}}{T_M^{(RNS)}}$ of using RNS in terms of the performance of the task for variants of matrices of finite size $N\times N$ with N=2, 4, 8, 10 and 100.

Option 1. Let $l=1$ $(\rho=8)$. In this case, formula (4) will have the form:

$$K_{eff.}^{(1)}=\frac{2\cdot\left[N\cdot 2\cdot 8^2+(N-1)\cdot(2\cdot 8-1)\right]}{4\cdot N-2}=$$

$$= \frac{2 \cdot [128 \cdot N + 15 \cdot (N-1)]}{4 \cdot N - 2}.$$

Let, for example, $N = 2$. In this case, the efficiency coefficient is:

$$K_{eff.}^{(1)} = \frac{2 \cdot [128 \cdot 2 + 15 \cdot 1]}{4 \cdot 2 - 2} = \frac{542}{6} \approx 90$$

Similarly, we will carry out the calculations for the values $N = 4, 8, 10, 100$.

Option 2. Let $l = 2$ $(\rho = 16)$. In this case, formula (4) will have the form:

$$K_{eff.}^{(2)} = \frac{2 \cdot \left[N \cdot 2 \cdot 16^2 + (N-1) \cdot (2 \cdot 16 - 1) \right]}{4 \cdot N - 2} =$$
$$= \frac{2 \cdot [512 \cdot N + 31 \cdot (N-1)]}{4 \cdot N - 2}$$

Let $N = 2$. In this case, the efficiency coefficient is:

$$K_{eff.}^{(2)} = \frac{2 \cdot [512 \cdot 2 + 31 \cdot 1]}{4 \cdot 2 - 2} = \frac{2110}{6} \approx 352$$

Similarly, we will carry out the calculations for the values $N = 4, 8, 10, 100$.

Option 3. Let $l = 3$ $(\rho = 24)$. In this case, formula (4) will have the form:

$$K_{eff.}^{(3)} = \frac{2 \cdot \left[N \cdot 2 \cdot 24^2 + (N-1) \cdot (2 \cdot 24 - 1) \right]}{4 \cdot N - 2} =$$
$$= \frac{2 \cdot [1152 \cdot N + 47 \cdot (N-1)]}{4 \cdot N - 2}$$

Let $N = 2$. In this case, the efficiency coefficient is:

$$K_{eff.}^{(3)} = \frac{2 \cdot [1152 \cdot 2 + 47 \cdot 1]}{4 \cdot 2 - 2} = \frac{4702}{6} \approx 784$$

Similarly, we will carry out the calculations for the values $N = 4, 8, 10, 100$.

Option 4. Let $l = 4$ $(\rho = 32)$. In this case, formula (4) will have the form:

$$K_{eff.}^{(4)} = \frac{2 \cdot \left[N \cdot 2 \cdot 32^2 + (N-1) \cdot (2 \cdot 32 - 1) \right]}{4 \cdot N - 2} =$$
$$= \frac{2 \cdot [2048 \cdot N + 63 \cdot (N-1)]}{4 \cdot N - 2}.$$

Let $N = 2$. In this case, the efficiency coefficient is:

$$K_{eff.}^{(4)} = \frac{2 \cdot [2048 \cdot 2 + 63 \cdot 1]}{4 \cdot 2 - 2} = \frac{8318}{6} \approx 1386.$$

Similarly, we will carry out the calculations for the values $N = 4, 8, 10, 100$.

Option 5. Let $l = 8$ $(\rho = 64)$. In this case, formula (4) will have the form:

$$K_{eff.}^{(8)} = \frac{2 \cdot \left[N \cdot 2 \cdot 64^2 + (N-1) \cdot (2 \cdot 64 - 1) \right]}{4 \cdot N - 2} =$$
$$= \frac{2 \cdot [8192 \cdot N + 127 \cdot (N-1)]}{4 \cdot N - 2}$$

Let $N = 2$.

$$K_{eff.}^{(8)} = \frac{2 \cdot [8192 \cdot 2 + 127 \cdot 1]}{4 \cdot 2 - 2} = \frac{33022}{6} \approx 5504.$$

Similarly, we will carry out the calculations for the values $N = 4, 8, 10, 100$.

III. An Accelerated Method for Implementing the Procedure of Multiplying Two Matrices

The results of the calculation and comparative analysis of the effectiveness $K_{eff.}^{(l)}$ of the use of RNS are given in Tables I, II. Consider a faster (accelerated) method for implementing the multiplication procedure in the PNS of two matrices (Fig. 1). This method of acceleration allows you to combine in time the implementation of the multiplication of the form $\{a_{ik} \times b_{ik}\}$, for $k = const$ (see the composition of BO). Where in $i, j, k = \overline{1, N}$.

TABLE I. Estimated Data and Comparative Analysis of the Effectiveness of Using RNS (Normal Procedure)

l	N	T/τ Normal procedure		
		$T_M^{(PNS)}$	$T_M^{(RNS)}$	$K_{eff.}^{(l)}$
$l=1$	2	1084	12	90
	4	8912	112	80
	8	72256	960	75
	10	$1415 \cdot 10^2$	$19 \cdot 10^2$	74
	100	$14285 \cdot 10^4$	$199 \cdot 10^4$	72
$l=2$	2	4220	12	352
	4	34256	112	306
	8	276032	960	287
	10	$5399 \cdot 10^2$	$19 \cdot 10^2$	284
	100	$54269 \cdot 10^4$	$199 \cdot 10^4$	273
$l=3$	2	9404	12	784
	4	75984	112	678
	8	610880	960	636
	10	$11943 \cdot 10^2$	$19 \cdot 10^2$	629
	100	$119853 \cdot 10^4$	$199 \cdot 10^4$	602
$l=4$	2	16636	12	1386
	4	134096	112	1197
	8	$10768 \cdot 10^2$	960	1122
	10	$21047 \cdot 10^2$	$19 \cdot 10^2$	1108
	100	$211037 \cdot 10^4$	$199 \cdot 10^4$	1060
$l=8$	2	66044	12	5504
	4	530384	112	4736
	8	$42512 \cdot 10^2$	960	4428

	10	$83063 \cdot 10^2$	$19 \cdot 10^2$	4372
	100	$831773 \cdot 10^4$	$199 \cdot 10^4$	4180

TABLE II. ESTIMATED DATA AND COMPARATIVE ANALYSIS OF THE EFFECTIVENESS OF USING RNS (ACCELERATED PROCEDURE)

l	N	T/τ Accelerated procedure		
		$T_M^{(PNS)}$	$T_M^{(RNS)}$	$K_{eff.}^{(l)}$
$l=1$	2	147	3	47
	4	158	7	22,6
	8	173	15	11,5
	10	188	19	9,8
	100	233	199	1,2
$l=2$	2	543	3	181
	4	574	7	82
	8	605	15	40,3
	10	636	19	33,5
	100	729	199	3,6
$l=3$	2	1199	3	399,6
	4	1246	7	178
	8	1293	15	86,2
	10	1340	19	70,5
	100	1481	199	7,4
$l=4$	2	2111	3	703,6
	4	2174	7	310,6
	8	2237	15	149,1
	10	2300	19	121
	100	2489	199	12,5
$l=8$	2	8319	3	2773
	4	8446	7	1207
	8	8573	15	572
	10	8700	19	458
	100	9081	199	46

In this case, the implementation time $T_M^{(PNS)}$ of the operation of accelerated multiplication of two matrices A and B in RNS is:

$$T_M^{(PNS)} = \tau_{mult.}^{(PNS)} + N_{add.} \cdot \tau_{add.}^{(PNS)} =$$

$$= 2 \cdot \rho^2 \cdot \tau + (]\log_2 N[) \cdot (2\rho - 1) \cdot \tau =$$

$$= \tau \cdot \{2 \cdot \rho^2 + (]\log_2 N[) \cdot (2\rho - 1)\} \qquad (5)$$

For accelerated multiplication, the efficiency $K_{eff.}^{(l)}$ is determined by the expression:

$$K_{eff.}^{(l)} = \frac{\tau \cdot \{2 \cdot \rho^2 + (]\log_2 N[) \cdot (2 \cdot \rho - 1)\}}{(4 \cdot N - 2) \cdot \frac{\tau}{2}} \text{ or}$$

$$K_{eff.}^{(l)} = \frac{2 \cdot \{2 \cdot \rho^2 + (]\log_2 N[) \cdot (2 \cdot \rho - 1)\}}{4 \cdot N - 2}$$

In accordance with formula (6), we will carry out a calculation and comparative analysis of the implementation efficiency $K_{eff.}^{(l)}$ of the considered algorithm in the PNS and in the RNS for variants of matrices of finite size $N \times N$ at $N = 4, 8, 10$ and 100.

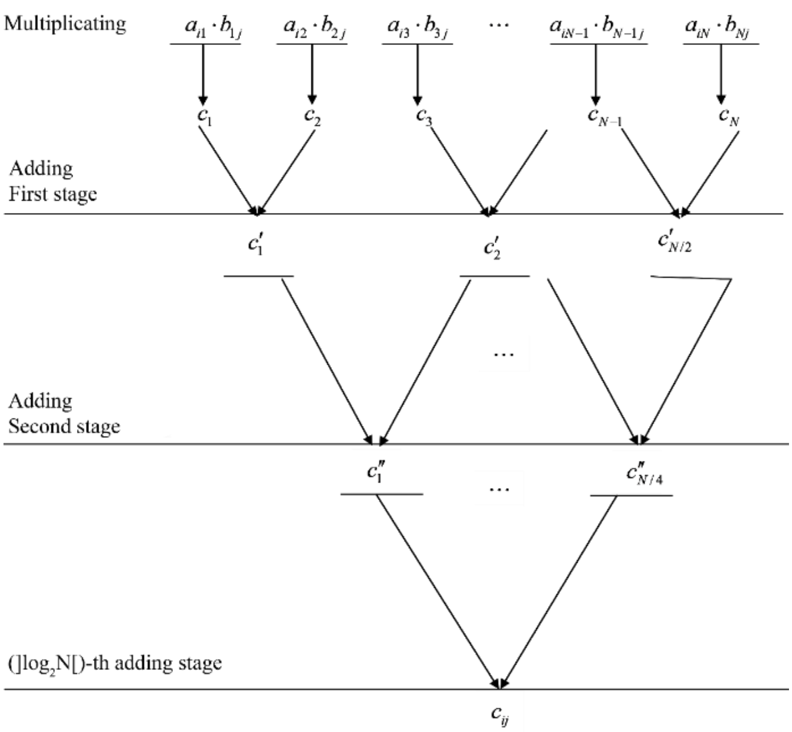

Fig. 1. Scheme of implementation of the algorithm for the accelerated multiplication of two matrices in RNS

IV. CONCLUSION

The results of the studies showed that a significant impact on the performance of the CS is affected by the methods of implementing operations that are part of the program commands. At the same time, the methods of implementation of operations to a greater extent depend on the number system (NS) which is used in the CS. So, RNS, unlike PNS, defines three principles for the implementation of arithmetic operations: the adder principle, the principle of ring shift and the tabular principle. In turn, these principles underlie many methods for implementing arithmetic operations of RNS. When calculating the performance, the fastest data processing method in the RNS was used, based on the tabular principle. An analysis of the results of calculations and a comparative analysis of the performance of the CS operating in the RNS and in the binary PNS showed that from the point of view of increasing the CoS performance it is preferable to use the CS in the RNS [21]–[24].

The results of the analysis of the calculations showed that with an increase in the length of the CS discharge grid, which is characteristic of the current trend in the development of computer systems and tools for processing large amounts of information in real time, the efficiency of using a non-positional number system in RNS to increase data processing speed increases [25]–[27].

REFERENCES

[1] A. Yadin, *Computer Systems Architecture*. Chapman and Hall/CRC, 2016.

[2] R. S. Alford, *Computer Systems Engineering Management*. CRC Press, 2018.

[3] D. A. Reed, “Performance instrumentation techniques for parallel systems,” in *Performance Evaluation of Computer and Communication Systems*, Berlin, Heidelberg, 1993, pp. 463–490, doi: 10.1007/BFb0013864.

[4] S. A. Wright, “Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems,” *Future Gener. Comput. Syst.*, vol. 92, pp. 900–902, Mar. 2019, doi: 10.1016/j.future.2018.11.020.

[5] “Proceedings of PMBS 2018: Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems,” in *2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, Nov. 2018, pp. 1–1, doi: 10.1109/PMBS.2018.8641570.

[6] J. O. Tuazon, “Residue number system in computer arithmetic,” Doctor of Philosophy, Iowa State University, Digital Repository, Ames, 1969.

[7] S. Timarchi and K. Navi, “Efficient Class of Redundant Residue Number System,” in *2007 IEEE International Symposium on Intelligent Signal Processing*, Oct. 2007, pp. 1–6, doi: 10.1109/WISP.2007.4447506.

[8] P. V. A. Mohan, *Residue Number Systems*. Cham: Springer International Publishing, 2016.

[9] M. G. Arnold, “The residue logarithmic number system: theory and implementation,” in *17th IEEE Symposium on Computer Arithmetic (ARITH’05)*, Jun. 2005, pp. 196–205, doi: 10.1109/ARITH.2005.44.

[10] V. Krasnobayev, A. Kuznetsov, A. Yanko, B. Akhmetov, and T. Kuznetsova, “Processing of the Residuals of Numbers in Real and Complex Numerical Domains,” in *Data-Centric Business and Applications*, vol. 48, T. Radivilova, D. Ageyev, and N. Kryvinska, Eds. Cham: Springer International Publishing, 2021, pp. 529–555.

[11] P. V. Ananda Mohan, “Specialized Residue Number Systems,” in *Residue Number Systems: Theory and Applications*, P. V. A. Mohan, Ed. Cham: Springer International Publishing, 2016, pp. 177–193.

[12] V. Krasnobayev, A. Kuznetsov, V. Popenko, A. Kononchenko, and T. Kuznetsova, “Determination of Positional Characteristics of Numbers in the Residual Class System,” in *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, May 2020, pp. 151–156, doi: 10.1109/DESSERT50317.2020.9125030.

[13] “Introduction to Matrix Algebra.” https://www.lulu.com/content/2143570 (accessed Aug. 16, 2020).

[14] “Matrix Reference Manual.” http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html (accessed Aug. 16, 2020).

[15] C. P. López, “Matrix Algebra,” in *MATLAB Matrix Algebra*, C. P. López, Ed. Berkeley, CA: Apress, 2014, pp. 45–82.

[16] K. B. Petersen and M. S. Pedersen, “[http://matrixcookbook.com],” p. 72.

[17] J. Gilbert and L. Gilbert, “Chapter 3 - Matrix Multiplication,” in *Linear Algebra and Matrix Theory*, J. Gilbert and L. Gilbert, Eds. San Diego: Academic Press, 1995, pp. 59–112.

[18] V. Kuchukov and M. Babenko, “The Application of Modular Arithmetic for Matrix Calculations,” in *2019 Ivannikov Ispras Open Conference (ISPRAS)*, Dec. 2019, pp. 49–54, doi: 10.1109/ISPRAS47671.2019.00013.

[19] “Wolfram Demonstrations Project brings ideas to life with over 11k interactive #WolframNotebooks for education, research, recreation & more. #WolframDemo #WolfLang.” /MatrixMultiplication/ (accessed Aug. 16, 2020).

[20] “Matrices, Matrix Arithmetic, Determinants,” in *Computational and Algorithmic Linear Algebra and n-Dimensional Geometry*, 0 vols., WORLD SCIENTIFIC, 2011, pp. 157–217.

[21] V. Krasnobayev, A. Kuznetsov, I. Lokotkova, A. Kiian, and T. Kuznetsova, “Techniques for Raising the Remainder to a Power in the System of Residual Classes,” in *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, May 2020, pp. 145–150, doi: 10.1109/DESSERT50317.2020.9125049.

[22] V. Krasnobayev, A. Kuznetsov, S. Koshman, and S. Moroz, “Improved Method of Determining the Alternative Set of Numbers in Residue Number System,” in *Recent Developments in Data Science and Intelligent Analysis of Information*, vol. 836, O. Chertov, T. Mylovanov, Y. Kondratenko, J. Kacprzyk, V. Kreinovich, and V. Stefanuk, Eds. Cham: Springer International Publishing, 2019, pp. 319–328.

[23] Y. Zhang, “An FPGA implementation of redundant residue number system for low-cost fast speed fault-tolerant computations,” Thesis, 2018.

[24] P. V. An, a Mohan, CDAC, and Bangalore, “Implementation of Residue Number System Based Digital Filters – A Quarterly Publication of ACCS.” https://journal.accsindia.org/implementation-of-residue-number-system-based-digital-filters/ (accessed Aug. 16, 2020).

[25] H. Kopetz, Ed., “Real-Time Operating Systems,” in *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Boston, MA: Springer US, 1997, pp. 211–225.

[26] R. F. Hodson, *Real-Time Expert Systems Computer Architecture*. CRC Press, 2018.

[27] *Hard real-time computing systems*, vol. 416. Boston, MA: Springer US, 1997.