Вісник Тернопільського національного технічного університету
https://doi.org/10.33108/visnyk_tntu
Scientific Journal of the Ternopil National Technical University
2023, № 2 (110) https://doi.org/10.33108/visnyk_tntu2023.02
ISSN 2522-4433. Web: visnyk.tntu.edu.ua

UDC 681.3

VERY-LARGE-SCALE INTEGRATION DEVICE FOR PARALLEL VERTICAL GROUP COMPUTING THE SUM OF SQUARED DIFFERENCES

Ivan Tsmots[1]; Ihor Ihnatiev[2]; Stepan Ivasiev[2]

[1]Lviv Polytechnic National University, Lviv, Ukraine,
[2]West Ukrainian National University, Ternopil, Ukraine

Summary. *Is a paper that proposes a new method for computing sum-of-squares differences in a parallel vertical environment. The method is based on a group approach, which allows you to divide the task into several subtasks and calculate them in parallel.*

The article considers the problem of calculating the sum of squared differences between elements of large data arrays. Applying traditional methods of calculating such sums in parallel environments can be inefficient due to the exchange of large amounts of data between nodes. The proposed method allows to reduce the amount of transmitted data and increase the efficiency of calculations. The article proposes a new method for calculating the sum of squared differences, which allows to increase the efficiency of calculations in a parallel vertical environment. Testing of the method on different data sets shows its high efficiency compared to traditional methods of calculating sums of squared differences in parallel environments. The proposed method can be applied in various areas that require the processing of large volumes of data, and allows to increase the efficiency of calculations and reduce their execution time. The methods, algorithms and structures of devices for computing the sum of squared differences have been analyzed and their defects have been defined in the article. It has been defined that the device for computing the sum of squared differences should support the next: high device utilization; the use of capabilities and benefits of VLSI; short-term development and moderate price. The development of the device has been suggested by computing the sum of squared differences using modularity principles, coordination between data flow and computing capability of the device, pipelining and space parallelism, localization and simplification of links with elements. The proposed method can be useful for researchers in the fields of parallel computing and data processing, and can find applications in various fields such as data science, machine learning, image processing, and bioinformatics.

Key words: *sum of squared differences, device, real time, parallel vertical group method, data flow rate, VLSI device, algorithms.*

https://doi.org/10.33108/visnyk_tntu2023.02.005 *Received 22.02.2023*

Statement of the problem. Radio basis function (RBF) networks are used for problem solving in forecasting and classification and management of systems. The peculiarity of RBF networks is in their high speed in study and ability to solve complicated nonlinear problems [1–5]. RBF network is composed of three layers: input layer, hidden layer and output layer. The date vector is extended from the input layer to the neurons of the hidden layer. Their number is correspondent to the number of the selecting centers of RBF network. Thus, each neuron of the hidden layer receives information about the input vector. Radial basis function is calculated in each neuron of the hidden layer hi(x)

$$h_i(x) = \exp[-\frac{(\| x^b - x_i^e \|)^2}{2\delta_i^2}], \quad (1)$$

where x_i^b – input vector, x_i^e – etalon center, δ – spreading parameter for one-dimensional function h_i. The given function preprocesses input vectors and define their close location to

Corresponding author: Ihor Ihnatiev; e-mail: ignatyevki@gmail.com

etalon centers x_i^e, and the meaning $h_i(x)$ sets the strength of association between the input vector x^b and separate etalon centers x_i^e.

Analysis of the available investigations. Hardware implementation supports a high speed of RBF networks for solving problems in a real-time operation mode. Modern element base of RBF networks is the next: full-custom design and semi-custom design very-large-scale integration devices (VLSI), microprocessors, microcontrollers, transputers and neurochips. Hardware RBF networks are mainly realized by hardware algorithm structure for solving problems using a programmable logical device (PLD). Such approach to hardware implementation requires the development of new methods, algorithms and VLSI devices, which realize basic operations of RBF networks. Computing the sum of squared differences belongs to such basic operations:

$$y = \| x_i^e - x_i^b \|^2 = (x_1^e - x_1^b)^2 + (x_2^e - x_2^b)^2 + ... + (x_N^e - x_N^b)^2 \quad (2)$$

The real-time operation mode limits the time of realization of such operation, which can not be over a limit of data flow it means that computation should be done without delays. Hardware implementation can support a high-speed process using pipelining and space parallelism.

Therefore, the development of VLSI devices for computing the sum of squared differences and their utilization is an urgent problem.

The Objective of the work. Requirements and principles for the development of the device for computing the sum of squared differences. The device for computing the sum of squared differences should support [5]:

- high effective utilization;
- effective use of capabilities and benefits of VLSI devices;
- short-term development and moderate price;
- coordination between data flow and computing capability of the device;
- requirements for concrete utilization;
- reduce the number of interface outputs and internal system links;
- real-time operation.

Statement of the task. The device was developed in the ISE Design Suite 14.7 software environment. Based on Spartan 3a programmable logic integrated circuit, which made it possible to simulate the device and check its operation.

Experimental model. The structural organization of the device for computing the sum of squared differences is defined by set of characteristics, the main of them are the following: the number of operands, which are simultaneously processed; the mode of operation; the way of organization of links between processing elements (PE). The devices for computing the sum of squared differences can be divided into synchronic and diachronic according to the mode of operation. In the last case, such devices are called single-cycle because input data are processed without intermediate storage. Speed of operation of a single-cycle device is defined by the time of PE operation, which is the longest data processing. Single-cycle devices are serial from the point of view of realization of algorithm of data processing. Synchronous devices for computing the sum of squared differences should be used for intensive data processing. The computation is done according to the pipeline principle. Pipeline devices are divided into stages by the buffer memory. To support a high-speed operation and an effective use of PE, the simplest operations should be performed with equal time of operation. The results of operations are recorded in the buffer memory in the pipeline device for computing the sum

of squared differences during timing pulses. The frequency of such timing pulses F_{TI} is equal to:

$$F_{TI} = \frac{1}{t_{BM} + t_{OM}}, \quad (3)$$

where t_{BM} – recording time in the buffer memory; *to*– time of operation in PE.

The main goal of the development of the device for computing the sum of squared differences is getting a highly effective, module and a regular VLSI device with computing strength $Ps = F_{TI} s n_s$ coordination between intensity of input data flow $P_d = F_d m n_к$, where F_d – frequency of data flow; s and m – quantity of channels according to data flow and processing; n_s and n_c – channels capacity according to data flow and processing.

Output information for the development of a highly effective device for computing the sum of squared differences is the next:

- number and operands capacity;
- intensity of input data flow P_d;
- requirements to interface;
- accurate computations;
- technical and economic requirements and limits.

Generally, the development of the device for computing the sum of squared differences in real time mode can be defined by the following tasks:

- to define the components, characteristics and quantity of PE;
- to define the parameters of the buffer memory;
- to define necessary links between PE and ways of exchange;
- to synthesize control package;
- to evaluate the instrument parameters.

The process of the development of the device for computing the sum of squared differences in real time mode can be specified by the next stages:

- algorithm development for computing the sum of squared differences and its representation as a concrete coordinated flow graph;
- development of the structure of PE according to the predefined operation code;
- development of the control unit (CU);
- development of links topology, functions of synchronization of exchange between PE and synthesis of the device structure;
- development of the device interface.

The set of the corresponding structures of the algorithm, which consists of the CU and limited set of PE combined with the switching system supporting all technical requirements, is the result of the development.

A high effective utilization of the device for computing the sum of squared differences is achieved by means of minimization of costly equipment supporting the real time mode. Transfer from the algorithm for computing the sum of squared differences to the structure of the device is formally come to minimization of costly equipment supporting the real time mode.

The main ways of minimization of costly equipment during the development of the device for computing the sum of squared differences in the real time mode [6] is the development of algorithm, which supports coordination between the intensity of data flow and computing capacity of the device by the change of duration of pipeline tact $T_C = t_{BM} + t_{OM}$ and capacity n_s of data processing channels.

To evaluate the developed device, the criterion of an effective utilization E is used, taking into consideration the quantity of outputs of interface, uniformity of the structure, the quantity and locality of links, combining productivity with costly equipment and evaluating productivity of elements. The quantitative value of the effective utilization of the equipment for the device for computing the sum of squared differences is defined in the following way:

$$E = \frac{Rmn_{\kappa}}{T_{\kappa} n(k_1 \mathrm{Wd} + \mathrm{k}_2 Q + k_3 Y)} \tag{4}$$

where R – complexity of the algorithm for computing the sum of squared differences; Tp – pipeline tact; n – operands capacity; Wd – costly equipment for the device utilization, k_1 – coefficient of uniformity, k_2 – coefficient of regular links, Q – quantity defined by lines of links, k_3 – coefficient of the quantity of outputs of interface, Y – the quantity of outputs of communication interface.

The following principles were chosen for the development of the device for computing the sum of squared differences in the real time mode:

- the use of the base of elementary arithmetic operations during the development of the algorithm for computation;
- concurrency of the computation process;
- realization of the algorithm for computing the sum of squared differences as the only macro-operation;
- modularity and regularity property;
- localization and reduce of the number of links between processing elements;
- preset architecture using programmable logical devices.

Parallel vertical group method for computing the sum of squared differences. Parallel vertical group method for computing the sum of squared differences [5] requires each operand to be in the form of groups consisting of k capacities. Operands are presented in the following way:

$$X_j = \sum_{i=1}^{n} 2^{-(i-1)} x_{ji} = \sum_{g=1}^{\mathrm{h}} 2^{-(g-1)k} (x_{j[(g-1)k+1]} + 2^{-1} x_{j[(g-1)k+2]} + ... + 2^{-(k-1)} x_{j[(g-1)k+k]}), \tag{5}$$

where x_{ji} – meaning of i-o capacity of j-o operand; n – operand capacity, h – number of groups on which the operand is broken.

Squaring is the base of computation operation of the sum of squared differences. To do such operation, the vertical algorithm is used:

$$\begin{aligned} X^2 &= (0.01) \wedge x_1 + 2^{-1}(0.x_1 01) \wedge x_2 + 2^{-2}(0.x_1 x_2 01) \wedge x_3 + \\ &... + 2^{-(n-1)}(0.x_1 x_2 ... x_{n-1} 01) \wedge x_n = \sum_{i=1}^{n} 2^{-(i-1)} P_i \end{aligned}, \tag{6}$$

where P_i – partial result of squaring, which is defined in the following way:

$$P_i = (0.x_1 x_2 ... x_{i-1} 01) \wedge x_i . \tag{7}$$

Forming a partial result of squaring for a group consisting of *k* capacities of P_{Kg} group partial result of squaring is the development of the investigated algorithm [6–7]:

$$P_{Kg} = P_{g1} + 2^{-1} P_{g2} + \ldots + 2^{-(k-1)} P_{gk} = \sum_{r=1}^{k} 2^{-(r-1)} P_{gr}, \tag{8}$$

where P_{gr} – partial result of squaring.

The algorithm of squaring with the use of forming of group partial results P_{Kg} is presented in the following way:

$$X^2 = \sum_{g=1}^{h} 2^{-(g-1)k} P_{Kg} \cdot \tag{9}$$

The computation of the sum of squared differences is done on the base of multi-operand approach, which is in simultaneous processing of all operands and forming of macro-partial result of the sum of squared differences [3]. The computation of the sum of squared differences will be done using parallel vertical group method resented in the next way:

$$\begin{aligned} y &= (X_1^e - X_1^b)^2 + (X_2^e - X_2^b)^2 + \ldots + (X_N^e - X_N^b)^2 = \Delta X_1^2 + \Delta X_2^2 + \ldots + \Delta X_N^2 = \\ &= \sum_{g-1}^{h} 2^{-(g-1)k} P_{1Kg} + \ldots + \sum_{g-1}^{h} 2^{-(g-1)k} P_{NKg} = \\ &= \sum_{g=1}^{h} 2^{-(g-1)k} \sum_{j=1}^{N} P_{jKg} = \sum_{g=1}^{h} 2^{-(g-1)k} P_{Mg} \end{aligned}, \tag{10}$$

where *N* – number of couples of operands, P_{Mg} – *g*roup macro-partial result of the sum of squared differences.

The structure of the device for parallel vertical group computing the sum of squared differences. Depending on the way of forming and summing of macro-partial results of the sum of squared differences P_{Mg} the following utilization variations are possible:

- serial forming and summing up P_{Mg};
- parallel forming and serial summing up P_{Mg};
- parallel forming and summing up P_{Mg}.

The developed structure of the component using the computation of the sum of squared differences with parallel forming and serial summing up P_{Mg} is presented in Figure1, where R – register, IA*N* – *N*-input adder, A – adder, BS – control unit, PE – processing element.

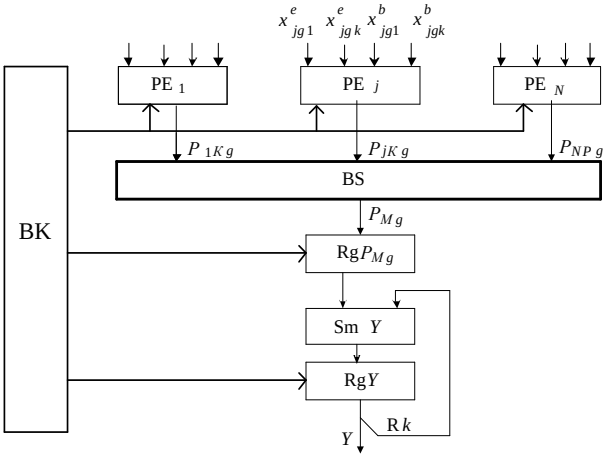


Figure 1. Structure of the device for computing the sum of squared differences

The main elements of the given structure are: PE$_j$ – for forming group partial results of squaring P_{jKg}; IAN – for forming parallel computation of macro-partial result of the sum of squared differences P_{Mg}; AY – supports serial computation of the sum of squared differences using the next formula:

$$Y_g = 2^{-k} Y_{g-1} + P_{Mg}, \tag{11}$$

where Y_0=0.

The structure of PE$_j$ shown in Figure 2, where S (Від) – subtract, Tr – trigger, C (PC) – converter of the parallel code into vertical group one, OM $|\Delta X_j|$ – difference module calculator, F(Ф)P_{rg} – former of partial result of squaring.

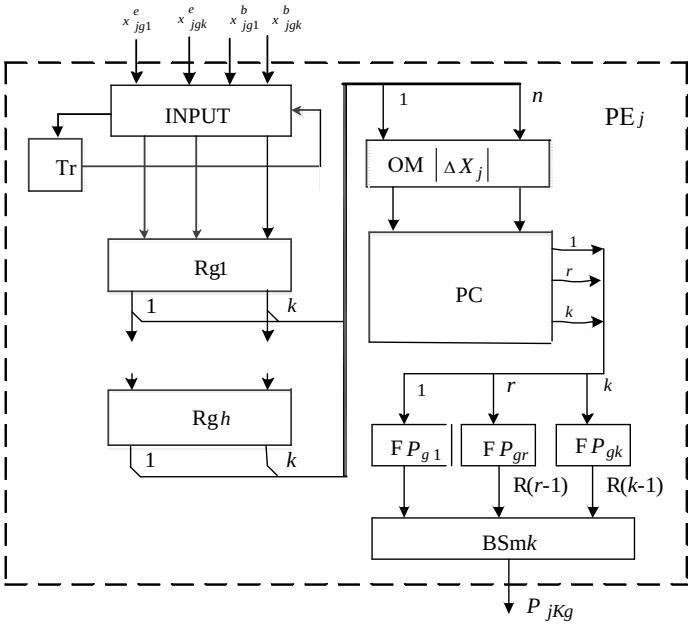


Figure 2. Developed structure of PE

Input operands X_j^e and X_j^b in input of PE$_j$ is serially performed [1, 2] by groups with k capacities starting with lower order bit. In each PE$_j$ using subtract during h cycle, the difference ΔX_j is calculated registered by R1,…,Rh. The computed difference ΔX_i enters the module calculator OM $|\Delta X_j|$ and the result of processing is the module $|\Delta X_j|$. In the next operating cycle, we receive partial results of squaring in former FP$_{rg}$. Forming of partial results of squaring P_{rg} is performed starting with a higher order bit of the module $|\Delta X_j|$ according to the formula (5). Formed k of partial results of squaring P_{rg} enter multiple-input adder MAk shifting to the right in (r-1)-capacities, where they are added. The sum received in outlet of multiple-input adder MAмk is a group partial result of squaring P_{jKg}. Group partial results of squaring P_{1Kg},…, R_{NKg} are added using multiple-input adder MAN. The received sum, which is a macro-partial result of the sum of squared differences P_{Mg}, is registered in RP_{Mg}. In adder AY, in each cycle, the summing up of results from the output of the register RP_{Mg} to the sum accumulated earlier from the register RY shifting to the right in k capacities is performed.

Using the given ways, the summing process is considered as performing the only operation based on the main operation of summation of meanings of bits of bit edge that is a vertical model of computation:

$$Z = \sum_{i=1}^{n} 2^{-i} \sum_{j=1}^{M_i} C_{ji}, \tag{12}$$

where C_{ji} – meaning of capacities; M_i – number of items in i-y bit edge.

The computing process is reduced to transformation of multiserial code into uniserial one by existing vertical methods of computation of operation for group summing up. Such transformation is based on the operation of transformation from three-digit code into two-digit code:

$$\Sigma = \begin{cases} C_{(j-1)1}\ldots C_{(j-1)(n-1)} C_{(j-1)n} \\ + \\ C_{j1}\ldots C_{j(n-1)} C_{jn} \\ + \\ C_{(j+1)1}\ldots C_{(j+1)(n-1)} C_{(j+1)n} \end{cases} = \begin{cases} 0 S_1 \ldots S_{n-1} S_n \\ + \\ P_0 P_1 \ldots P_{n-1} 0 \end{cases}. \tag{13}$$

Transformation of three-digit code into two-digit one is performed using the layer of single-bit adders, which are not linked. To cut the time of transformation of multiserial code into uniserial layers of single-bit adder, it is necessary to combine it according to the principle of Wallace tree.

Software ISE Design Suite 14.7. was used for simulating multiple-input adders. The developed model of multiple-input adder presented in Figure 3 was developed on the base of the given software.

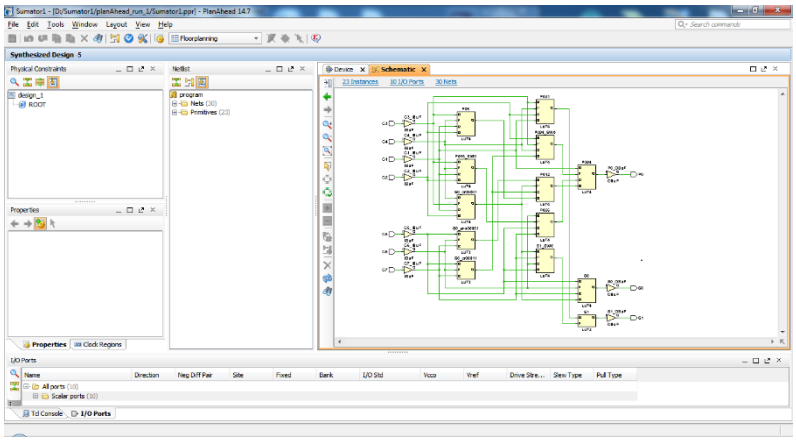


Figure 3. Model of multiple-input adder

In software Xilinx, the scheme of location of input and output ports on the crystal of programmable logical device was developed. Figures 4 and 5 illustrate this scheme.

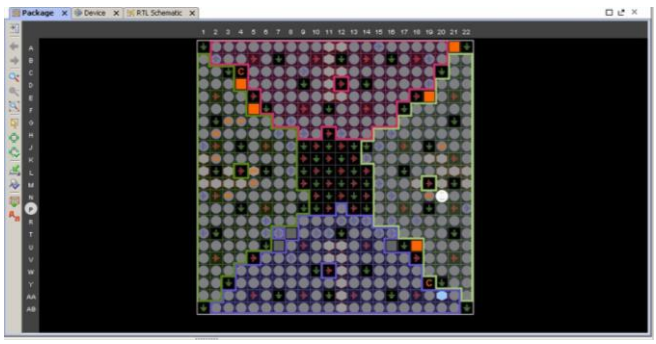
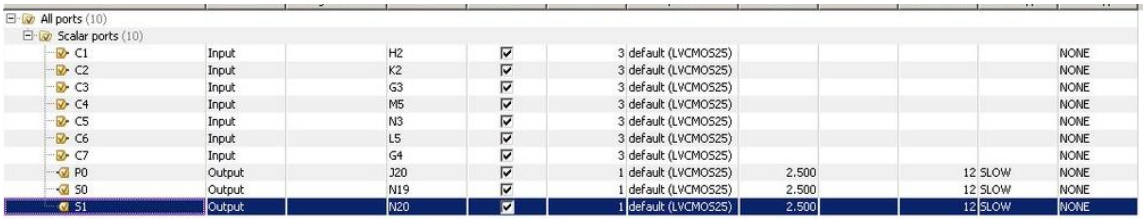


Figure 4. Location of input and output ports on the crystal



All ports (10)									
Scalar ports (10)									
C1	Input		H2	☑	3 default (LVCMOS25)				NONE
C2	Input		K2	☑	3 default (LVCMOS25)				NONE
C3	Input		G3	☑	3 default (LVCMOS25)				NONE
C4	Input		M5	☑	3 default (LVCMOS25)				NONE
C5	Input		N3	☑	3 default (LVCMOS25)				NONE
C6	Input		L5	☑	3 default (LVCMOS25)				NONE
C7	Input		G4	☑	3 default (LVCMOS25)				NONE
P0	Output		J20	☑	1 default (LVCMOS25)	2.500		12 SLOW	NONE
S0	Output		N19	☑	1 default (LVCMOS25)	2.500		12 SLOW	NONE
S1	Output		N20	☑	1 default (LVCMOS25)	2.500		12 SLOW	NONE

Figure 5. Initialization of ports

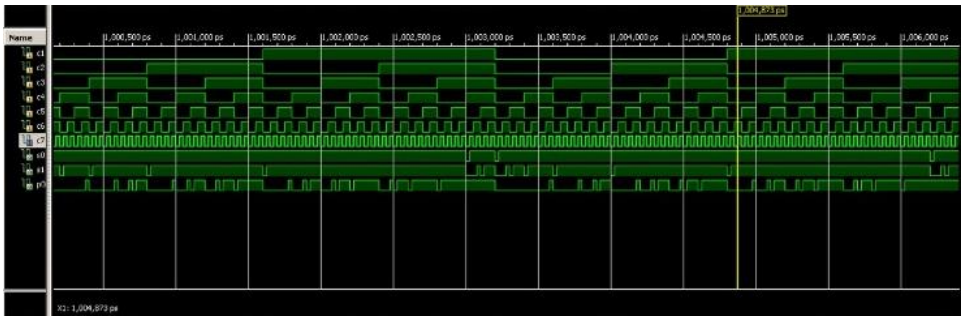


Figure 6. Time picture of the device performance

Combination of collecting and computing of data files is the specific feature of the performance of the developed device for computing the sum of squared differences. Such combination for computing the sum of squared differences in the given device is used during *h* cycles.

Conclusions. The development of the device for computing the sum of squared differences was suggested according to the following principles: modularity, coordination between data flow intensity and computing capability of the device, pipelining and space parallelism, localization and simplification of links between elements. The parallel vertical group method for computing the sum of squared differences using multi-operand approach and basing on forming and summing up macro-partial results, the number of which is defined by digit capacity of groups of operands entry has been developed. The parallel vertical group method for computing and the base of elementary arithmetic operations, which increases the speed, reduces costly equipment and orients on realization of very-large-scale integration devices have been used for the device utilization. It has been proved that the increase of an effective use of VLSI devices for computing the sum of squared differences can be achieved by partial or complex use of methods, which support cut of time for forming and summing up macro-partial results. The multi-input adders using software ISE Design Suite 14.7 and the device for computing the sum of squared differences have been simulated.

References

1. Tsmots I., Rabyk V., Skorokhoda O., Teslyuk T. Neural element of parallel-stream type with preliminary formation of group partial products. Electronics and information technologies (ELIT-2019) : proceedings of the XIth International scientific and practical conference, 16 –18 September, 2019, Lviv, Ukraine. 2019. P. 154–158. https://doi.org/10.1109/ELIT.2019.8892334
2. Tsmots I. H., Lukashchuk Yu. A., Khavalko V. M., Rabyk V. H. Modeli neiropodibnoho elementa paralelno-paralelnoho typu. Modeliuvannia ta informatsiini tekhnolohii. 2019. Vyp. 86. P. 119–126/
3. Tsmots I., Teslyuk V., Teslyuk T., Ihnatyev I. Basic Components of Neuronetworks with Parallel Vertical Group Data Real-Time Processing. Advances in Intelligent Systems and Computing II, Advances in Intelligent Systems and Computing 689. Springer International Publishing AG 2018. P. 558–576. https://doi.org/10.1007/978-3-319-70581-1_39
4. Wu R, Guo X, Du J, Li J (2021) Accelerating neural network inference on FPGA-based platforms – A survey. Electronics 10:1025. URL: https:// doi. org/ 10. 3390/ elect ronic s1009 1025. https://doi.org/10.3390/electronics10091025

5. Sze M., Chen S., Yang Y. and Huang T. S. “Efficient Processing of Deep Neural Networks: A Tutorial and Survey,” Proceedings of the IEEE. Vol. 105. No. 12. P. 2295–2329, Dec. 2017. https://doi.org/10.1109/JPROC.2017.2761740
6. Chen T., Du Z., Sun N., Wang J., Wu C., Chen Y. and Temam O. “DianNao: A Small-Footprint High-Throughput Accelerator for Ubiquitous Machine-Learning,” Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). P. 269–284, Mar. 2014. https://doi.org/10.1145/2541940.2541967
7. Zhang Y., Chen T., Du S. S. and Wang J. “Maximizing CNN Accelerator Efficiency through Resource Partitioning and Pipeline Parallelism,” Proceedings of the 2016 ACM SIGARCH International Conference on Computer Architecture (ISCA). P. 573–586, Jun. 2016.
8. D. H. D. Zhou, Y. Zhang, Z. Zhou, and J. Cong, “FPGA-Based Deep Learning Accelerator with Stacked Sparse Autoencoder,” Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA). P. 26–35, Feb. 2016.
9. Yuan Wang, Chen-Yi Lee, and Tsi-Chung Chen. “Parallel Implementation of Sum-of-Squares-of-Differences for Image Matching.” IEEE Transactions on Circuits and Systems for Video Technology. Vol. 26. No. 9. 2016. P. 1711–1721. https://doi.org/10.1109/TCSVT.2015.2462012
10. Rajib Dey, Sushmita Roy, and Somnath Paul. “Efficient Hardware Implementation of Sum of Absolute Difference and Sum of Squared Difference for Real Time Video Processing.” 2018 International Conference on Signal Processing and Communications (SPCOM), 2018, p. 1–5.
11. D. V. Le, D. T. Anh, T. Q. Anh, and N. T. Thanh. “A Novel Fast and Low Power Sum of Squared Differences Architecture for Motion Estimation in Video Coding.” 2017 7th International Conference on Communications and Electronics (ICCE), 2017, p. 11–16.
12. F. B. Shams, S. A. Samad, and S. A. Samad. “FPGA Based Parallel Architecture for Sum of Absolute Differences and Sum of Squared Differences Using Novel Pipelining.” 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017, p. 63–68.
13. Trung-Kien Le, Thanh-Tung Do, Van-Anh Nguyen, Thanh-Binh Nguyen, and Duc-Minh Pham. “Design and Implementation of High Performance Sum of Absolute Differences and Sum of Squared Differences Circuits for Video Coding.” 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2018, p. 226–229.
14. Jiaqi Yan, Zhaohui Yang, Shuai Zhang, Qingyu Hou, and Junzhao Du. “A Novel Algorithm and VLSI Architecture for Sum-of-Squares-of-Differences in Image Matching.” Journal of Signal Processing Systems. Vol. 89. No. 3. 2017. P. 465–478.
15. Yi-Fan Lin and Chen-Yi Lee. “A Low-Power Parallel Processing Architecture for Sum-of-Squared-Differences-Based Image Matching.” IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Vol. 26. No. 10. 2018. P. 1925–1937.
16. Xinyu Liu, Jianpeng Xue, Hailiang Zhang, and Xiande Huang. “An Efficient Reconfigurable Hardware Architecture for Sum of Squared Differences Algorithm.” 2018 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2018, p. 1–6.
17. M. Emre Celebi and Yasemin Yardimci. “A Hardware Design of Sum of Squared Differences and Its Application on Stereo Matching.” 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018, p. 1–4.
18. Tsmots I., Teslyuk V., Kryvinska N., Skorokhoda O., Kazymyra I. Development of a generalized model for parallel-streaming neural element and structures for scalar product calculation devices. Journal of Supercomputing. 2022. https://doi.org/10.1007/s11227-022-04838-0

УДК 681.3

ПРИСТРІЙ ПАРАЛЕЛЬНО-ВЕРТИКАЛЬНОГО ГРУПОВОГО ОБЧИСЛЕННЯ СУМИ КВАДРАТНИХ РІЗНИЦЬ

Іван Цмоць[1]; Ігор Ігнатєв[2]; Степан Івасьєв[2]

[1]Національний університет «Львівська політехніка», Львів, Україна
[2]Західноукраїнський національний університет, Тернопіль, Україна

Резюме. *В основу методу покладено груповий підхід, який дозволяє розділити завдання на кілька підзадач і розраховувати їх паралельно. Розглянуто задачу обчислення суми квадратів різниць*

між елементами великих масивів даних. Застосування традиційних методів обчислення таких сум у паралельних середовищах може бути неефективним через обмін великими обсягами даних між вузлами. Запропонований метод дозволяє зменшити обсяг даних, що передаються, і підвищити ефективність обчислень. Запропоновано новий метод обчислення суми квадратів різниць, що дозволяє підвищити ефективність обчислень у паралельному вертикальному середовищі. Тестування методу на різних наборах даних показує його високу ефективність порівняно з традиційними методами обчислення сум квадратів різниць у паралельних середовищах. Запропонований метод може бути застосований у різних сферах, що вимагають опрацювання великих обсягів даних, і дозволяє підвищити ефективність обчислень і скоротити час їх виконання. Проаналізовано методи, алгоритми та структуру пристроїв обчислення суми квадратів різниць, визначено їх недоліки. Визначено, що пристрій для обчислення суми квадратів різниць повинен підтримувати: високе використання пристрою; використання можливостей і переваг НВІС; короткотерміновий розвиток і помірну ціну. Розроблення пристрою запропоновано шляхом обчислення суми квадратів різниць із використанням принципів модульності, координації між потоком даних і обчислювальними можливостями пристрою, конвеєрного та просторового паралелізму, локалізації та спрощення зв'язків з елементами. Запропонований метод може бути корисним для дослідників у галузі паралельних обчислень і опрацювання даних, а також може знайти застосування в різних галузях, таких, як розпаралелення даних, машинне навчання, опрацювання зображень і біоінформатика.

Ключові слова: *сума квадратів різниць, пристрій, реальний час, метод паралельного вертикального підсумовування, потоки даних, пристрій НВІС, алгоритми, ПЛІС.*

https://doi.org/10.33108/visnyk_tntu2023.02.005 *Отримано 22.02.2023*