PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Multifunctional image processor based on rank differences signals weighing-selection processing method and their simulation

Vladimir G. Krasilenko, Alexander A. Lazarev, Diana V. Nikitovich

Vladimir G. Krasilenko, Alexander A. Lazarev, Diana V. Nikitovich, "Multifunctional image processor based on rank differences signals weighing-selection processing method and their simulation," Proc. SPIE 11187, Optoelectronic Imaging and Multimedia Technology VI, 111871Q (18 November 2019); doi: 10.1117/12.2538468

Event: SPIE/COS Photonics Asia, 2019, Hangzhou, China

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019 Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

Multifunctional image processor based on rank differences signals weighing-selection processing method and their simulation

Vladimir G. Krasilenko*[a], Alexander A. Lazarev[b], Diana V. Nikitovich[b]
[a]Vinnitsa Social Economy Institute, 86/131 Kelestskaya Str., 21021, Vinnitsa, Ukraine;
[b]Vinnitsa National Technical University, Ukraine.

ABSTRACT

A new iterative process for sorting array of signals, which differs from the known structures by uniformity and versatility, and allows direct and inverse sorting of analog or digital signal arrays was proposed in this paper. Simple relational nodes are basic elements of the proposed sorting structures. Such elements can be implemented on a different element basis, for example, on devices of selecting a maximum or minimum of two analog or digital signals, which can be implemented on CMOS current mirrors and carry out the continuous logic limited difference function. The homogeneous sorting structure on such elements implementation, consisting of two layers and a multichannel sampling and holding device was offered. Nine signals corresponding to a selection window of a matrix sensor are fed to this structure, and are sorted by five iterative steps, and at the output we receive the signals sorted by the rank, which, using the code controlled programmable multiplexer, generates an output signal, that corresponds to the selected rank. Technical parameters of such relational preprocessor were evaluated. The paper considers results of design and modeling of CL BC based on current mirrors (CM) for creating picture type image processors (IP) with matrix parallel inputs-outputs. Such sorting nodes have a number of advantages: high speed and reliability, simplicity, small power consumption, high integration level. The inclusion of an iterative node for sorting signals into a modified nonlinear IP structure makes possible to significantly simplify its design and increase the functional capabilities of such processor. The simulation results confirm the proposed approaches to the design of sorting nodes of analog signals of the iterative type. The power consumption of the processors does not exceed 2mW, the response and processing times are 10μs and can be less by an order of magnitude, the supply voltage is 1.8÷3.3V, and the operating currents are optimally in the range of 10÷20μA. The energy efficiency of the proposed preprocessor with the iterative sorting node is $25x10^9$ operations per second per watt, which corresponds to the best technical solutions. In the work we show, that after sorting or comparative analysis of signals by levels, a promising opportunity appears to implement image processors with enhanced functionality using the new method of weighting-selecting rank differences of signals. The essence of the method is that by composing the differences of the signals ordered by rank and the upper level of their range, we can simultaneously form several resulting output signals, choosing the necessary difference signals from their set according to the control commands and weighing them additionally before the summation. We show that using this approach and the method of processing we can significantly expands the set of operations and functions for image filtering, simplifying hardware implementation of IP, especially for analog and mixed technologies. We determined set of basic executable instruction-functions of the processors, presenting the simulation results in Mathcad, PSpice OrCad and other environments. We discussed the comparative evaluation of various modifications and options for implementing processor. We analyzed the new approach for the programmable selecting function or set of functions, including the selecting the required differences between the ranks of signals and their weights. We show the results of design and modeling the proposed new FPGA-implementations of MIP. Simulation results show that processing time in such circuits does not exceed 25 nanoseconds. Circuits are simple, have low supply voltage (2.5 V), low power consumption (50mW), digital accuracy. Calculations show that in the case of using an Altera FPGA chip EP3C16F484 Cyclone III family, it is possible to implement MIP with register memory for image size of 64*64 and window 3*3 in the one chip. For the chip for 2.5V and clock frequency 200MHz the power consumption will be at the level of 200mW, and the calculation time for pixel of filters will be at the level of 25ns.

Keywords: analog relational preprocessor, image nonlinear processing, simulation, continuous logic, current mirror, sorting networks, methods of selection and rank preprocessing, rank filtering.

* krasvg@i.ua

Optoelectronic Imaging and Multimedia Technology VI, edited by Qionghai Dai, Tsutomu Shimura, Zhenrong Zheng,
Proc. of SPIE Vol. 11187, 111871Q · © 2019 SPIE · CCC code: 0277-786X/19/$21 · doi: 10.1117/12.2538468

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

1. INTRODUCTION

The 20th century can rightfully be called the Boolean, since Boolean two-valued logic covers almost the entire binary-discrete (virtual) world, including all the variety of means and fields of application of information technologies. Despite this, in the vast majority of cases, various manufacturing processes and technologies of the physical macrocosm, especially those related to measurement and control processes, are accompanied not by discrete, but by accompanying continuous (continuous analog) processes and signals[1]. And for the successful development of more generalized continuous information technologies and artificial intelligence[2, 3], both from a philosophical and mathematical point of view, it is necessary to strengthen the semantic power of logical-algebraic calculi, introduce their new basic definitions, concepts and, at least on an intuitive level, use them to describe and simulate machine-mechanical analogues of thought processes[4]. All this led to contradictions in the use of discrete logics in the continuum, in order to resolve which, to describe and model each continuous subject area and class of its problems, their own formal logical-algebraic (LA) apparatuses and calculi in the form of mathematical (symbolic) subject oriented logics and special algebras. The basis of information technologies in the analog field is precisely the continual LA calculus: Lukasiewicz's infinite-valued logic[5], continuous logic with all its variants and generalizations[6, 7], additive-multiplicative logic (AM) algebra[8], predicate selection algebra[9], equivalence algebra[10] and others. They determined the continual biologically inspired stage of development of LA-calculus and a new, more energy-efficient, direction of building models and their hardware implementations of artificial intelligence. The basis of matrix logics and the emerging ability to integrate an array of photo-detectors with digital electronics in silicon make it possible to implement an intelligent pixel-like stack architecture of a three-dimensional chip and promising matrix sensors and measuring and computing systems, the first and many demonstration systems of which have already been created[11-23.] Since the circuits of smart detectors from any semiconductor fabric are almost already created to prove the use of optics or optoelectronics for external and internal interconnections and microcircuits, our approach prefers an intelligent pixel-like architecture. Since each pixel of such architecture, having its own analog (analog-to-digital node) with memory, combines parallel signal detection with parallel signal processing in one circuit, it guarantees the fastest processing, if you also provide parallel input-output. .Note that many logics, models and image processing algorithms, artificial neural-fuzzy systems, their basic procedures of composition-decomposition and fuzzy inference are based on multi-place, multi-input mini-max operations: min (x_1, x_2,…. x_n) and max (x_1, x_2, …. x_n), when defining variables on a unit interval[6, 7] : $x_i \in [0,1]$. Parallel high-speed processing of a 1D or 2-D array using non-traditional MIMO systems, the corresponding matrix logics (multi-valued, continuous, neural-fuzzy and others) and the corresponding mathematical apparatus becomes a strategic direction[1÷23]. In works[16-18, 20-22] the fundamentals of designing universal (multifunctional) logic elements (ULE) of a matrix logical structure with fast programmable tuning were considered. The use of architectures with time-pulse-coded analog variables (TPCA), which were considered in works[16-18, 20, 22] and based on a generalized unified methodological basis for creating a whole family of universal (multifunctional) elements of various logics, although they provide many advantages regarding the expansion of the set of functions and their programmable samples do not satisfy the time requirements while maintaining a satisfactory dynamic range in terms of accuracy. Such a time-pulse-encoded representation of matrix continuous logic variables by two-level signals, although it allows you to expand the functionality and simplify the tuning circuit for the required function, does not satisfy the increased requirements for speed and accuracy of calculations. Therefore, there is an urgent need to improve the hardware implementations of nodes that perform these and similar operations, especially for those applications where the number of operations and input variables are significant. Arrays of matrix elements of continuous and threshold, rank, and order logic are also required for promising implementations of future computing architectures based on 3D chips, including optical training neural networks (NN) with a two-dimensional structure[14], equivalence models (CLEM) NN[15-17], especially to create convolutional and new self-learning equivalence-convolutional structures[24-27]. A special place among the methods of effective high-speed image processing working with arrays is occupied by the class of structured nonlinear methods and algorithms that performed the conversion of the form:

$$\mathbf{B}=\{b_{ke}\}=\mathbf{F}(\mathbf{A})=\{\mathbf{\Phi}_{\kappa l}(\mathbf{A}_{\kappa l})\},$$

where $\mathbf{\Phi}_{\kappa l}(\mathbf{A}_{\kappa l})$ — nonlinear function which is determined by subset of rank and (or) index statistician of selection. It is formed by signal samples from some neighboring of this element in the sequence of the well-organized samples of signals[28]. By virtue of the last this subclass was adopted by rank algorithms. The algorithms of extreme filtration, using values of minimum and maximum on samples of neighborhood space, are the special cases of the rank algorithms. Any r-th index statisticians $v_s(\mathbf{r})$ of display (k, l) the set neighboring of which form other (N_s-1) the elements of selection it is possible to bind to the local histograms of distributing of values of neighboring elements and with the proper functions of the well-organized choice F_n^m ($\vec{x}$) element, where $\vec{x}$ =(x_1, x_2, …. x_n). Such functions at any values of changing

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

variables choose that size which at the location all right not decreasing are occupied by m-th place. These functions can be represented by a logical formula[6]:

$$F^{(r)}(x_1,\dots.x_n)=x^{(r)}\ , r=\overline{1,n}\,,$$

where r-rank of the base operations of continuous logic (CL)[7]. Thus for r = n this operation passes to n-local disjunction, for r=1 to n-local conjunction. The algebra formed in a number of C = [0, 1] with base operations f(r) and complementarity operation (-) is named ordering Boolean algebra. The row of specific laws is inherent to it: tautology, commutative, distributive[7]. As examples of functions of ordinal logic can be the median and inversion of the median, as well as a number of others[7]. Rank algorithms are locally-adaptive on the same essence: simplicity of local adaptation, invariance to spatial links and to signals dimension, almost algorithms complication independence from the sizes of neighboring. Also, at calculation of concrete ranks the simplifications related to informative surplus of images are possible. The proposed models[29] of neuron classifiers and identification algorithms based on a hierarchical description of the forms of lattice functions allow one to theoretically quite subtly isolate and compare these forms. Such models also require the use of similar operations of order logic and the like. Thus, analysis of foregoing allows drawing conclusion about the necessity of development of hardware sorting devices with the sufficient number of inputs (executing «ordering» of input values and supervisory after the well-organized locations (variables). Known wave structures that order the ranking of the analog or digital variables and their base cells that forms such converter-orders[29], as association of two logical charts: charts of selecting of a less value min(x1, x2) and charts of function max(x1, x2). Besides, in works[7, 30-32] it has been shown that the operation of min, max of continuous logic (CL) are the basic operations and that some operations of CL, such as equivalence and nonequivalence, which are realized by various means[16-18, 21-23, 27], and their generalized family[32], allow to receive a number of advantages in so identified «equivalence to a paradigm» neural-network models[10, 13, 15] and array of CL ADC[23, 30, 31]. The use of CL-transformations and CL functions (CLF) made it possible to create promising energy-efficient analog-to-digital converters for intelligent detectors on their basis[23, 30, 31]. But the implementation of such continuously logical base cells (CL_BCs) based on current mirrors (CMs) for CL ADC and for analog sorting networks, proposed in works[30, 33], which perform the operations min (x1, x2) and max (x1, x2), is rather complicated for digital variables[6] and although simple, only 13÷20 CMOS transistors, their number to build a traditional know wave network structure of such cells, especially at large values ($n \approx 9 \div 25$), is very large. At the same time, an increase in the depth of the structure, the number of layers of CL BCs, all the same increases both the complexity and the sorting time. But at the same time with an increase in the number of analog BCs accuracy decreases, so more accurate implementations need to be sought. To create fully parallel algorithms and tools for image processors[31, 32, 34] morphological image processing[35-38], especially for implementing such basic operations as dilation, erosion, opening, closing[38], etc., the above mentioned min-max operations on sets of signals are also necessary, which represent structural windows or selected fragments processed images. Many of the above-mentioned morphological operations need to be repeated many times and for all the current fragments of the image being processed, therefore, there is an urgent need to reduce the execution time and the underlying min-max operations and ranking operations. Sorting algorithms have been widely researched due to the need for sorting in many applications [39-42]. Sorting algorithms have been specialized for particular sorting situations, such as, high-speed sorting[43], sorting using a single CPU and multiple CPUs, parallel image and big data processing[44]. To create more advanced intelligent sensors and image processing processors[32, 34] with enhanced functionality and combining analog or analog-digital preprocessing of signals in a dedicated and accessible structural area (window) also requires sampling, storage, sorting and selection of signals. In papers[33, 45-46] approaches to creation of programmable relational optoelectronic processors as base elements for sorting networks were shown. But for such relational processors based on sorting devices working both with analog signals with amplitude and time coding, and with digital signals, including iterative sorting structures for ordering signals, the image processing algorithms themselves have been poorly modeled using such processors. Therefore, the aim of this work is to perform a series of model experiments on specific images and demonstrate the essence of the transformation processes and the advantages of multifunctional processors using such well-known improved sorting nodes. But for such relational processors based on sorting devices working both with analog signals with amplitude and time coding, and with digital signals, including iterative sorting structures for ordering signals, the image processing algorithms themselves have been poorly modeled using such processors.

Therefore, the goal of our work is to search for compromise new options for implementing both signal sorting nodes, including analog and digital, providing increased accuracy and speed, and based on them relational non-linear image processing processor with advanced functionality. And since such processors can be used as multifunctional nodes of ordinal logic, extremum selectors, nodes of ordering and sorting data, rank filters, recognizers of fragment classifiers, etc., the question of their implementation is acute. In addition, taking into account the recent emergence of a new

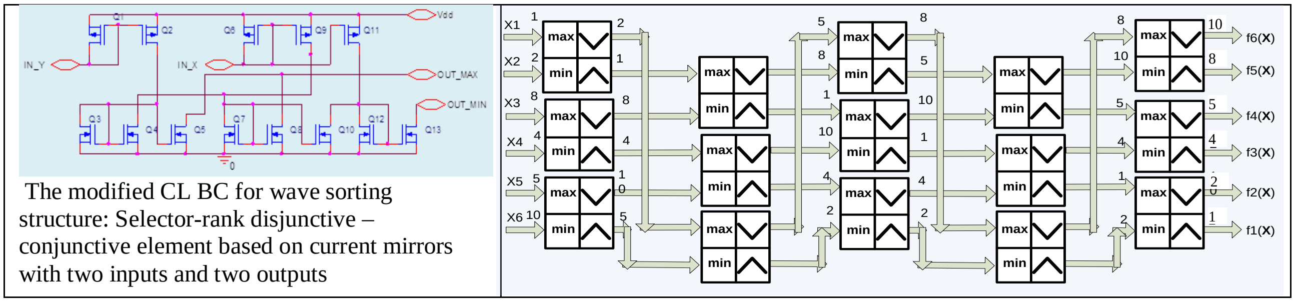
Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

element base, our task is to prove the possibility of creating on the FPGA, practically in one chip, an image preprocessor (IP) with enhanced technical characteristics and a wide range of commands through the use of a new method of processing pre-ranked signals and (or) their differences. To achieve this goal, it is necessary to simulate the algorithms and methods themselves, and then based on them design and simulate the technical options for the implementation of non-linear image processing processors and their main nodes, including sorting nodes, which significantly expand the functionality and range of tasks solved by such multifunctional relational preprocessors images.

2. A BRIEF OVERVIEW OF NONLINEAR PROCESSING RELATIONAL PREPROCESSOR BASED ON SORTING NODE

2.1 Structure of the basic relational preprocessor (BRP) of nonlinear image processing

The structure of the basic relational preprocessor (BRP) of nonlinear image processing and its modified conveyor homogeneous with regular connections wave structure (MCHWS), which performs sorting and ordering of analog or digital signals and is one of the main processor nodes, are shown in Fig. 1 and were considered in works[33, 45]. In work[33] considered analog processors based on the main sorting unit (SU) of analog or digital signals, consisting of a conveyor of layers of selector-rank disjunctive-conjunctive elements (SRDCE) , the specifics and implementation features of which depend on the type and form of signal representation. The modified wave structure used SRDCEs or CL BCs with many ordered outputs. Comparative analysis of structures of ranging of analog signals (SRAS) shows the wave structures require less of equipment, along with other advantages (regularity, homogeneity, base cells only of 2 forms and 2 CLFs). This winning is especially and is increased at growth of number of variables n. Therefore our approach oriented to wave structure is special important at considerable n and provides integration at the requirement of multi-channeling of such devices, especially for time-coded signals as CL-variables[16, 33, 45]. The SRDCE circuits of only 13 CMOS transistors (Ts), and adding 4 or 6 Ts to them, which compare the currents, it is easy to get a digital potential output of the comparator. The SRDCE as base cell executing organization of two analog optical signals that in fact is two currents was designed and simulated[45, 46]. As can be seen from Fig. 1, the well-known wave structure[33, 45] has (n-1) layers consisting of completely identical BCs, the number of which is determined by rounding to integer {n/2}, where n is the number of input variables. If this number is an even number, for example, n = 6, then in each of the five layers there will be 3 basic cells, and there will be 15 of them in total (Fig. 1). Note that for more homogeneity and regularity of bonds in the structure, it is advisable to choose n even. Therefore, for image processing, even with a minimum window size of 3x3, it is necessary to make a structure of 45 basic cells, since: (10/2)x(10-1) = 45. Already with a window of 7x7, you will need ((49+ 1) / 2) x (49-1) = 25x48 = 1200 cells, which greatly complicates the structure. In addition, due to the increase in the number of layers, the delay time of signals and errors due to their accumulation also increase. The 4-input sorting structure of analog signals using of 6 the modified CL BCs based on 2-input and 2-output SRDCEs is represented in work[33], in Fig. 2 and explains the essence of analog signal processing (pixels of the current window), with their preliminary sorting. It executes sorting of these 4 signals represented by currents and is in fact the device of order logic, calculating simultaneously all operations of the proper ranks r = 1÷4. The multiplexer allows the control code (y_1, y_2, ... y_n) to choose the rank n and, accordingly, the type of required operation or function.



The modified CL BC for wave sorting structure: Selector-rank disjunctive – conjunctive element based on current mirrors with two inputs and two outputs

Fig. 1. The modified CL BC (left) of BRP with N inputs and output; SU based on MCHWS (right) for п =6 (SU have n-1=5 layers consisting of completely identical CL BCs (SRDCEs)

2.2 Structure of the modified relational preprocessor (MRP) with iterative sorting node[46]

The modified iterative sorting structure based on a multichannel sampling and storage device and two linear arrays, consisting of basic continuous-logic analog cells (disjunctive-conjunctive elements of the selector rank (SRDCE) with ordered outputs) is shown in Fig. 3 (left). Details of the structure and their base cells will be discussed in the report[46]. The simulation results of proposed MRP with PSpice Orcad are shown in Fig. 3, 4 for different modes of operation of the

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

iterative sorting node. They show that for used 1.5μm CMOS transistors (Ts), the total sorting time of 10 signals (9 input information variables and one auxiliary) with permissible errors does not exceed 6÷18μs (for evaluation, we take 10μs). This time is made up of the five required clocks, but the rewriting beat in the SHD and the read beat can be different. We doubled the last one and therefore the total time was proportional to 6 cycles. The levels of input signals in the figures are indicated by different colors, which allow you to see the dynamic of transitions and the change of signal levels during exchanges, permutations. At the inputs we gave signals, ordered by their levels in the reverse order. This made it possible to more clearly demonstrate the process of ranking in which the signal with the highest level appeared at the top output of the circuit. Let us estimate the complexity of such a sorting node. Each SHD consists of 16 Ts, there are only 10. And the two lines (layers) of basic cells with min-max operations (comparisons and exchanges in essence!) consist of 10 cells, each of which is performed on 13 Ts. Therefore, the total number of Ts will be equal to: 16x10+13x10= =290.

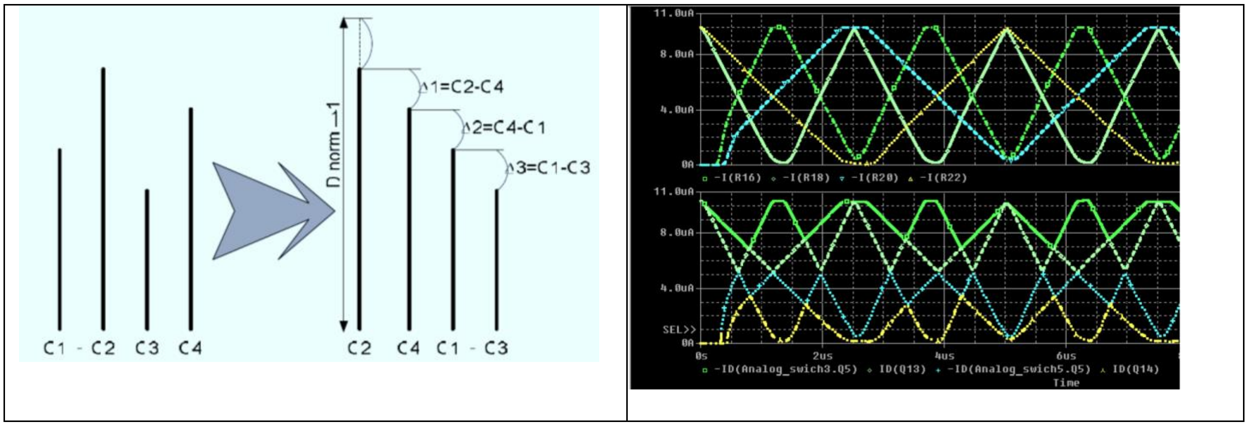


Fig. 2. Graphical representation of the processor operations (left); Simulation results of the 4-input sorting structure of analog signals using of 6 the modified base analog cell based on 2-input and 2-output SRDCEs for small currents (right)

Taking into account the presence of some other auxiliary circuits: clock signal generators, a multiplexer and matching buffers; we can assume that only up to 400 Ts will be needed. Even for the fastest and most advanced algorithms and sorting schemes, the total number of comparison and exchange operations is proportional to (nlogn)x1.5 and for n=10 is about 50. Thus, taking into account that for the simulated circuit the power consumption was 2mWs and Tproc = 10^{-5}s, we obtain for the simultaneous formation of ten output functions the energy efficiency estimate at the level: 500op/(10^{-5}s · 2x10^{-3} W)=25x10^{9} op / s · W. And this means that at least several hundred of them could be placed on the chip. Structure of 1D array 8 bit CL_ADCs with analog signals preprocessing was described[32]. It uses the same SHD, similar cells and iterative approaches, and this allows for additional in such processors to implement AD-transformation, both before and after sorting the signals.

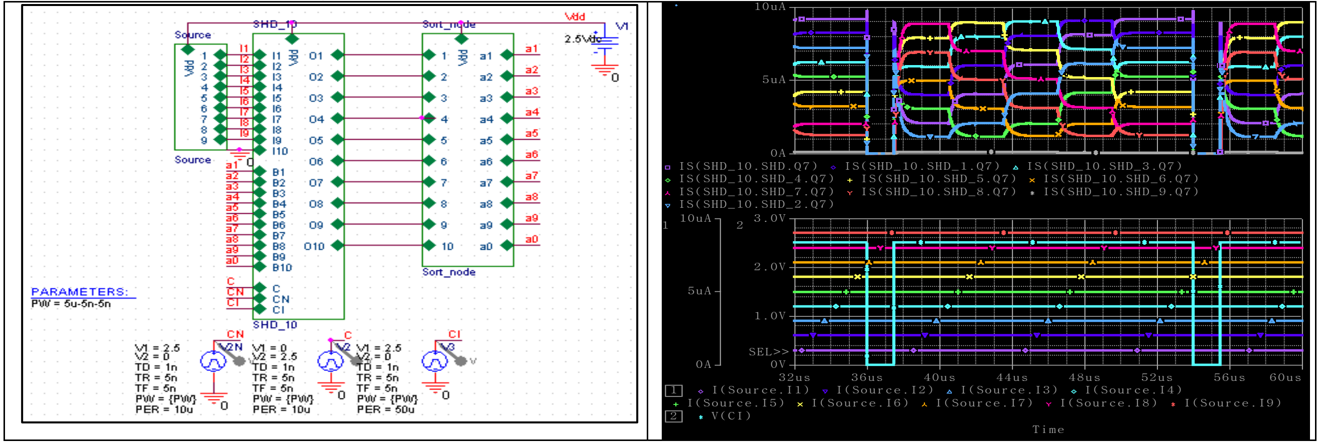

SHD_10
Sort_node
Source
Vdd
V1
2.5Vdc
PARAMETERS:
PW = 5u-5n-5n
V1 = 2.5
V2 = 0
TD = 1n
TR = 5n
TF = 5n
PW = {PW}
PER = 10u
CN
C
CI
10uA
5uA
0A
IS(SHD_10.SHD.Q7) IS(SHD_10.SHD_1.Q7) IS(SHD_10.SHD_3.Q7)
IS(SHD_10.SHD_4.Q7) IS(SHD_10.SHD_5.Q7) IS(SHD_10.SHD_6.Q7)
IS(SHD_10.SHD_7.Q7) IS(SHD_10.SHD_8.Q7) IS(SHD_10.SHD_9.Q7)
IS(SHD_10.SHD_2.Q7)
3.0V
2.0V
1.0V
0V
SEL>>
32us
36us
40us
44us
48us
52us
56us
60us
I(Source.I1) I(Source.I2) I(Source.I3) I(Source.I4)
I(Source.I5) I(Source.I6) I(Source.I7) I(Source.I8) I(Source.I9)
V(CI)
Time

Fig. 3. A modified iterative sorting structure based on a multichannel SHDs and two linear arrays consisting of basic continuous-logic analog cells (disjunctive-conjunctive elements of the selector rank with ordered outputs) (left); Simulation results of iterative sorting node for Vdd=2.5V, Dmax=10μA, T=18μs (right).

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

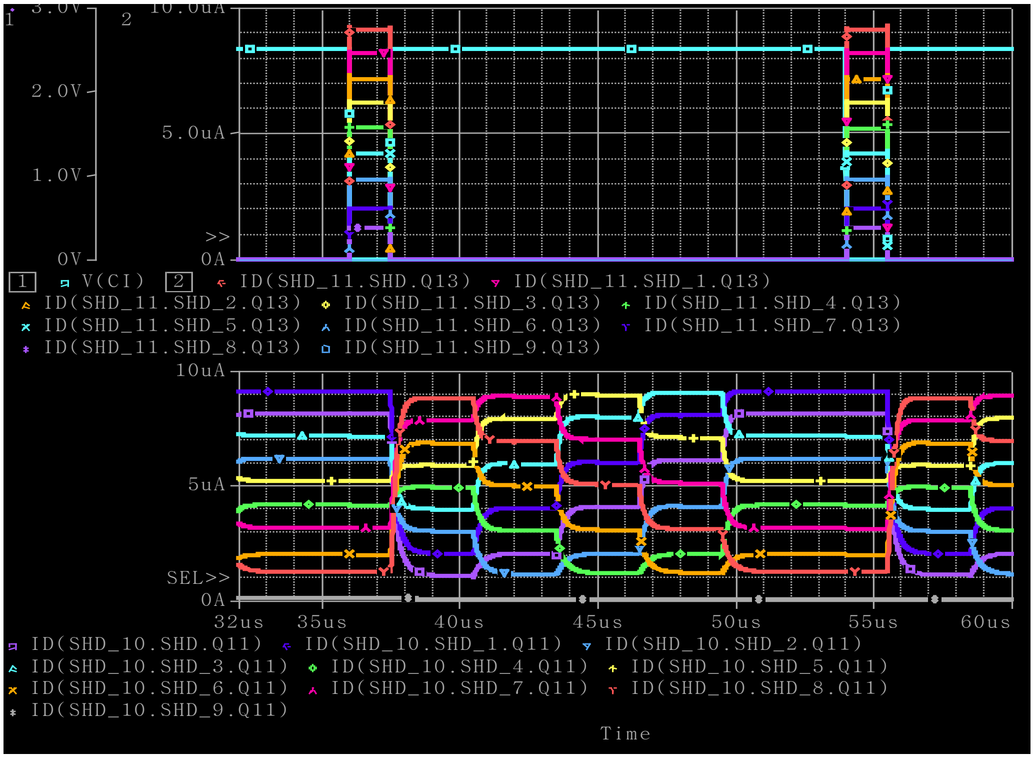


Fig. 4. Simulation results of iterative sorting node for Vdd=2.5V, Dmax=10μA, T=18μs in the case of one ramp / falling signal and nine constants

2.3 Digital modified image preprocessor (DMIP) based on FPGA with conveyor sorting node[47]

For the convenience of data input, we have developed and modeled a digital modified image preprocessor (DMIP) based on FPGA with sorting node circuit with register memory for fast sequential image input and automatic sequential search of processed windows. Structure of DMIP based on FPGA with 10 inputs and 1 output; SU based on MCHWS consisting of layers of digital comparison switching circuits[47] is shown in Fig. 5 and simulation results will be additionally considered in our report and in section 4. Here we note only the main results. The processing cycle in the pipelined structure of DMIP and SU based on the Altera FPGA chip EP3C16F484 Cyclone III family did not exceed 25 nanoseconds, which makes it possible to achieve an input / output rate of pixels of the processed and processed images at the level of 40MHz. During the processing cycle, DMIP essentially performs (9 * ln9 - estimates for the best algorithms!) Sorting operations and generates all the ranks and their differences, which gives, taking into account the wide variety of output functions, performance estimates of at least 10^9 operations per second.

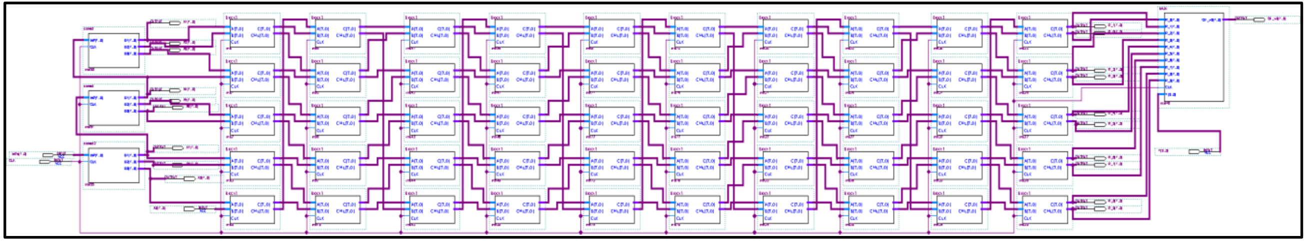


Fig. 5. Structure of DMIP based on FPGA with 10 inputs and 1 output with register memory and conveyor sorting node

Since such processors, described in sections 2.1-2.3, have output signals that are ranked by value and not by difference of values, by some modification[45] they can be used to organize an additional calculation of the difference of signals having neighboring ranks. Besides, the difference in signal values is also necessary for such a function as nonequivalence. Based on the operations of bounded difference and nonequivalence, a whole set of other continuous logic complex operations and functions are constructed. For example, early we can select one of n signals by rank using multiplexer. And now we can also form signals difference between max signal and next by order. So we can find signal that is proportional to difference of any two signals from ordered set. Such approach allows to formed output

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

complement analog signals. If one of reference level is D=1 (255), than difference between the reference and any of signals is the continuous logic complement of the analog signal. A block diagram of such a modified processor for four outputs, a circuit of an internal analog switch and a sub-block for calculating difference and additional analog signals are considered in work[45], where it is shown that such processors can work with both analog and time-pulse-encoded signals. Therefore, we will develop this idea further, taking into account the fact that the selection, amplification, weighting and addition of analog signals, especially currents on the SM, are quite simple.

3. MODELING NONLINEAR IMAGE PROCESSING ALGORITHMS USING A MULTIFUNCTIONAL PROCESSOR BASED ON THE SORTING NODE AND METHOD OF PROCESSING WEIGHTING-SELECTING SIGNALS OF RANK DIFFERENCES

3.1 Modeling the process of wave sorting of signals with Mathcad

Let us consider the aspects and results of modeling the process of wave sorting of signals of vectors Va (n, m) corresponding to the intensities of the pixels of the current windows WAn, m of image A. Software modules and parts of the listings from which the aspects of transformations and ordering of signal components of vectors corresponding to image windows are obvious are shown in Fig. 6-9. Five pairs of a 10-component vector are pairwise and then with a shift (essentially ring) are compared and rearranged by forming two outputs (maximum and minimum). Then the states of the comparators K are analyzed and state vectors are formed:

$$\mathbf{KT1(Va)} := \mathbf{stack}\left(\mathbf{K}\left(\mathbf{Va}_0,\mathbf{Va}_1\right),\mathbf{K}\left(\mathbf{Va}_0,\mathbf{Va}_1\right),\mathbf{K}\left(\mathbf{Va}_2,\mathbf{Va}_3\right),\mathbf{K}\left(\mathbf{Va}_2,\mathbf{Va}_3\right),\mathbf{K}\left(\mathbf{Va}_4,\mathbf{Va}_5\right),\mathbf{K}\left(\mathbf{Va}_4,\mathbf{Va}_5\right),\mathbf{K}\left(\mathbf{Va}_6,\mathbf{Va}_7\right),\mathbf{K}\left(\mathbf{Va}_6,\mathbf{Va}_7\right),\mathbf{K}\left(\mathbf{Va}_8,\mathbf{Va}_9\right),\mathbf{K}\left(\mathbf{Va}_8,\mathbf{Va}_9\right)\right)$$

$$\mathbf{KT2(Va)} := \mathbf{stack}\left(\mathbf{K}\left(\mathbf{Va}_0,\mathbf{Va}_1\right),\mathbf{K}\left(\mathbf{Va}_0,\mathbf{Va}_1\right),\mathbf{K}\left(\mathbf{Va}_2,\mathbf{Va}_3\right),\mathbf{K}\left(\mathbf{Va}_2,\mathbf{Va}_3\right),\mathbf{K}\left(\mathbf{Va}_4,\mathbf{Va}_5\right),\mathbf{K}\left(\mathbf{Va}_4,\mathbf{Va}_5\right),\mathbf{K}\left(\mathbf{Va}_6,\mathbf{Va}_7\right),\mathbf{K}\left(\mathbf{Va}_6,\mathbf{Va}_7\right),\mathbf{K}\left(\mathbf{Va}_9,\mathbf{Va}_8\right),\mathbf{K}\left(\mathbf{Va}_9,\mathbf{Va}_8\right)\right)$$

Further, based on them, the P_Sort matrix is formed as a set of permutations and VaSort vectors of sorted signals for each window. Depending on the selected rank, all rank output values of the function for the central pixel of the window are formed:

$$\mathbf{rang\_p(n,m,rp)} := \left(\mathbf{VaSort(n,m)}\right)_{\mathbf{rp}}$$

$\mathbf{V\_sor0}_{\mathbf{n,m}} := \mathbf{rang\_p(n,m,}0)$, …. etc. Using reverse permutations, verification is performed to verify the quality of sorting, see Fig. 9, and the reconstructed VaSortV vectors are formed and from them the reconstructed I_ApVO image. As you can see from the results, the sorting algorithm works correctly, and the results of dissecting the image of the “butterfly” obtained for different ranks are shown in Fig. 10. The visibility of the operation of ranking filtering algorithms is demonstrated by the signal diagrams of the conversion of one particular line (part of it) of the image shown in Fig. 11.

3.2 Rank differences signals weighing-selection processing method

The essence of the proposed method is as follows. The signals of the selected current WAG window 3x3 in size, corresponding to the tx, ty-pixel $\text{Amo}_{\text{tx, ty}}$ of some image, are fed to the input of the sorting node and ordered signals V_sor0-V-sor8 are formed at its outputs, which are denoted as Ds (r) where r is the rank. These signals corresponding to the ranks, using the control vector Y are selected (weighed) by the switching node in accordance with the formula:

$$\mathbf{Fs\_Am(Y)} := \sum_{\mathbf{r}=0}^{9} \mathbf{Y_r}\cdot\mathbf{Ds(r)}$$

Similarly, from the calculated (Dr0 = D - V_sor0, Dr1 = V_sor0– V_sor1, Dr2 = V_sor1– V_sor2, ... Dr8 = V_sor7– V_sor8, Dr9 = V_sor8) rank differences, denoted by Dr (r), are selected using the similar control vector Y (weighed), if necessary, the second switching node in accordance with the formula:

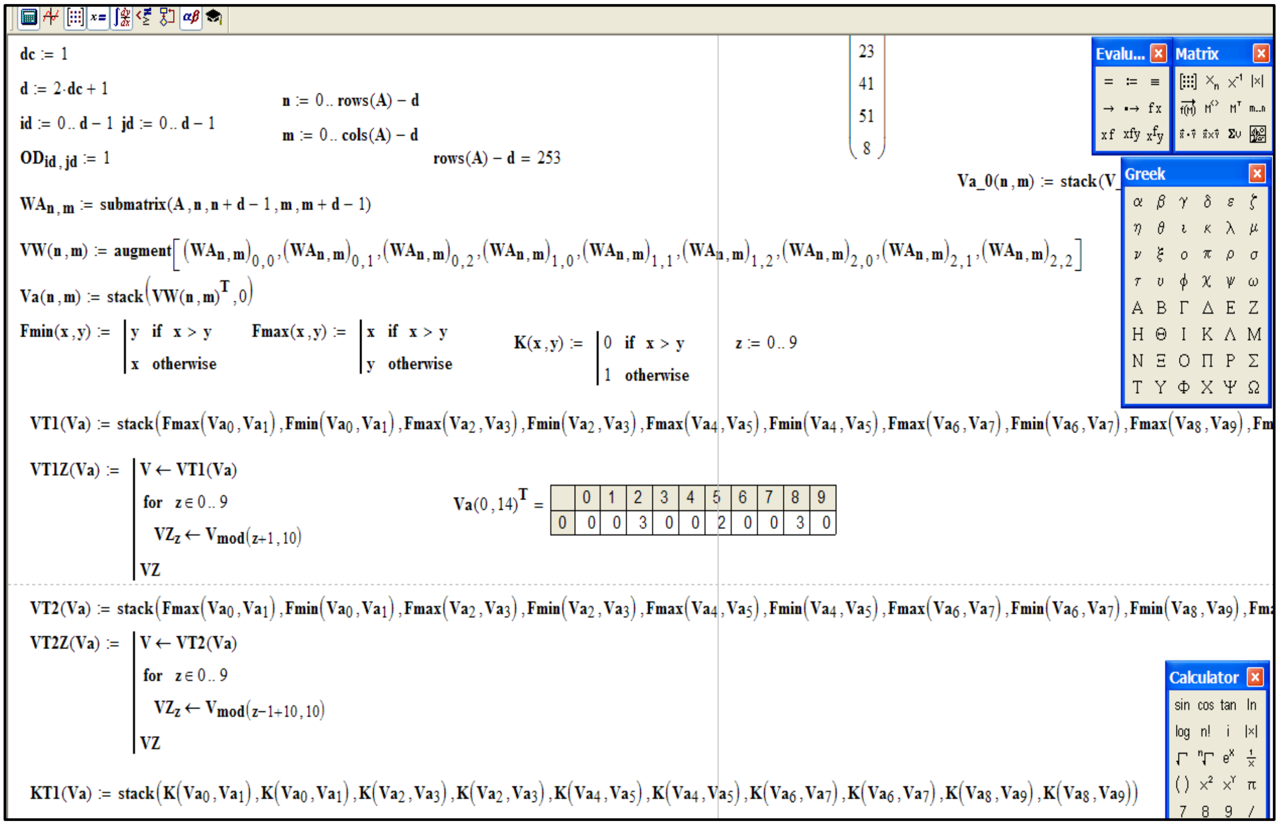
$$\mathbf{F\_Am(Y)} := \sum_{\mathbf{r}=0}^{9} \mathbf{Y_r}\cdot\mathbf{Dr(r)}$$

Examples of control vectors and a specific example of a window with signals to be processed, with the results obtained for it for different vectors with explanatory calculations, are shown in Fig. 12. It can be seen from them that a significant number of functions, operations from window signals, including any selected rank, the difference of the selected ranks, addition to the signal, weighted sums of the selected ranks, etc. can be generated at the processor output. Thus, taking into account especially simple implementations of the operations of summation-subtraction, both for digital and analog signals, the proposed method significantly simplifies the implementation and extends the functionality, set of operations.

3.3 Modeling of weighing-selection processing method of rank differences signals with Mathcad

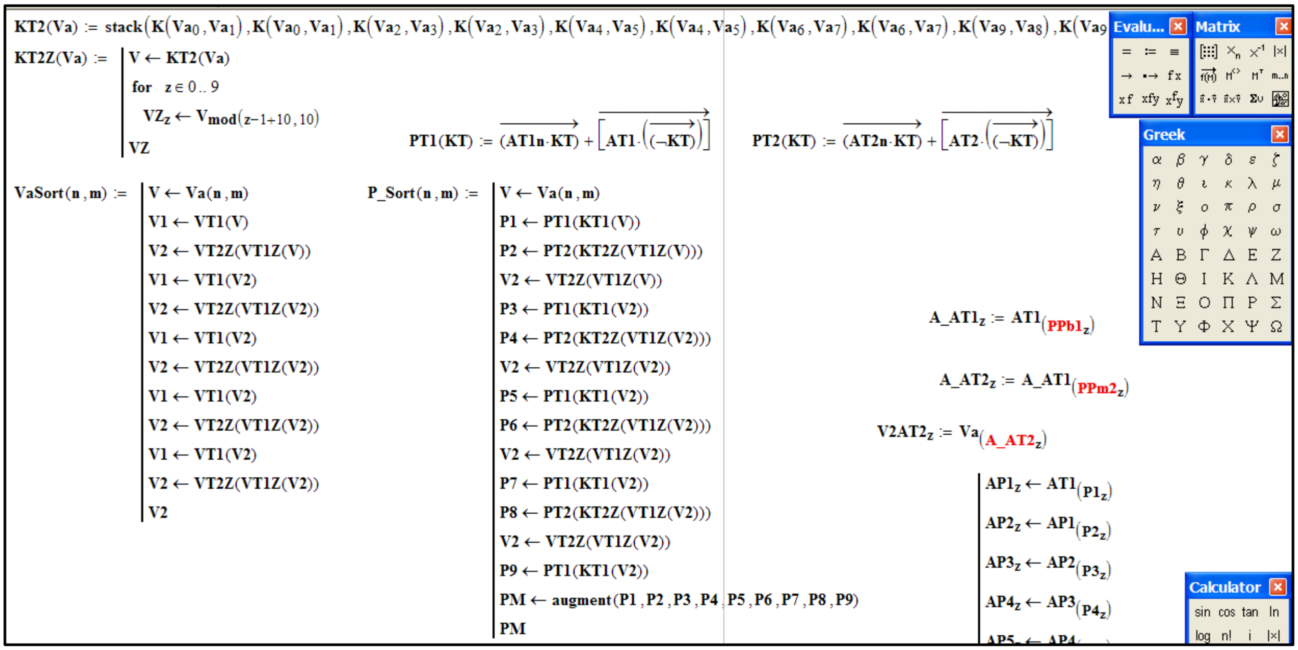
The visibility of the operation of rank filtering algorithms using the rank differences signals weighing-selection processing method is shown by the diagrams of transformation signals of one specific row (its part) of the image shown in Fig. 13-15. Examples of transformations by the proposed method and algorithms for color images and its spectral components are shown in Fig. 16-18. The types of control vectors Y are partially shown in Fig. 12, 19.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use



```
dc := 1
d := 2·dc + 1                                   n := 0.. rows(A) − d
id := 0.. d − 1  jd := 0.. d − 1                m := 0.. cols(A) − d
OD_id,jd := 1                                                   rows(A) − d = 253

23
41
51
8

Va_0(n,m) := stack(V

WA_n,m := submatrix(A, n, n + d − 1, m, m + d − 1)

VW(n,m) := augment[(WA_n,m)_0,0, (WA_n,m)_0,1, (WA_n,m)_0,2, (WA_n,m)_1,0, (WA_n,m)_1,1, (WA_n,m)_1,2, (WA_n,m)_2,0, (WA_n,m)_2,1, (WA_n,m)_2,2]

Va(n,m) := stack(VW(n,m)^T, 0)

Fmin(x,y) := | y  if  x > y     Fmax(x,y) := | x  if  x > y     K(x,y) := | 0  if  x > y     z := 0.. 9
             | x  otherwise                  | y  otherwise                | 1  otherwise

VT1(Va) := stack(Fmax(Va0, Va1), Fmin(Va0, Va1), Fmax(Va2, Va3), Fmin(Va2, Va3), Fmax(Va4, Va5), Fmin(Va4, Va5), Fmax(Va6, Va7), Fmin(Va6, Va7), Fmax(Va8, Va9), Fm

VT1Z(Va) := | V ← VT1(Va)
            | for  z ∈ 0.. 9
            |   VZ_z ← V_mod(z+1, 10)
            | VZ

                    |   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
Va(0,14)^T =        | 0 | 0 | 0 | 3 | 0 | 0 | 2 | 0 | 0 | 3 | 0 |

VT2(Va) := stack(Fmax(Va0, Va1), Fmin(Va0, Va1), Fmax(Va2, Va3), Fmin(Va2, Va3), Fmax(Va4, Va5), Fmin(Va4, Va5), Fmax(Va6, Va7), Fmin(Va6, Va7), Fmin(Va8, Va9), Fm

VT2Z(Va) := | V ← VT2(Va)
            | for  z ∈ 0.. 9
            |   VZ_z ← V_mod(z−1+10, 10)
            | VZ

KT1(Va) := stack(K(Va0, Va1), K(Va0, Va1), K(Va2, Va3), K(Va2, Va3), K(Va4, Va5), K(Va4, Va5), K(Va6, Va7), K(Va6, Va7), K(Va8, Va9), K(Va8, Va9))
```

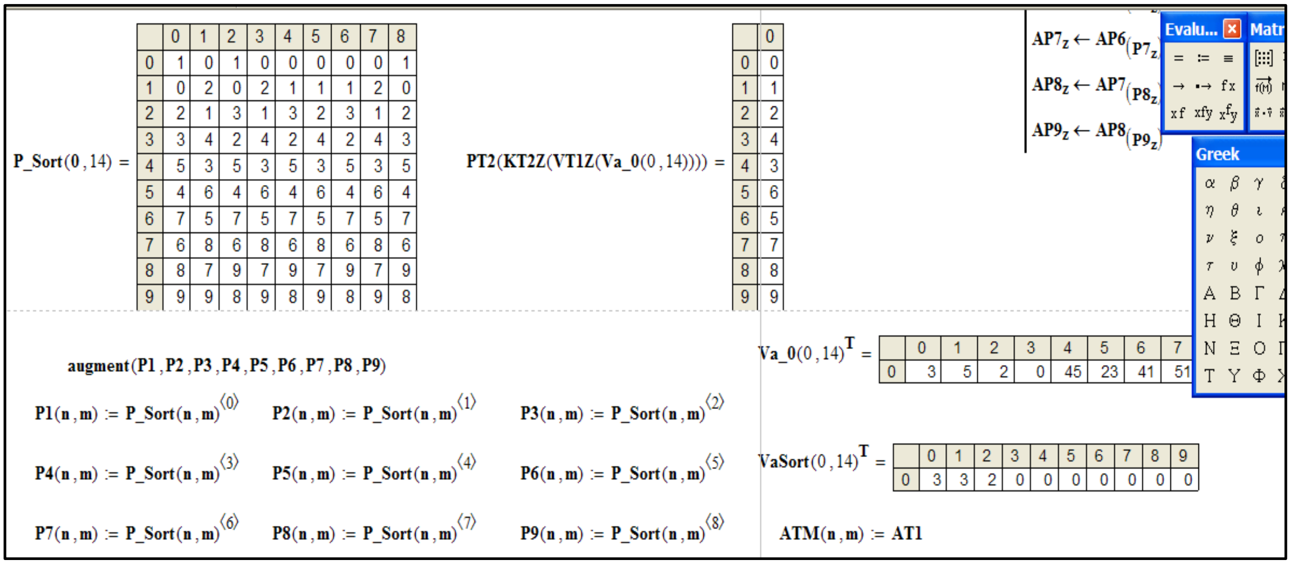
Fig. 6. Listing (part 1 - multi-step push-pull ordering process) for modeling the process of wave sorting of signals of vectors Va (n, m) corresponding to the pixel intensities of the current windows WAn, m of image A



```
KT2(Va) := stack(K(Va0, Va1), K(Va0, Va1), K(Va2, Va3), K(Va2, Va3), K(Va4, Va5), K(Va4, Va5), K(Va6, Va7), K(Va6, Va7), K(Va9, Va8), K(Va9

KT2Z(Va) := | V ← KT2(Va)
            | for  z ∈ 0.. 9
            |   VZ_z ← V_mod(z−1+10, 10)
            | VZ

PT1(KT) := →(AT1n·KT) + →[AT1·(→(¬KT))]        PT2(KT) := →(AT2n·KT) + →[AT2·(→(¬KT))]

VaSort(n,m) := | V ← Va(n,m)                 P_Sort(n,m) := | V ← Va(n,m)
               | V1 ← VT1(V)                                | P1 ← PT1(KT1(V))
               | V2 ← VT2Z(VT1Z(V))                         | P2 ← PT2(KT2Z(VT1Z(V)))
               | V1 ← VT1(V2)                               | V2 ← VT2Z(VT1Z(V))
               | V2 ← VT2Z(VT1Z(V2))                        | P3 ← PT1(KT1(V2))
               | V1 ← VT1(V2)                               | P4 ← PT2(KT2Z(VT1Z(V2)))
               | V2 ← VT2Z(VT1Z(V2))                        | V2 ← VT2Z(VT1Z(V2))
               | V1 ← VT1(V2)                               | P5 ← PT1(KT1(V2))
               | V2 ← VT2Z(VT1Z(V2))                        | P6 ← PT2(KT2Z(VT1Z(V2)))
               | V1 ← VT1(V2)                               | V2 ← VT2Z(VT1Z(V2))
               | V2 ← VT2Z(VT1Z(V2))                        | P7 ← PT1(KT1(V2))
               | V2                                         | P8 ← PT2(KT2Z(VT1Z(V2)))
                                                            | V2 ← VT2Z(VT1Z(V2))
                                                            | P9 ← PT1(KT1(V2))
                                                            | PM ← augment(P1, P2, P3, P4, P5, P6, P7, P8, P9)
                                                            | PM

A_AT1_z := AT1_(PPb1_z)

A_AT2_z := A_AT1_(PPm2_z)

V2AT2_z := Va_(A_AT2_z)

| AP1_z ← AT1_(P1_z)
| AP2_z ← AP1_(P2_z)
| AP3_z ← AP2_(P3_z)
| AP4_z ← AP3_(P4_z)
| AP5_z ← AP4_(
```

Fig. 7. Listing (part 2, multi-step processes of forming VaSort vectors with sorted values and corresponding permutations) for modeling the process of wave sorting of signals of vectors Va (n, m) corresponding to the pixel intensities of the current windows WAn, m of image A

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use



P_Sort(0,14) =

	0	1	2	3	4	5	6	7	8
0	1	0	1	0	0	0	0	0	1
1	0	2	0	2	1	1	1	2	0
2	2	1	3	1	3	2	3	1	2
3	3	4	2	4	2	4	2	4	3
4	5	3	5	3	5	3	5	3	5
5	4	6	4	6	4	6	4	6	4
6	7	5	7	5	7	5	7	5	7
7	6	8	6	8	6	8	6	8	6
8	8	7	9	7	9	7	9	7	9
9	9	9	8	9	8	9	8	9	8

PT2(KT2Z(VT1Z(Va_0(0,14)))) =

	0
0	0
1	1
2	2
3	4
4	3
5	6
6	5
7	7
8	8
9	9

```
AP7_z ← AP6_(P7_z)
AP8_z ← AP7_(P8_z)
AP9_z ← AP8_(P9_z)
```

Va_0(0,14)^T =

	0	1	2	3	4	5	6	7
0	3	5	2	0	45	23	41	51

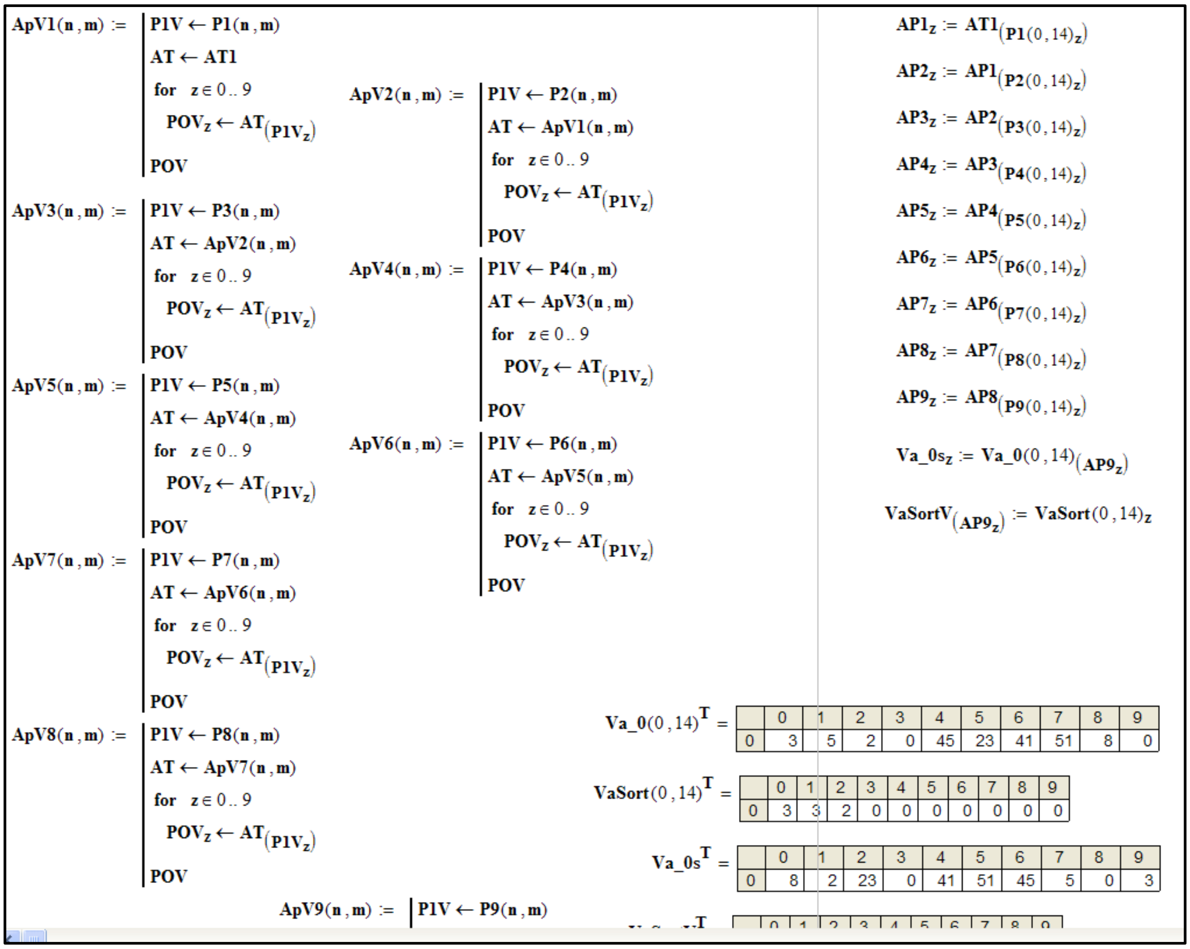
```
augment(P1,P2,P3,P4,P5,P6,P7,P8,P9)

P1(n,m) := P_Sort(n,m)^⟨0⟩    P2(n,m) := P_Sort(n,m)^⟨1⟩    P3(n,m) := P_Sort(n,m)^⟨2⟩

P4(n,m) := P_Sort(n,m)^⟨3⟩    P5(n,m) := P_Sort(n,m)^⟨4⟩    P6(n,m) := P_Sort(n,m)^⟨5⟩

P7(n,m) := P_Sort(n,m)^⟨6⟩    P8(n,m) := P_Sort(n,m)^⟨7⟩    P9(n,m) := P_Sort(n,m)^⟨8⟩
```

VaSort(0,14)^T =

	0	1	2	3	4	5	6	7	8	9
0	3	3	2	0	0	0	0	0	0	0

```
ATM(n,m) := AT1
```

Fig. 8. Listing (part 3 — formation of permutations based on binary vectors of KT1, KT2 states of comparators) for modeling the process of wave sorting of signals of vectors Va (n, m) corresponding to pixel intensities of current windows WAn, m of image A



```
ApV1(n,m) := | P1V ← P1(n,m)
             | AT ← AT1
             | for  z ∈ 0..9
             |    POV_z ← AT_(P1V_z)
             | POV

ApV2(n,m) := | P1V ← P2(n,m)
             | AT ← ApV1(n,m)
             | for  z ∈ 0..9
             |    POV_z ← AT_(P1V_z)
             | POV

ApV3(n,m) := | P1V ← P3(n,m)
             | AT ← ApV2(n,m)
             | for  z ∈ 0..9
             |    POV_z ← AT_(P1V_z)
             | POV

ApV4(n,m) := | P1V ← P4(n,m)
             | AT ← ApV3(n,m)
             | for  z ∈ 0..9
             |    POV_z ← AT_(P1V_z)
             | POV

ApV5(n,m) := | P1V ← P5(n,m)
             | AT ← ApV4(n,m)
             | for  z ∈ 0..9
             |    POV_z ← AT_(P1V_z)
             | POV

ApV6(n,m) := | P1V ← P6(n,m)
             | AT ← ApV5(n,m)
             | for  z ∈ 0..9
             |    POV_z ← AT_(P1V_z)
             | POV

ApV7(n,m) := | P1V ← P7(n,m)
             | AT ← ApV6(n,m)
             | for  z ∈ 0..9
             |    POV_z ← AT_(P1V_z)
             | POV

ApV8(n,m) := | P1V ← P8(n,m)
             | AT ← ApV7(n,m)
             | for  z ∈ 0..9
             |    POV_z ← AT_(P1V_z)
             | POV

ApV9(n,m) := | P1V ← P9(n,m)
```

```
AP1_z := AT1_(P1(0,14)_z)
AP2_z := AP1_(P2(0,14)_z)
AP3_z := AP2_(P3(0,14)_z)
AP4_z := AP3_(P4(0,14)_z)
AP5_z := AP4_(P5(0,14)_z)
AP6_z := AP5_(P6(0,14)_z)
AP7_z := AP6_(P7(0,14)_z)
AP8_z := AP7_(P8(0,14)_z)
AP9_z := AP8_(P9(0,14)_z)

Va_0s_z := Va_0(0,14)_(AP9_z)

VaSortV_(AP9_z) := VaSort(0,14)_z
```

Va_0(0,14)^T =

	0	1	2	3	4	5	6	7	8	9
0	3	5	2	0	45	23	41	51	8	0

VaSort(0,14)^T =

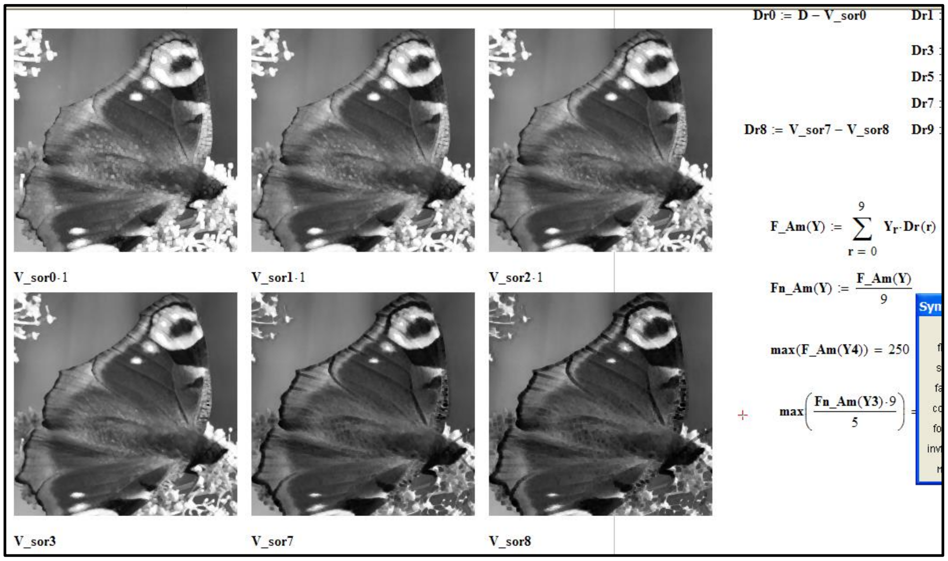
	0	1	2	3	4	5	6	7	8	9
0	3	3	2	0	0	0	0	0	0	0

Va_0s^T =

	0	1	2	3	4	5	6	7	8	9
0	8	2	23	0	41	51	45	5	0	3

Fig. 9. Listing (part 4 - verification using reverse permutations and restoration of the original vectors for windows) for modeling the process of wave sorting of signals of vectors Va (n, m) corresponding to the pixel intensities of the current windows WAn, m

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use



$\mathbf{Dr0} := \mathbf{D} - \mathbf{V\_sor0}$ Dr1

Dr3

Dr5

Dr7

$\mathbf{Dr8} := \mathbf{V\_sor7} - \mathbf{V\_sor8}$ Dr9

$$\mathbf{F\_Am(Y)} := \sum_{r=0}^{9} Y_r \cdot \mathbf{Dr}(r)$$

$$\mathbf{Fn\_Am(Y)} := \frac{\mathbf{F\_Am(Y)}}{9}$$

$$\max(\mathbf{F\_Am(Y4)}) = 250$$

$$\max\left(\frac{\mathbf{Fn\_Am(Y3)}\cdot 9}{5}\right) =$$

Fig. 10. The results of image transformations for different rank values: 0, 1, 2, 3, 7, 8, respectively, and part of the formulas.

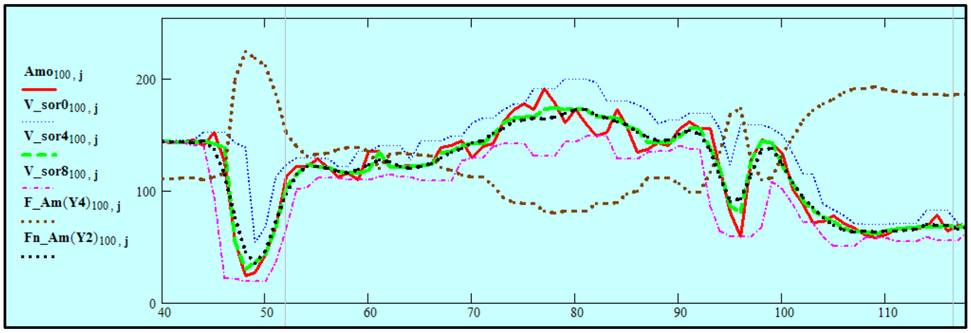
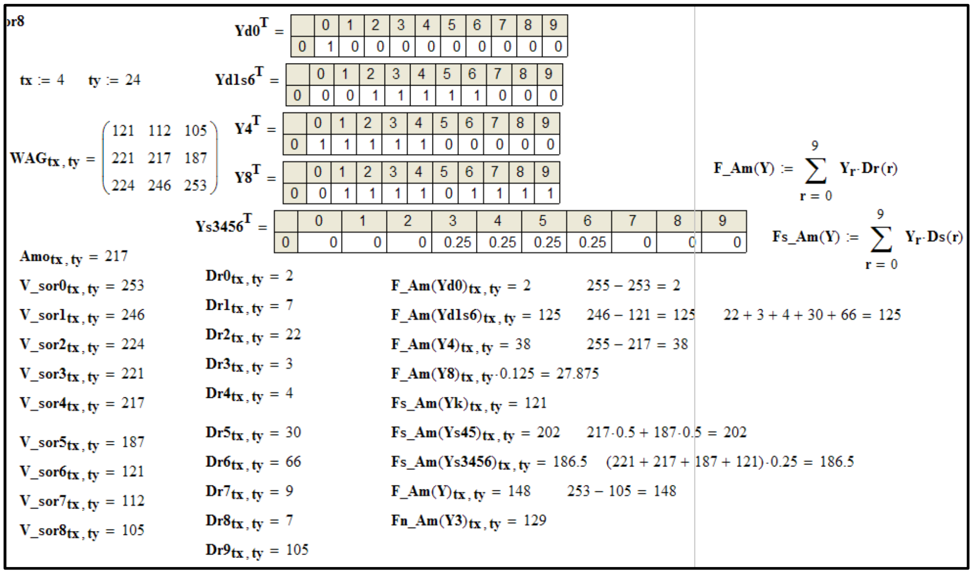


Fig. 11. A good example of processing using an MIP image line (Matcad Window): Original line (red) and received rank and other output functions.



or8

$\mathbf{tx} := 4 \quad \mathbf{ty} := 24$

$$\mathbf{WAG}_{tx,ty} = \begin{pmatrix} 121 & 112 & 105 \\ 221 & 217 & 187 \\ 224 & 246 & 253 \end{pmatrix}$$

$\mathbf{Yd0}^T =$

	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0

$\mathbf{Yd1s6}^T =$

	0	1	2	3	4	5	6	7	8	9
0	0	0	1	1	1	1	1	0	0	0

$\mathbf{Y4}^T =$

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	0	0	0	0	0

$\mathbf{Y8}^T =$

	0	1	2	3	4	5	6	7	8	9
0	0	1	1	1	1	0	1	1	1	1

$\mathbf{Ys3456}^T =$

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0.25	0.25	0.25	0.25	0	0	0

$$\mathbf{F\_Am(Y)} := \sum_{r=0}^{9} Y_r \cdot \mathbf{Dr}(r)$$

$$\mathbf{Fs\_Am(Y)} := \sum_{r=0}^{9} Y_r \cdot \mathbf{Ds}(r)$$

$\mathbf{Amo}_{tx,ty} = 217$

$\mathbf{V\_sor0}_{tx,ty} = 253$

$\mathbf{V\_sor1}_{tx,ty} = 246$

$\mathbf{V\_sor2}_{tx,ty} = 224$

$\mathbf{V\_sor3}_{tx,ty} = 221$

$\mathbf{V\_sor4}_{tx,ty} = 217$

$\mathbf{V\_sor5}_{tx,ty} = 187$

$\mathbf{V\_sor6}_{tx,ty} = 121$

$\mathbf{V\_sor7}_{tx,ty} = 112$

$\mathbf{V\_sor8}_{tx,ty} = 105$

$\mathbf{Dr0}_{tx,ty} = 2$

$\mathbf{Dr1}_{tx,ty} = 7$

$\mathbf{Dr2}_{tx,ty} = 22$

$\mathbf{Dr3}_{tx,ty} = 3$

$\mathbf{Dr4}_{tx,ty} = 4$

$\mathbf{Dr5}_{tx,ty} = 30$

$\mathbf{Dr6}_{tx,ty} = 66$

$\mathbf{Dr7}_{tx,ty} = 9$

$\mathbf{Dr8}_{tx,ty} = 7$

$\mathbf{Dr9}_{tx,ty} = 105$

$\mathbf{F\_Am(Yd0)}_{tx,ty} = 2 \qquad 255 - 253 = 2$

$\mathbf{F\_Am(Yd1s6)}_{tx,ty} = 125 \qquad 246 - 121 = 125 \qquad 22 + 3 + 4 + 30 + 66 = 125$

$\mathbf{F\_Am(Y4)}_{tx,ty} = 38 \qquad 255 - 217 = 38$

$\mathbf{F\_Am(Y8)}_{tx,ty} \cdot 0.125 = 27.875$

$\mathbf{Fs\_Am(Yk)}_{tx,ty} = 121$

$\mathbf{Fs\_Am(Ys45)}_{tx,ty} = 202 \qquad 217 \cdot 0.5 + 187 \cdot 0.5 = 202$

$\mathbf{Fs\_Am(Ys3456)}_{tx,ty} = 186.5 \qquad (221 + 217 + 187 + 121) \cdot 0.25 = 186.5$

$\mathbf{F\_Am(Y)}_{tx,ty} = 148 \qquad 253 - 105 = 148$

$\mathbf{Fn\_Am(Y3)}_{tx,ty} = 129$

Fig. 12. Simulation of DMIP based on FPGA in Mathcad an example of processing a image window (window fragments)

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

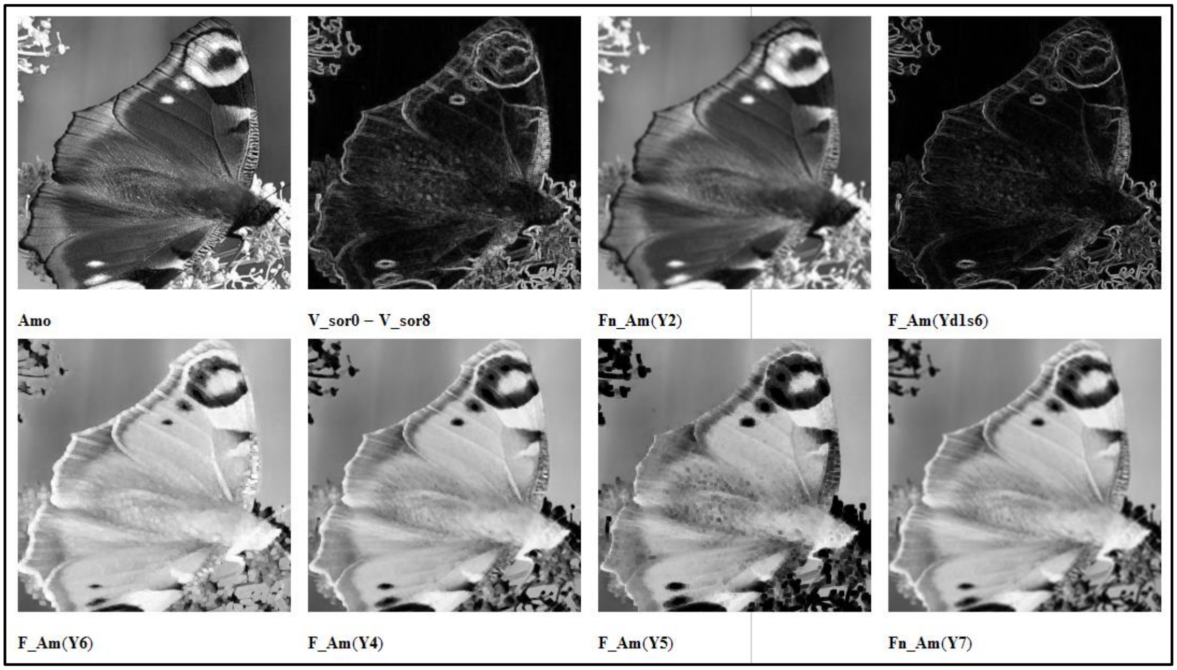


Fig. 13. The results of the Amo image transformations for different rank values and different functions defined by the control vector Y (the types of the vectors is partially shown in Fig. 12 and Fig. 19)

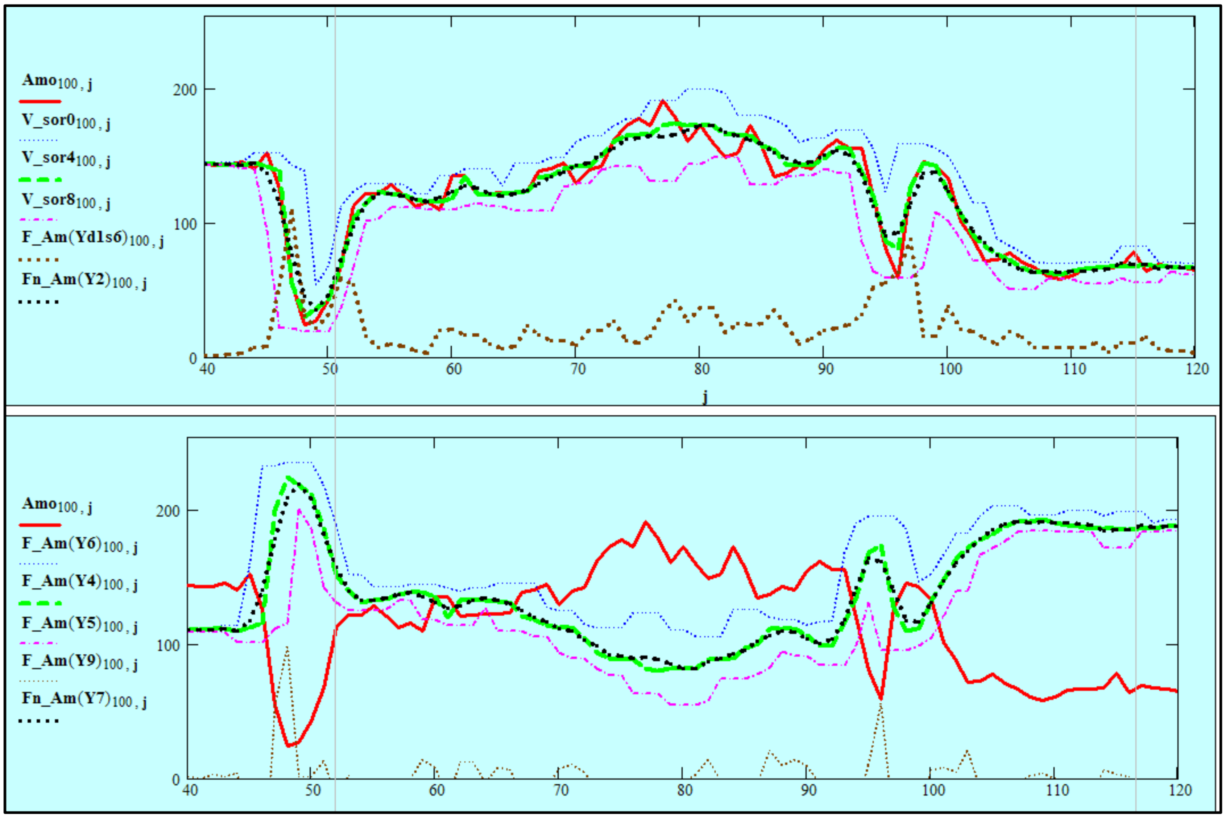


Fig. 14. A good example of nonlinear image processing algorithms using a processor based on the sorting node image line for functions and image in Fig. 13 (Matcad Window): Original line (red) and received rank and other output functions.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

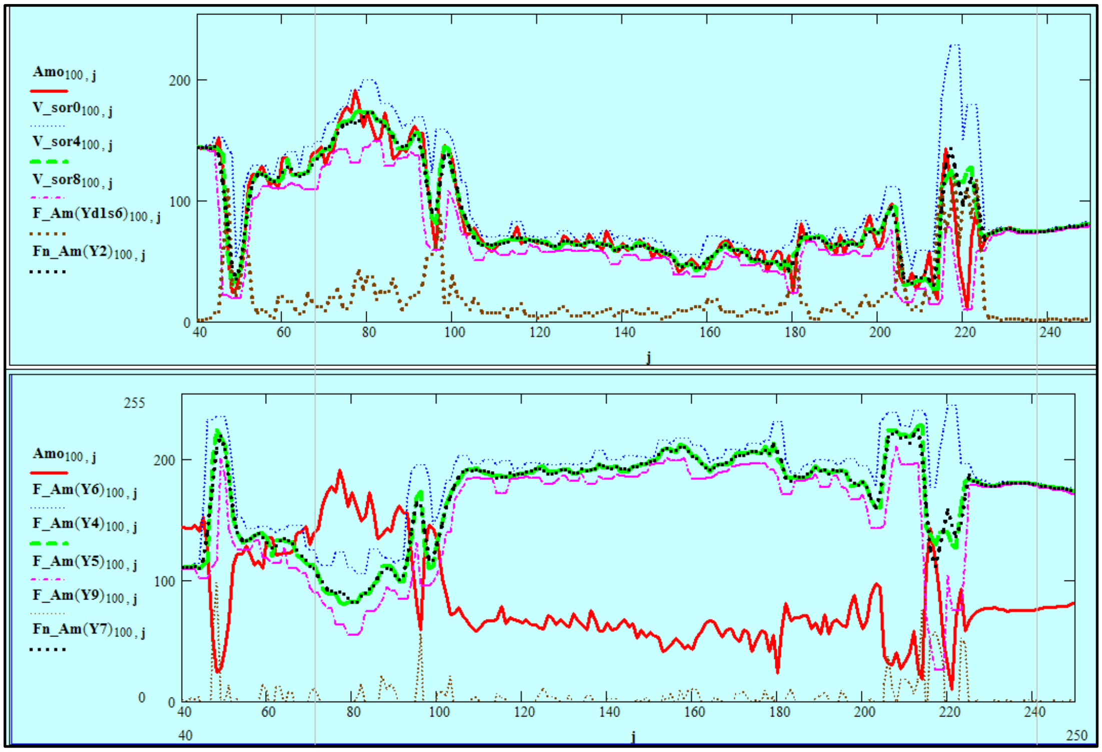


Fig. 15. A good example of nonlinear image processing algorithms using a processor based on the sorting node image line for functions and image in Fig. 13 (Matcad Window): Original line (red) and received rank and other output functions.

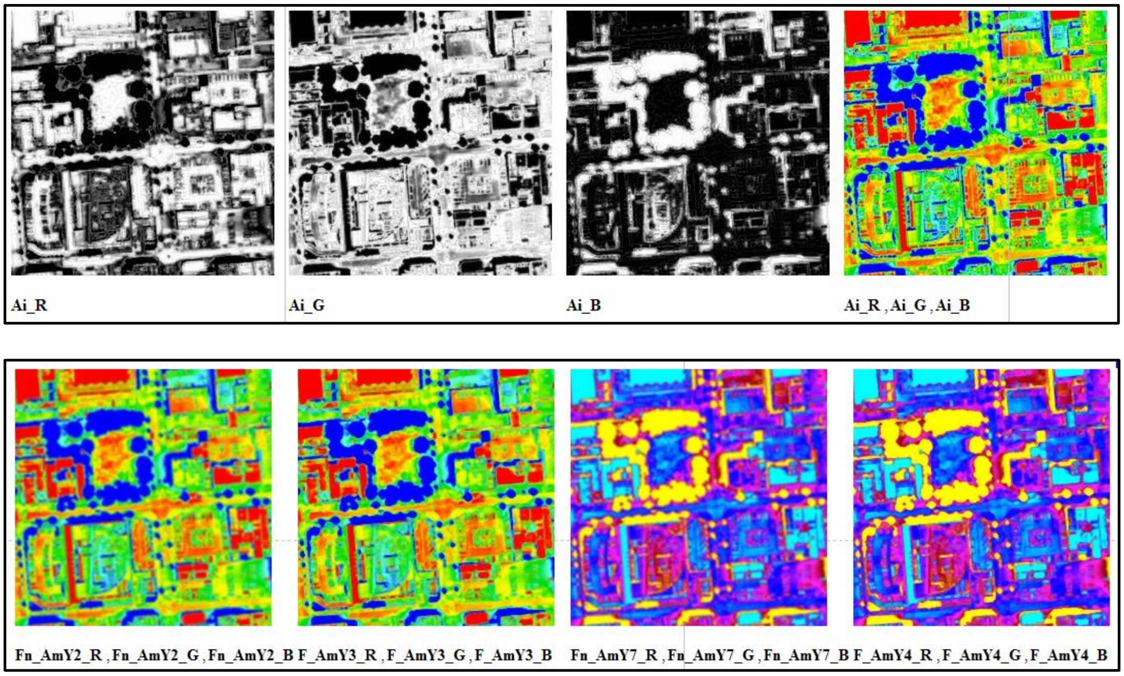

Ai_R
Ai_G
Ai_B
Ai_R , Ai_G , Ai_B
Fn_AmY2_R , Fn_AmY2_G , Fn_AmY2_B
F_AmY3_R , F_AmY3_G , F_AmY3_B
Fn_AmY7_R , Fn_AmY7_G , Fn_AmY7_B
F_AmY4_R , F_AmY4_G , F_AmY4_B

Fig. 16. The results of the transformations of the color image for different ranks and functions defined by the control vectors Y2, Y3, Y7, Y4 (for the last two there is a complementarity of components), at the top is the original image (its components), and at the bottom are converted.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

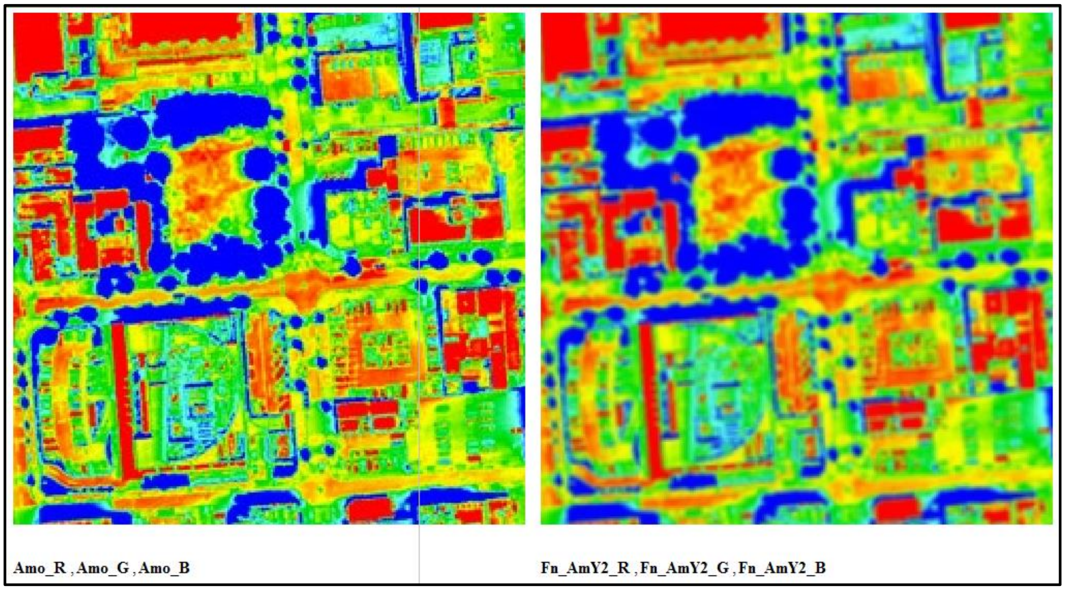


Fig. 17. The results of the color image transformations for the control vector Y2 (the type of vectors is partially shown in Fig. 12, 19), on the left is the original image and on the right is the transformed image (different scale).

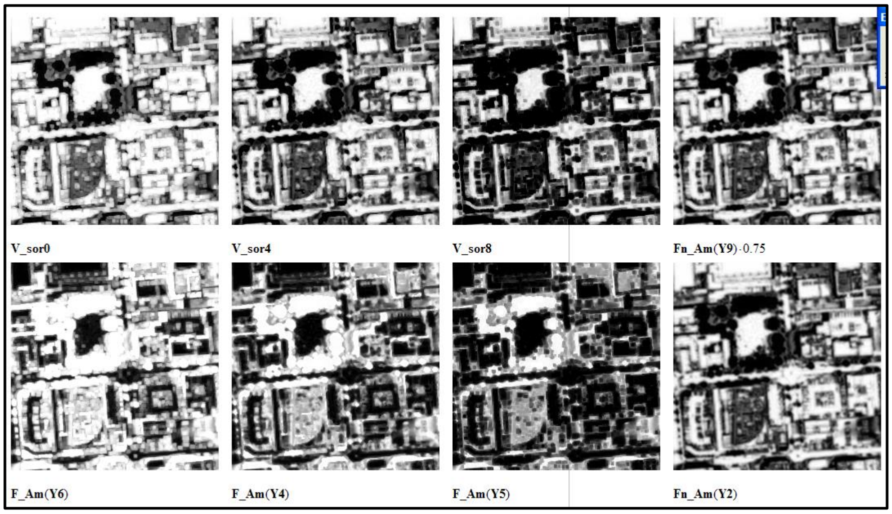


Fig. 18. The results of transformations R of the color image component for the control vectors Y2, Y4, Y5, Y6 and Y9

4. DESIGN AND MODELING OF DIGITAL MULTIFUNCTIONAL IMAGE PROCESSORS

Structure of DMIP_1 based on FPGA with 10 parallel inputs and 1 output; SU based on MCHWS consisting of layers of digital comparison switching circuits is shown in Fig.20 and simulation results are shown in Fig. 21. Here, at first, a variant with one output and supply of all input signals in parallel is shown. For the convenience of data input, we have developed and modeled a processor DMIP_2 circuit with register memory for fast sequential image input and automatic sequential search of processed windows. The FPGA structure of DMIP_2 with serial input and registers memory to form a vector of signals to be sorted and 1 output (only one from ranks) is shown in Fig. 5. And the results of its modeling are shown in Fig. 22, 23. The FPGA Structure of DMIP_3 with serial input and registers memory to form a vector of signals to be sorted and 2 outputs is shown in Fig. 24. The simulation results of DMIP_3 with 2 outputs for rank and rank differences signals weighing-selection processing are shown in Fig. 25-27. As can be seen from Fig. 22, 25 the resources of the Altera FPGA chip EP3C16F484 Cyclone III family are not fully used in the first case, and in the second for the processor with register memory and two outputs almost completely (there is a small margin). The processing cycle in the pipelined structure of DMIP_2 (3) and SU did not exceed 25 nanoseconds, which makes it possible to achieve an input / output rate of pixels of the processed and processed images at the level of 40MHz. During the processing cycle, DMIP_3 essentially performs (9 * ln9-estimates for the best algorithms!) Sorting operations and generates all the ranks and their differences, which gives, taking into account the wide variety of output functions, performance estimates of at least 10^9 operations per second. Only three Altera FPGA chips EP3C16F484 Cyclone III are needed to process color images at the same speed.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

$$Y2 := \begin{pmatrix}0\\1\\2\\3\\4\\5\\6\\7\\8\\9\end{pmatrix} \quad Y3 := \begin{pmatrix}0\\1\\2\\3\\4\\5\\5\\5\\5\\5\end{pmatrix} \quad Y4 := \begin{pmatrix}1\\1\\1\\1\\1\\0\\0\\0\\0\\0\end{pmatrix} \quad Y5 := \begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix} \quad Y6 := \begin{pmatrix}1\\1\\1\\1\\1\\1\\1\\1\\1\\0\end{pmatrix} \quad Y7 := \begin{pmatrix}9\\8\\7\\6\\5\\4\\3\\2\\1\\0\end{pmatrix} \quad Y8 := \begin{pmatrix}0\\1\\1\\1\\1\\0\\1\\1\\1\\1\end{pmatrix} \quad Y9 := \begin{pmatrix}0\\1\\1\\1\\1\\-1\\-1\\-1\\-1\\0\end{pmatrix}$$

$$Yd0 := \begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix} \quad Yd2s7 := \begin{pmatrix}0\\0\\1\\1\\1\\1\\1\\0\\0\\0\end{pmatrix} \quad Yk := \begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\0\\0\\0\end{pmatrix} \quad Ys45 := \begin{pmatrix}0\\0\\0\\0\\0.5\\0.5\\0\\0\\0\\0\end{pmatrix} \quad Ys3456 := \begin{pmatrix}0\\0\\0\\0.25\\0.25\\0.25\\0.25\\0\\0\\0\end{pmatrix}$$

Fig. 19. Simulation of DMIP based on FPGA (the types of control vectors Y)

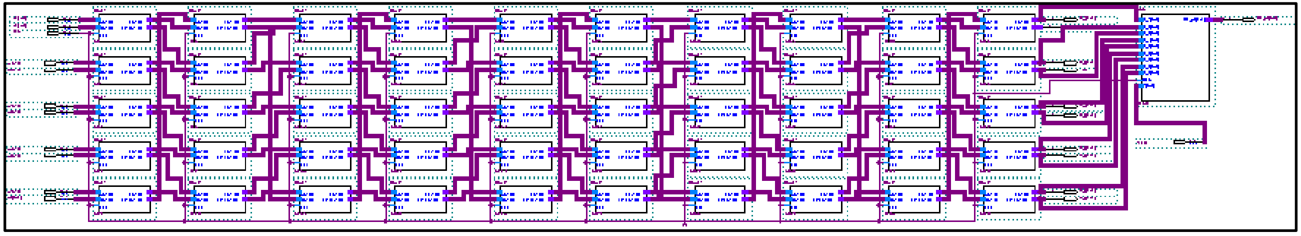
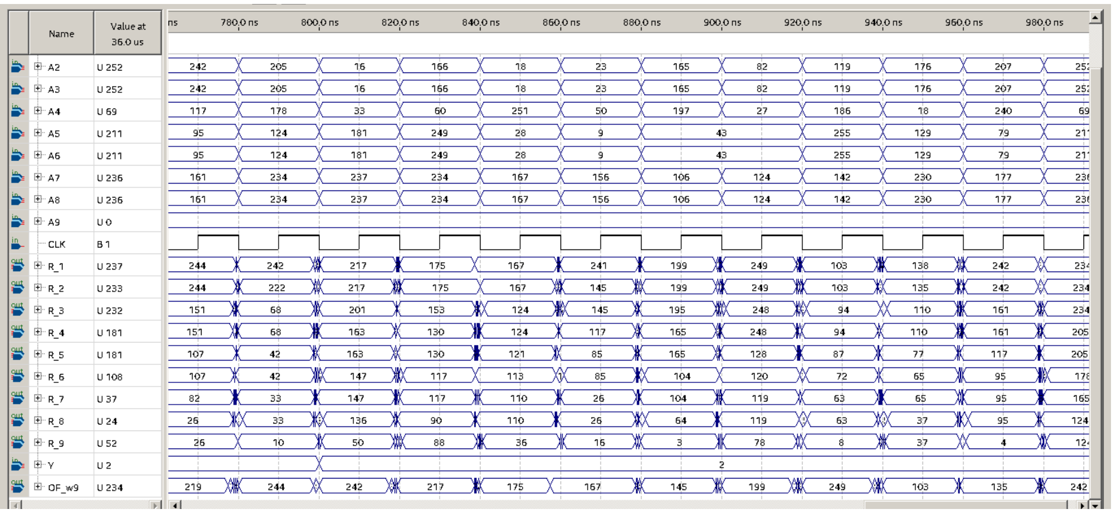


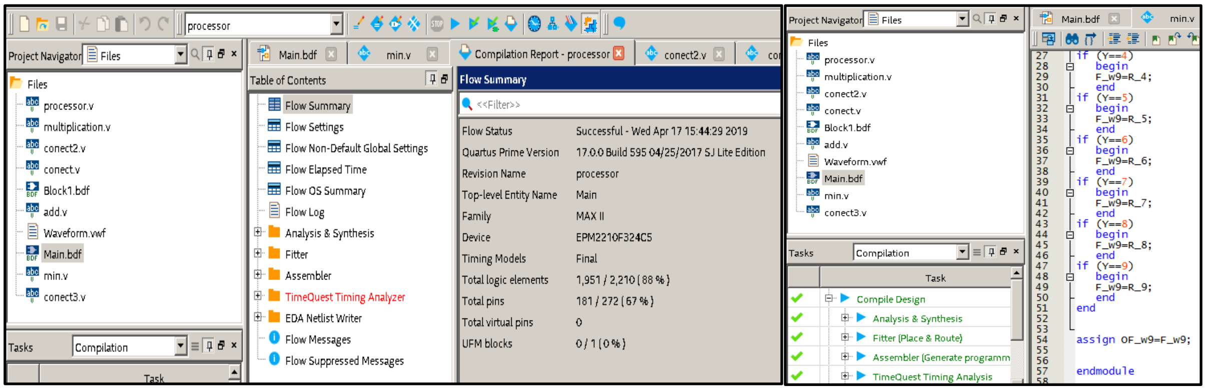
Fig. 20. Structure of DMIP_1 based on FPGA with 10 parallel inputs and 1 output; SU based on MCHWS consisting of layers of digital comparison switching circuits



Name	Value at 36.0 us
A2	U 252
A3	U 252
A4	U 69
A5	U 211
A6	U 211
A7	U 236
A8	U 236
A9	U 0
CLK	B 1
R_1	U 237
R_2	U 233
R_3	U 232
R_4	U 181
R_5	U 181
R_6	U 108
R_7	U 37
R_8	U 24
R_9	U 52
Y	U 2
OF_w9	U 234

ns 780,0 ns 800,0 ns 820,0 ns 840,0 ns 860,0 ns 880,0 ns 900,0 ns 920,0 ns 940,0 ns 960,0 ns 980,0 ns
A2: 242 205 16 166 18 23 165 82 119 176 207 252
A3: 242 205 16 166 18 23 165 82 119 176 207 252
A4: 117 178 33 60 251 50 197 27 186 18 240 69
A5: 95 124 181 249 28 9 43 255 129 79 211
A6: 95 124 181 249 28 9 43 255 129 79 211
A7: 161 234 237 234 167 156 106 124 142 230 177 236
A8: 161 234 237 234 167 156 106 124 142 230 177 236
R_1: 244 242 217 175 167 241 199 249 103 138 242 234
R_2: 244 222 217 175 167 145 199 249 103 135 242 234
R_3: 151 68 201 153 124 145 195 248 94 110 161 234
R_4: 151 68 163 130 124 117 165 248 94 110 161 205
R_5: 107 42 163 130 121 85 165 128 87 77 117 205
R_6: 107 42 147 117 113 85 104 120 72 65 95 178
R_7: 82 33 147 117 110 26 104 119 63 65 95 165
R_8: 26 33 136 90 110 26 64 119 63 37 95 124
R_9: 26 10 50 88 36 16 3 78 8 37 4 124
Y: 2
OF_w9: 219 244 242 217 175 167 145 199 249 103 135 242

Fig. 21. Simulation results of sorting node of structure of DMIP_1 based on FPGA with 10 parallel inputs and 1 output (output switching)



Flow Summary

Flow Status	Successful - Wed Apr 17 15:44:29 2019
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	processor
Top-level Entity Name	Main
Family	MAX II
Device	EPM2210F324C5
Timing Models	Final
Total logic elements	1,951 / 2,210 (88 %)
Total pins	181 / 272 (67 %)
Total virtual pins	0
UFM blocks	0 / 1 (0 %)

```
if (Y==4)
    begin
    F_w9=R_4;
    end
if (Y==5)
    begin
    F_w9=R_5;
    end
if (Y==6)
    begin
    F_w9=R_6;
    end
if (Y==7)
    begin
    F_w9=R_7;
    end
if (Y==8)
    begin
    F_w9=R_8;
    end
if (Y==9)
    begin
    F_w9=R_9;
    end
end


assign OF_w9=F_w9;


endmodule
```

Fig. 22. Simulation of FPGA Structure of DMIP_2 with serial input and registers memory to form a vector of signals to be sorted and 1 output (window fragments)

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

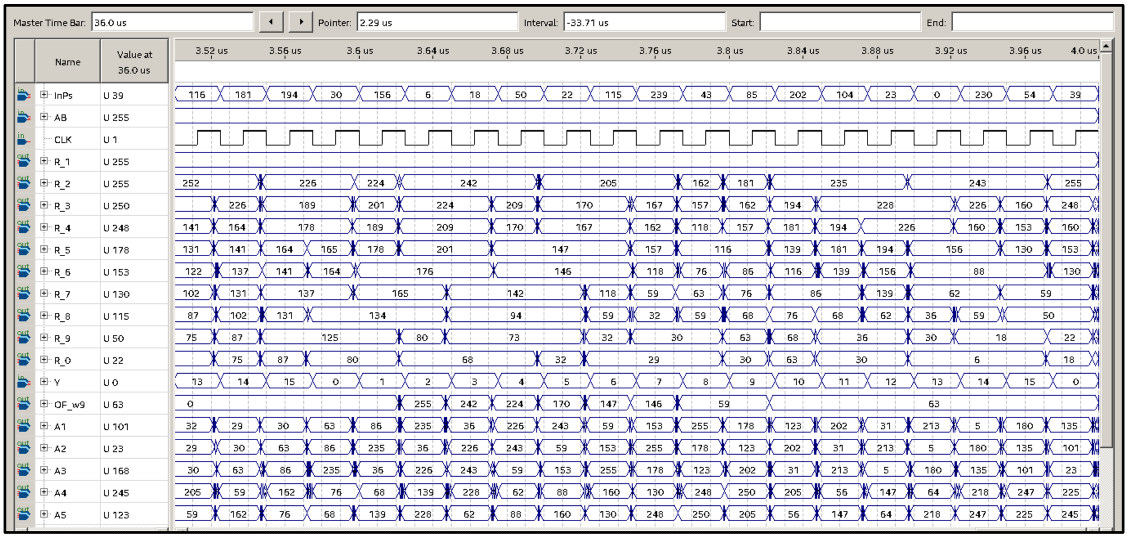


Fig. 23. Simulation results of sorting node of structure of DMIP_2 based on FPGA with serial input and registers memory and 1 output (issuing ranks, one switch)

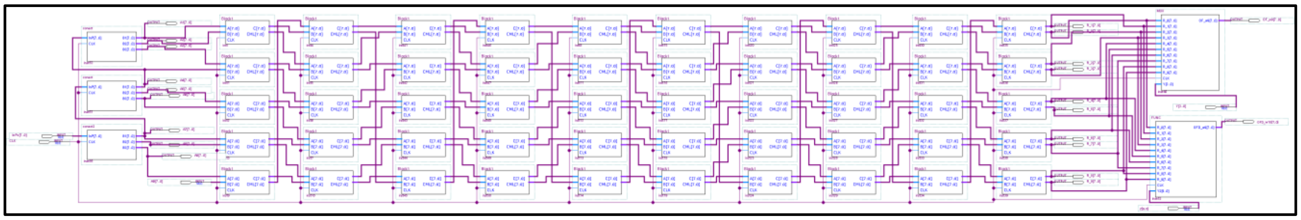
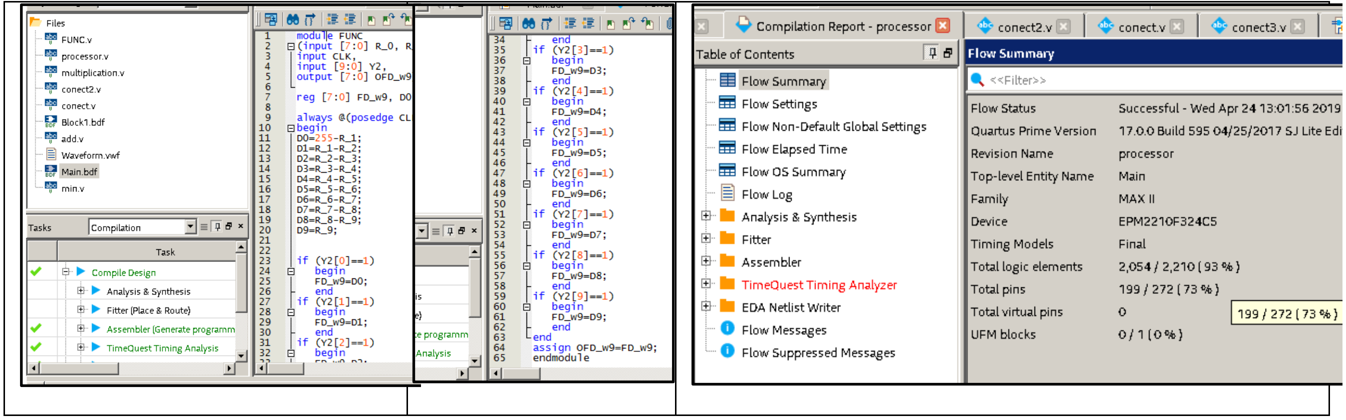


Fig. 24. Structure of DMIP_3 based on FPGA with serial input and registers memory to form a vector of signals to be sorted, 2 outputs for rank and rank differences signals weighing-selection processing



```
module FUNC
(input [7:0] R_0, R
input CLK,
input [9:0] Y2,
output [7:0] OFD_w9

reg [7:0] FD_w9, D0

always @(posedge CL
begin
D0=255-R_1;
D1=R_1-R_2;
D2=R_2-R_3;
D3=R_3-R_4;
D4=R_4-R_5;
D5=R_5-R_6;
D6=R_6-R_7;
D7=R_7-R_8;
D8=R_8-R_9;
D9=R_9;


if (Y2[0]==1)
   begin
   FD_w9=D0;
   end
if (Y2[1]==1)
   begin
   FD_w9=D1;
   end
if (Y2[2]==1)
   begin
```

```
    end
if (Y2[3]==1)
    begin
    FD_w9=D3;
    end
if (Y2[4]==1)
    begin
    FD_w9=D4;
    end
if (Y2[5]==1)
    begin
    FD_w9=D5;
    end
if (Y2[6]==1)
    begin
    FD_w9=D6;
    end
if (Y2[7]==1)
    begin
    FD_w9=D7;
    end
if (Y2[8]==1)
    begin
    FD_w9=D8;
    end
if (Y2[9]==1)
    begin
    FD_w9=D9;
    end
end
assign OFD_w9=FD_w9;
endmodule
```

Flow Summary

Flow Status	Successful - Wed Apr 24 13:01:56 2019
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edi
Revision Name	processor
Top-level Entity Name	Main
Family	MAX II
Device	EPM2210F324C5
Timing Models	Final
Total logic elements	2,054 / 2,210 (93 %)
Total pins	199 / 272 (73 %)
Total virtual pins	0
UFM blocks	0 / 1 (0 %)

199 / 272 (73 %)

Fig. 25. Simulation of structure of DMIP_3 based on FPGA with serial input and registers memory to form a vector of signals to be sorted, 2 outputs for rank and rank differences signals weighing-selection processing (window fragments, unit design listings)

CONCLUSIONS

A new iterative process for sorting signal arrays, which differs from the known signals sorting structures by uniformity, versatility, which allows direct and inverse sorting of an array of analog or digital signals was proposed. The basic elements of the proposed sorting unit (SU) are simple relational nodes. Such elements can be implemented on a different element basis, including, on devices for selecting a maximum or minimum of two analog or digital signals.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

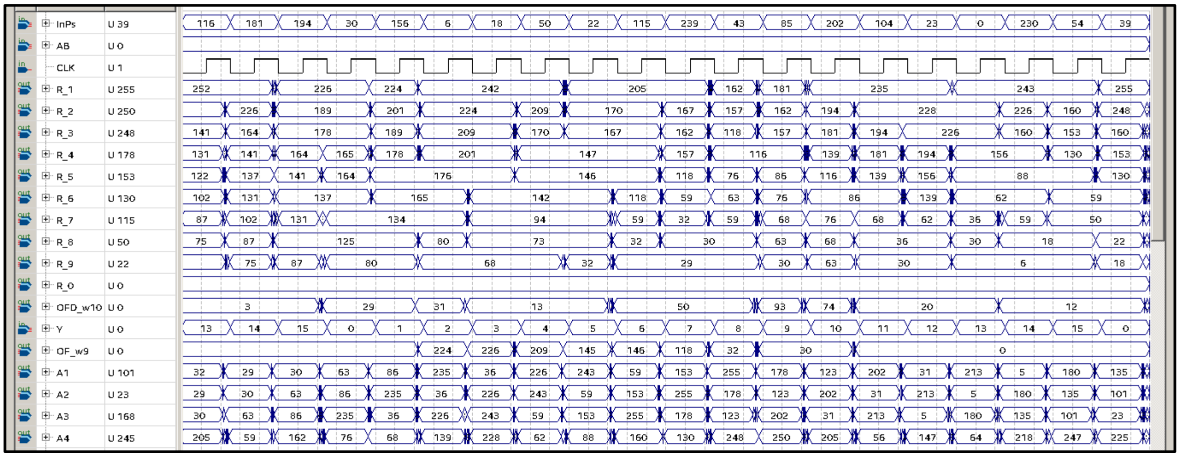


Fig. 26. Simulation of DMIP_3 with serial input and 2 outputs (issuing ranks by the first and weighted rank differences by the second multiplexers) in the case of the formation of additions to the largest at the second output.

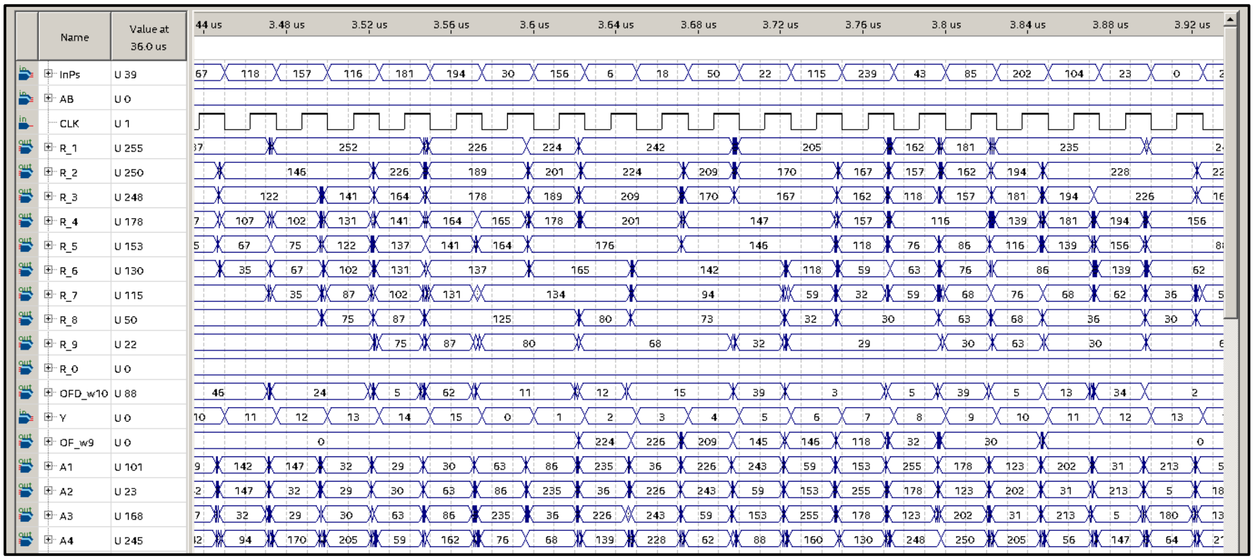


Fig. 27. Simulation results of sorting node of structure of DMIP_3 with 2 outputs (issuing ranks, two switches) in case of formation of a difference of ranks r2-r3

Such SU for creating picture type image processors (IP) have a number of advantages: high speed and reliability, simplicity, small power consumption, high integration level. The inclusion of an iterative node for sorting signals into a IP structure makes it possible to significantly simplify its design and increase the functional capabilities of such IP. The simulation results confirm the proposed approaches to the design of SUs of analog and digital signals, which simplify the complexity of the nodes by an order of magnitude, ensuring their uniformity, regularity and simplicity of scaling. The power consumption of the IPs does not exceed 2mW, the response and processing times are 10μs and can be less by an order of magnitude, the supply voltage is 1.8÷3.3V, and the operating currents are optimally in the range of 10÷20μA. The energy efficiency of the proposed preprocessor with the iterative SU is 25x10^9 op / s · W, which corresponds to the best technical solutions. In the work we are shown, that after sorting or comparative analysis of signals by levels of selected window of image, a promising opportunity appears to implement DMIPs with enhanced functionality using the new method of weighting-selecting rank differences of signals. The essence of the method is that by composing the differences of the signals ordered by rank and the upper level of their range, we can simultaneously form several resulting output signals, choosing the necessary difference signals from their set according to the control commands and weighing them additionally before the summation. We are shown that using this approach of processing the current window signals significantly expands the set of operations and functions for filtering images, simplifying hardware implementation of DMIP, especially for analog and mixed technologies. We determined set of basic possible executable instruction-functions by processors based on proposed method, presenting the simulation results. We show the results of

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

design and modeling the proposed new FPGA-implementations of DMIP. Simulation results show that processing time in such circuits does not exceed 25 nanoseconds. Circuits are simple, have low supply voltage (2.5 V), low power consumption (50mW), digital accuracy. Calculations show that in the case of using Altera FPGA chip EP3C16F484 of Cyclone III family, it is possible to implement DMIP with register memory for image size of 64*64 and window 3*3 in the one chip. For 2.5V power and clock frequency of 200MHz the power consumption will be at the level of 200mW, and the calculation time for pixel of filters will be at the level of 25ns.

REFERENCES

[1] Fey, D., "Architecture and technologies for an optoelectronic VLSI," Optic 112(7), 274-282 (2001).

[2] Lei Yi, Guangbao Shan, Song Liu, Chengmin Xie, High-performance processor design based on 3D on-chip cache, Microprocessors and Microsystems, Volume 47, 2016, Pages 486-490, ISSN 0141-9331.

[3] Krasilenko, V. G., Lazarev, A., Grabovlyak, S., "Design and simulation of a multiport neural network heteroassociative memory for optical pattern recognitions," Proc. of SPIE Vol. 8398, 83980N-1 (2012).

[4] Mori, M., Yatagai, T., "Optical learning neural networks with two dimensional structures," Proc. SPIE 3402, 226-232 (1997).

[5] Krasilenko, V., Nikolskyy, A., Zaitsev A., Voloshin V., "Optical pattern recognition algorithms on neural-logic equivalent models and demonstration of their prospects and possible implementations," Proc. SPIE 4387, 247 –260 (2001).

[6] Krasilenko, V. G., Nikolskyy, A. I., Lazarev, A. A. [Design and Simulation of Time-Pulse Coded Optoelectronic Neural Elements and Devices, Optoelectronic Devices and Properties], InTech (2011). ISBN: 978-953-307-204-3, DOI: 10.5772/16175. Available from: http://www.intechopen.com/books/optoelectronic-devices-andproperties/design-and-simulation-of-time-pulse-coded-optoelectronic-neural-elements-and-devices

[7] Krasilenko, V. G., Nikolskyy, A. I,. Lazarev, A. A., [Design and Modeling of Optoelectronic Photocurrent Reconfigurable (OPR) Multifunctional Logic Devices (MFLD) as the Universal Circuitry Basis for Advanced Parallel High- Performance Processing, Optoelectronics - Advanced Materials and Devices], InTech (2013). ISBN: 978-953-51-0922-8, DOI: 10.5772/54540.

[8] Krasilenko, V. G., Nikolskyy, A. I,. Lazarev, A. A., "Designing and simulation smart multifunctional continuous logic device as a basic cell of advanced high-performance sensor systems with MIMO-structure," in Photonics, Devices, and Systems VI, Proc. SPIE 9450, 94500N (2015).

[9] Guilfoyle, P., McCallum, D., "High-speed low-energy digital optical processors," Opt. Eng. 35(2), 436-442 (1996).

[10] Krasilenko, V.G., Bardachenko, V.F.,Nikolsky, A.I., Lazarev, A.A.,Kolesnytsky, O.K. "Design of optoelectronic scalar-relation vector processors with time-pulse coding," Proc. SPIE 5813, 333-341 (2005).

[11] Krasilenko, V., Ogorodnik, K., Nikolskyy, A., Dubchak, V., "Family of optoelectronic photocurrent reconfigurable universal (or multifunctional) logical elements (OPR ULE) on the basis of continuous logic operations (CLO) and current mirrors (CM), " Proc. SPIE, 8001, (2011).

[12] Krasilenko, V. G., Nikolskyy, A. I,. Lazarev, A. A., "Simulation of reconfigurable multifunctional continuous logic devices as advanced components of the next generation high-performance MIMO-systems for the processing and interconnection," Proc. SPIE 9009, 90090R (2014).

[13] Krasilenko, V. G., Nikolskyy, A. I., Lazarev, A. A., "Multichannel serial-parallel analog-to-digital converters based on current mirrors for multi-sensor systems", Proc. SPIE 8550, Optical Systems Design 2012, 855022 (2013).

[14] Volgin, L.I., Mishin, V.A., "Is the future digital or analog?," Information technologies in electric power industry: Cheboksary: RESCNIT, 86-89, (1998).

[15] Volgin, L.I., Mishin, V.A., "Nobody will think for us: Retrospective review and ontological comprehension of artificial intelligence," Sensors and Systems, Vol. 4 (8), 43-48 (1999).

[16] Volgin, L.I., Mishin, V.A., "Ontological aspects of artificial intelligence," Vestnik UlPU, Vol. 4 (8), 13-19 (1999).

[17] Odintsov, B.Y., Dick, V.V., "Syntax of knowledge models of intelligent systems," Instruments and Control Systems, Vol. 1, 15-17 (1998).

[18] Lukasiewicz, J., "Elementy logiki matematycznej," Warsawa, (1929).

[19] Levin, V.I., 'Continuous logic, its generalization and application," Automatica and telemechanica, N8, 3-22, (1990).

[20] Shimbirev, P.N., "Hybrid continuous logic devices," Moscow, Energoatomizdat, 174p., (1990).

[21] Volgin, L.I., "AM-algebra and its application," Ulyanovsk, UlSTU, 52p., (1997).

[22] Volgin, L.I., "Predicate Algebra of Choice and Its Modification (Fundamentals of the Theory and Elemental Basis), "Experience, Results, Problems: Increasing the Competitiveness of Electronic Equipment. Collection of articles, Tallinn, Valgus, Vol. 4, 64-104, (1986).

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

[23] Krasilenko, V. G., Saletsky, F. M., Yatskovsky, V. I., Konate, K., "Continuous logic equivalence models of Hamming neural network architectures with adaptive-correlated weighting," Proc. SPIE 3402, 398 – 408 (1998).

[24] V. LeCun, Y., Bengio, Y., “Convolutional networks for images, speech, and time-series,” In M. A. Arbib, editor, The Handbook of Brain Theory and Neural Networks, MIT Press, 1995.

[25] V. Shafiee, A., et al., "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," 2016 ACM/IEEE 43rd (ISCA), Seoul, 2016, pp. 14-26. doi: 10.1109/ISCA.2016.12.

[26] Krasilenko, V.G., Lazarev A.A., Nikitovich D.V., "Modeling and possible implementation of self-learning equivalence-convolutional neural structures for auto-encoding-decoding and clusterization of images," Proceedings of SPIE Vol. 10453, 104532N (2017).

[27] Krasilenko, V. G., Lazarev, A. A., Nikitovich, D. V., "Design and simulation of optoelectronic neuron equivalentors as hardware accelerators of self-learning equivalent convolutional neural structures (SLECNS)," Proceedings of SPIE Vol. 10689, 106890C (2018).

[28] Sidorov, V.I., Yaroslavsky, L.P., “Additive method for image processing,” Moscow, Nauka, 243p. (1988).

[29] Pozin, N.V., “Simulation of neural structures,” Moscow, Nauka, 264p. (1970).

[30] Krasilenko, V. G., Nikolskyy, A. I., Krasilenko, O. V., Nikolska, M. A., “Continuously logical complementary: dual equivalently analog-to-digital converters for the optical systems,” Proc. SPIE 8001-30, (2011).

[31] Krasilenko, V. G., Nikolskyy, A. I,. Lazarev, A. A., Krasilenko, O. V., Krasilenko, I. A., "Simulation of continuously logical ADC (CL ADC) of photocurrents as a basic cell of image processor and multichannel optical sensor systems", Proceedings of SPIE Vol. 8774, 877414 (2013).

[32] Krasilenko, V. G., Lazarev, A. A., Nikitovich, D. V., "Simulation of continuously logical base cells (CL BC) with advanced functions for analog-to-digital converters and image processors," Proc. SPIE 10438, 104380K (2017)

[33] Musa, P., Sudiro, S.A., Wibowo, E.P., Harmanto, S., Paindavoine, M., “Design and implementation of non-linear image processing functions for CMOS image sensor,” Optoelectronic Imaging and Multimedia Technology II, Proc. of SPIE Vol. 8558 (2012), http://spie.org/Publications/Proceedings/Paper/10.1117/12.2000538.

[34] Andrew J. Tickle Paul K. Harvey Jeremy S. Smith Q. Henry Wu, „Development and simulation of soft morphological operators for a field programmable gate array,“ Journal of Electronic Imaging 22(2), 023034 (Apr–Jun 2013) 023034-1-023034-15 2013 SPIE and IS&T [DOI: 10.1117/1.JEI.22.2.023034].

[35] Tickle, A. J., Harvey, P. K., Smith, J. S., “Applications of a morphological scene change detector (MSCD) for visual leak and failure identification in process and chemical engineering,” Proc. SPIE 7833, 78330W (2010).

[36] Tickle, A. J., “Applications of morphological operators on field programmable gate arrays,” Doctoral Thesis, Univ. of Liverpool, pp. 326–332 (2009).

[37] Jorge D. Mendiola-Santibañez Iván R. Terol-Villalobos, „Image enhancement and segmentation using weighted morphological connected slope filters,“ Journal of Electronic Imaging 22(2), 023022 (Apr–Jun 2013).

[38] D. E. Knuth, The Art of Computer Programming. Reading, MA, USA: Addison-Wesley, Mar. 2011.

[39] Y. Bang and S. Q. Zheng, “A simple and efficient VLSI sorting architecture,” in Proc. 37th Midwest Symp. Circuits Syst., vol. 1. 1994, pp. 70–73.

[40] T. Leighton, Y. Ma, and C. G. Plaxton, “Breaking the_(*n* log2*n*) barrier for sorting with faults,” *J. Comput. Syst. Sci.*, vol. 54, no. 2, pp. 265–304, 1997.

[41] W. Jianping, Y. Yutang, L. Lin, H. Bingquan, and G. Tao, “Highspeed FPGA-based SOPC application for currency sorting system,” in *Proc. 10th Int. Conf. Electron. Meas. Instrum. (ICEMI)*, Aug. 2011, pp. 85–89.

[42] J. L. Bentley and R. Sedgewick, “Fast algorithms for sorting and searching strings,” in Proc. 8th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA), Jan. 1997, pp. 360–369.

[43] V. Kundeti and S. Rajasekaran, “Efficient out-of-core sorting algorithms for the parallel disks model,” J. Parallel Distrib. Comput., vol. 71, no. 11, pp. 1427–1433, 2011.

[44] Krasilenko, V. G., Bardachenko, V. F., Nikolsky, A. I., Lazarev, A. A., „Programmed optoelectronic time-pulse coded relational processor as base element for sorting neural networks,“ Proc. SPIE 6576, 657610 (2007).

[45] Krasilenko, V. G., Nikolskyy, A. I,. Lazarev, A. A., Lazareva, M.V. "Design and simulation of programmable relational optoelectronic time-pulse coded processors as base elements for sorting neural networks", Proceedings of SPIE Vol. 7723, 77231G (2010).

[46] Krasilenko, V.G., Lazarev A.A., Nikitovich D.V., "Design and simulation of image nonlinear processing relational preprocessor based on iterational sorting node," Proc. SPIE 11028, Optical Sensors 2019, 110282X (11 April 2019).

[47] Krasilenko, V.G., Lazarev A.A., Nikitovich D.V., “Rank differences of signals by weighing-selection processing method for implementation of multifunctional image processing processor,” Proc. SPIE 11163, Emerging Technologies IV, Strasbourg, France (2019).

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 18 Nov 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use