# High-speed image processing systems in non-destructive testing

**D V Shashev**[1] **and S V Shidlovskiy**[1,2]

[1] Department of Innovative Technologies, National Research Tomsk State University, 36 Lenin ave., 634050 Tomsk, Russia
[2] Institute of Power Engineering, National Research Tomsk Polytechnic University, 30 Lenin ave., 634050 Tomsk, Russia

E-mail: dshashev@mail.ru

**Abstract**. Digital imaging systems are using in most of both industrial and scientific industries. Such systems effectively solve a wide range of tasks in the field of non-destructive testing. There are problems in digital image processing for decades associated with the speed of the operation of such systems, sufficient to efficiently process and analyze video streams in real time, ideally in mobile small-sized devices. In this paper, we consider the use of parallel-pipeline computing architectures in image processing problems using the example of an algorithm for calculating the area of an object on a binary image. The approach used allows us to achieve high-speed performance in the tasks of digital image processing.

## 1. Introduction
Digital imaging systems are using in most of both industrial and scientific industries. Such systems effectively solve a wide range of tasks in the field of non-destructive testing (NDT) [1, 2]. Due to advances in computer technology, it is possible to visualize various physical fields (acoustic, thermal, X-ray, optical and etc.) and to analyze a big data, which is effectively used in most modern methods in NDT.

There are problems in digital image processing for decades associated with the speed of the operation of such systems, sufficient to efficiently process and analyze video streams in real time, ideally in mobile small-sized devices.

In search of a solution to this problem, new approaches in software and hardware are continuously researched and developed. One of the alternative approaches described in this paper is the development of specialized software and algorithmic support for a new computing architecture of a parallel-pipeline type, called a reconfigurable computing environment (RCE).

## 2. Reconfigurable computing environment
RCE is a discrete mathematical model of a high-performance computing system consisting of identical and equally connected elementary universal elements (elementary computers (EC)), programmatically tuned to perform any function from a complete set of logical functions, memory and any connection with its neighbors [3]. ECs works only in the basis of logical functions AND-OR-NOT. According to the computer model proposed by E.V. Evreinov and V.G. Khoroshevsky [4], on the basis of which the RCE are built, the ECs in RCE are interconnected as shown in Figure 1. Each *ij* EC in RCE is

bidirectionally connected to neighboring ECs. Here, $y_1$, $y_2$, ..., $y_8$ and $f_1^M$, $f_2^M$, ..., $f_8^M$ respectively information inputs and outputs between ECs; $A_{ij}$, $B_{ij}$, ..., $H_{ij}$ - the name of the links between ECs; $x_{ij}$ - the main input, which receives the pixel value of the original image; $f_{ij}$ - the main output from which the value of the corresponding pixel of the resulting image is taken; $z_{ij}$ is the tuning input to which the EC tuning code is input.

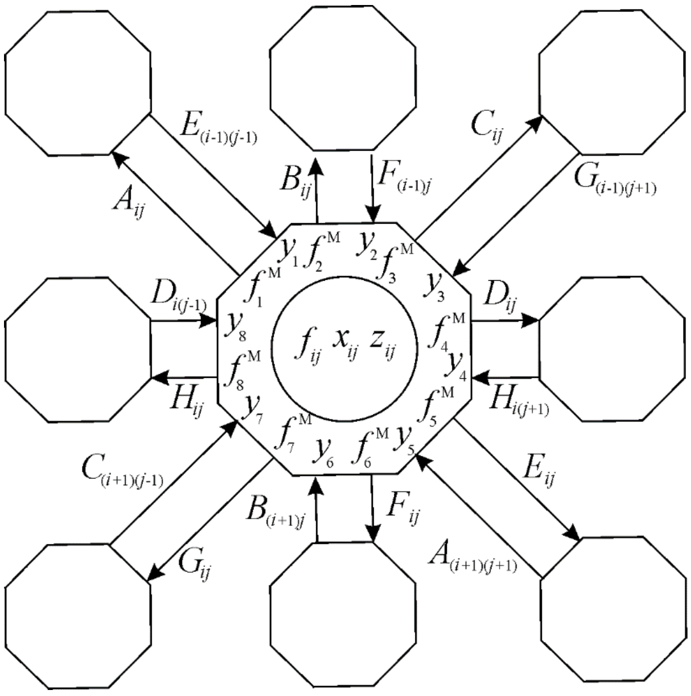


**Figure 1.** The structure of the interconnections of ECs in RCE

The basis for building the RCE are the following principles:

- homogeneity - all ECs are identical and are the same type connected to each other;
- short-range action - all ECs are connected only to the nearest ECs, signal transmission between remote ECs is carried out via intermediate ECs;
- universality - each EC realizes a set of logical functions;
- software configuration - each EC can be configured to perform one function using external setting signals and continue to save the tuning state until the next tuning signal arrives.

Due to this RCE have the following unique properties:

- high speed (due to parallel computing);
- high reliability (due to interchangeability of ECs);
- high manufacturability (due to the uniformity of ECs and connections between them);
- multifunctionality and adaptability (due to the possibility of changing the architecture for a specific task);
- independence to element basis (i.e., independence from the manufacturing technology of computing environment.

## 3. Image processing in reconfigurable computing environments

The basis for image processing in RCE is the concept that each pixel of the original image is processed by the corresponding EC of RCE, so in general the size of the RCE matrix is equal to the image dimension in pixels [5]. However, not all image processing algorithms can be implemented in this way. To do this, we enter the statement that the RCE in general can be multidimensional, while the size of the first layer of the RCE matrix is always equal to the dimension of the original image and its input receives the pixel values of the given image. Thus, the algorithm is executed sequentially, or in parallel (depending on the specific task), from the first RCE layer to the last one.

In this article, as an example, we consider the developed algorithm for calculating the area of an object on a binary image, performed using a simulation model of RCE.

### *3.1. Algorithm for calculating the area of an object on a binary image*

Let us explain the principle of the algorithm for calculating the area of an object (in pixels) on a binary image in a RCE.

We agree that the binary image has passed the necessary filtering, and also that the dimension of the given image is $(m \times m)$ pixels, where *m* is any natural number divisible by 3.

To implement the algorithm, it is necessary to use RCE with *k* layers, where:

$$k = \log_3 m. \tag{1}$$

The dimension of each layer is $(n_i \times n_i)$ cells, with:

$$n_i = \frac{m}{3^i}, \tag{2}$$

where $i = \overline{1,k}$ - the serial number of the RCE layer.

The work principles of the EC of the first RCE layer is described as follows. The original binary image *A* is conditionally divided into fragments $A_i$ with dimension of (3 × 3) pixels. The input of each EC receives information about the value of the pixels of the corresponding fragment $A_i$. Further, EC implements the logical scheme of the full adder of nine one-digit binary numbers, respectively, at the output of the cell we have a four-bit binary number, because in the case that all nine pixels of the fragment have a logical value of 1, then at the output we get the decimal number 9, which in binary code represents a four-digit binary number.

Now the first layer of RCE is conditionally divided into parts with the dimension of (3 × 3) ECs. Each EC of the second RCE layer receives input values of nine four-digit numbers from the corresponding part of the RCE. Further in the EC, a logical scheme of the full adder of nine four-bit binary numbers is realized.

The calculations described above are made from first RCE layer to the last, which consists of one RCE cell, the output of which contains an *n*-bit binary number numerically equal to the area of the object on the binary image.

Let's analyze the principle of the algorithm using the example of a test binary image with dimension of (9 × 9) pixels, shown in Figure 2(a).

In Figure 2(b) the test binary image is divided into fragments $A_i$ with dimension of (3 × 3) pixels. Using the description of finding the area of an object on a binary image, let's say that in this case it will be necessary to use RCE with the number of layers $k = 2$. Accordingly, the first layer of RCE is an matrix of ECs with dimension of (3 × 3) (Figure 2(c)), and the second layer consists of one EC (Figure 2(d)).

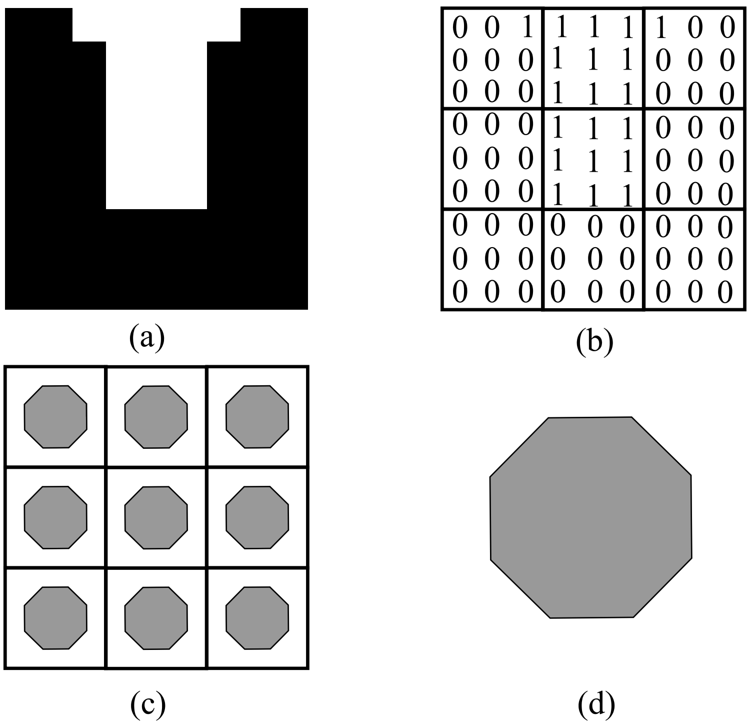
In this case, EC of RCE will realize the logical scheme of the full adder of nine binary four-digit numbers, and the output will be a binary seven-digit number.

Let us describe the sequence of calculating the area of an object on a test binary image. On the first layer of RCE, each EC receives nine single-digit binary numbers. Realizing the logical scheme of the full adder of nine single-digit binary numbers, EC can maximally output a four-digit binary number equal numerically to the decimal number 9. Then, nine binary four-digit numbers enter the input of the EC of the second RCE layer. After realizing the logical scheme of the full adder of nine binary four-digit numbers, the cell can maximally output a seven-digit binary number equal to the numerically decimal number 81. It is easy to calculate that the result of this algorithm execution to the test image is the area of the object equal to 20 pixels.

## *3.2. Results of simulation*

To test the efficiency of the algorithm described above, a simulation model of RCE was developed in MATLAB Simulink, shown in Figure 3.

This simulation model is conditionally divided into structural blocks (SB). SB1 delivers the test image, shown in Figure 2(a). SB2 here is the first RCE layer with dimension of (3 × 3) ECs. SB3 includes a second RCE layer with one EC and displays the value of the counted area in the "Display" block.



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2.** The test binary image (a), the values of the pixels of the test image (b), the abstract representation of the first RCE layer (c) and the abstract representation of the second RCE layer (d)

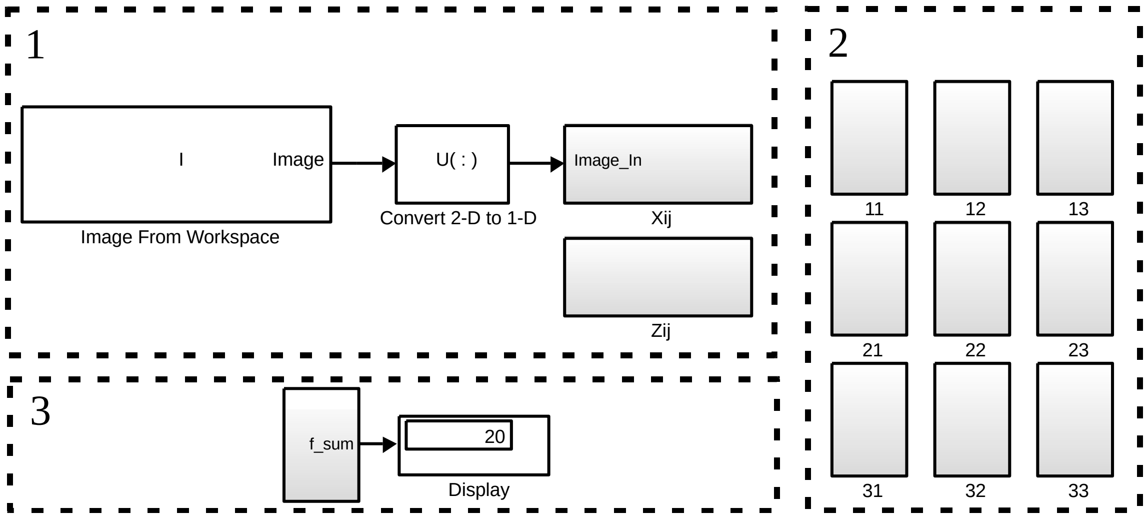


**Figure 3.** Simulation model of RCE for calculating the area of an object on a test binary image

As you can see, the "Display" block shows the correct value of the area of the object on a binary image equal to 20 pixels, which in turn confirms the operability of the algorithm and RCE model.

## 4. Conclusion

Due to the construction principles are achieved unique properties of the computers with the parallel-pipeline architecture, in particular RCE.

The hardware realization of algorithms, the complete adaptation of algorithms for the computational architecture of RCE, the use of natural parallelism make it possible to achieve high performance of a task with the help of RCE. Such an approach in the future will solve the problem of big data processing in real time in the field of digital image processing, including directly on a mobile small-sized device, which in turn will be effectively used in solving problems of NDT.

Using the example of the algorithm for calculating the area of an object on a binary image, the operability of the proposed approach is shown. Simulation modeling shows that, depending on the dimension of the original image, the algorithm is performed in 6-10 EC cycle time.

**Acknowledgments**
The reported study was funded by RFBR according to the research project No. 16-37-00082 mol_a.

## References

[1] Lopez F, Maldague X and Ibarra-Castanedo C 2014 *Opto-Electronics Review* **22(4)** 245-51
[2] Ramírez-Rozo T, Benítez-Restrepo H, García-Álvarez J and Castellanos-Domínguez G 2013 *Proc. Int. Conf. on Image Analysis and Processing – ICIAP 2013 (Naples)* (Berlin: Springer) pp 121-30
[3] Shidlovskiy S 2006 *Avtomaticheskoye upravleniye. Perestraivayemie strukturi* (Tomsk: Tomskiy gosudarstvennyy universitet) (in Russian)
[4] Khoroshevskiy V and Yevreinov E 1978 *Odnorodnie vychislitelnie sistemi* (Novosibirsk: Nauka) (in Russian)
[5] Shashev D and Shidlovskiy S 2015 *Optoelectronics, Instrumentation and Data Processing* **51(3)** 19-26