International Scholarly Research Network
ISRN Machine Vision
Volume 2012, Article ID 152601, 7 pages
doi:10.5402/2012/152601

Research Article

An Artificial Cellular Convolution Architecture for Real-Time Image Processing

H. Mahrous[1] and A. P. James[2]

[1] *Griffith School of Engineering, Griffith University, Nathan Campus, Nathan, QLD 4111, Australia*
[2] *Machine Intelligence Group, School of Computer Science, Indian Institute of Information Technology and Management-Kerala, Technopark Campus, Thiruvananthapuram 695581, India*

Correspondence should be addressed to A. P. James, apj@ieee.org

Received 20 July 2011; Accepted 28 August 2011

Academic Editors: J. Alvarez-Borrego and S. J. Horng

An artificial cell is comprised of the most basic elements in a hierarchical system, that has minimal functionality, but general enough to obey the rules of "artificial life." The ability to replicate, organize hierarchy, and generalize within an environment is some of the properties of an artificial cell. We present a hardware artificial cell having the properties of generalization ability, the ability of self-organization, and the reproducibility. The cells are used in parallel hardware architecture for implementing the real-time 2D image convolution operation. The proposed hardware design is implemented on FPGA and tested on images. We report improved processing speeds and demonstrate its usefulness in an image filtering application.

1. Introduction

Last two decades of study on artificial life revolve around the problem of understanding the organizational principles of life by treating "cells" as the basic blocks of information processing and transmission [1–3]. Contrary to biological cell research studies that measure organization principle of networks of genes to form a living organ, the artificial cell research investigates how networks of so-called artificial genes can produce essential mechanisms of life made out of electrical circuits [4].

Biological cells can be considered as the basic modules of physical life in embryonic systems and are an inspiration to the conceptual creation of artificial cellular systems. Ideally, the conceptual artificial cells can be organized hierarchically having the properties of the near perfect self-organization, self-reproduction (multiplying), and fault tolerance (cicatrization). These properties often result in a network of huge number of cells. Such bioinspired networks of artificial cells can be organized to perform a function or a set of functions. The implementation of artificial cells in hardware [5] remains an open problem.

In this paper, we propose an artificial cell that carries basic genetic features (basic arithmetic and logical operations) having the properties of generalization ability, the ability of self-organization, and reproducibility. The artificial cells are used to create cellular system networks that implement 2D image convolution, which represent a general and complex image processing operation having several practical machine vision applications.

2. Problem and Objective

In developing a robust digital logic system for application such as image processing, the traditional fault reduction and generalisation methodologies seem to be inefficient and expensive [4, 5]. Since biological cells are well known for their fault tolerance, self-organization, and self-producing properties, the design of artificial cells that imitate such properties could provide general hardware representations that are efficient and inexpensive. With this broad aim, we draw inspiration to design and implement a generic image processing operation: the so-called 2D image convolution operations. In a simplistic view, convolution can be

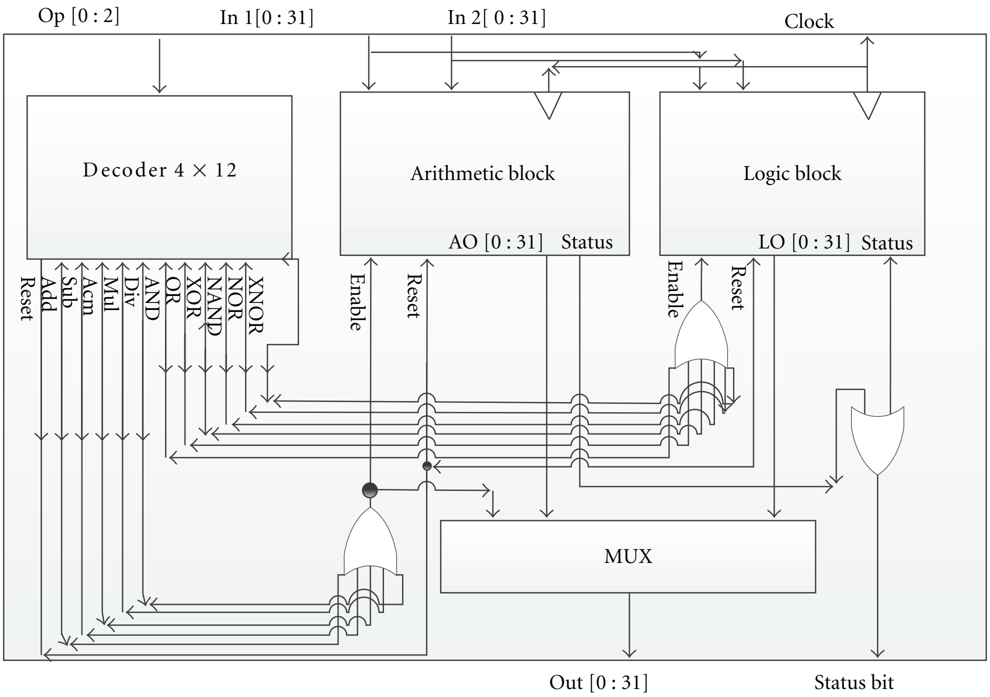


Figure 1: RTL block diagram of the proposed artificial cell.

Table 1: Operation code inputs.

3 bit input code	Basic operations
0000	Rest
0001	Addition
0010	Subtraction
0011	Accumulation
0100	Multiplication
0101	Division
1000	AND
1001	OR
1010	XOR
1011	NAND
1100	NOR
1101	XNOR

considered as a mathematical operation on two functions that produces a modified version of one of the original functions [6]. Convolution has a wide range of practical machine vision applications such as in real-time image processing machines like digital cameras, video cameras, printers, and scanners. Depending on the kernel values of a typical convolution operator, several image processing operations can be implemented. Typically, in an image processing task, the images are read through the imaging sensors and the stored in device memory. The image processing operations are implemented using software agents through a microcontroller or ALU-based framework. Such typical methods to implement convolution result in increased processing requirements with increased image resolution. As distinct from ALU-based framework, we propose a 2D convolution processor that overcomes the limitation of image scaling (increased space complexity) through a design of artificial cells and their networks.

The research methodology of this paper can be summarised in four stages.

(1) *RTL design of the artificial cell.* The cell architecture was designed in VHDL and tested in the typical FPGA framework. The architectural concepts were adapted from the idea of artificial cell and the genotype definition of the cell formed in terms of arithmetic and logical operations.

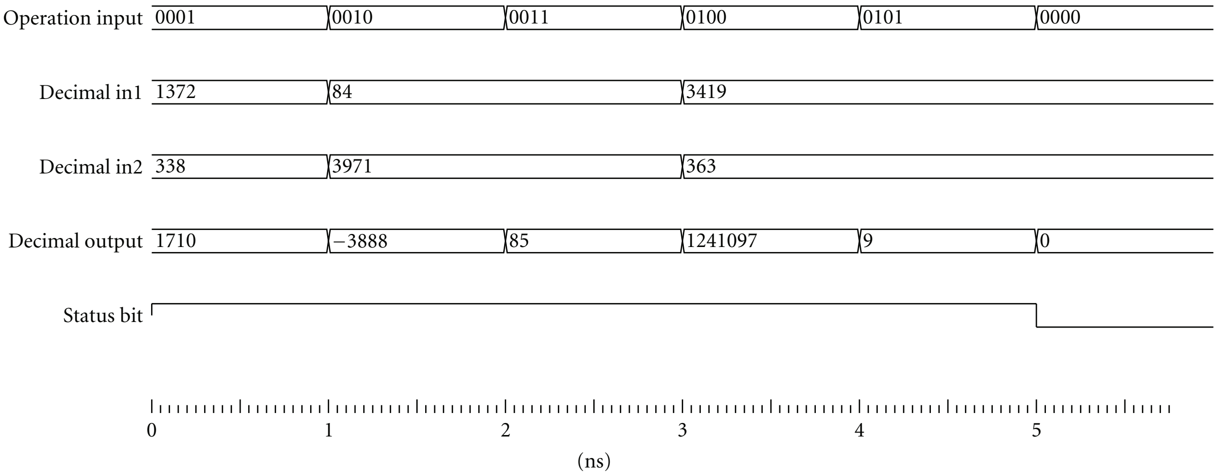
(2) *Verifying the Functionality of the Cell.* Several test cases are designed and the performance of the cell analyzed using a standard FPGA system simulator.

(3) *Construct Topologies of Cells to Achieve Convolution for a Single Cell.* After the functionality of the cell is verified, the architecture is implemented to perform the image convolution operation.

(4) *Performance Comparison.* The performance of the convolution operation implemented with the proposed architecture is compared with an ALU based benchmark algorithmic implementation done in MATLAB.

3. Proposed Artificial Cell Design

In image processing, convolution is an essential kernel-based operator used for (1) intensity averaging to remove signal noise gained from erroneous sensing, (2) edge or gradient



Entity: cell_module Architecture: Date: Mon Jun 06 12 : 19 : 11 AM Pacific Daylight Time 2011 Row: 1 Page: 1

Figure 2: Timing diagrams representing outputs from "arithmetic operation" of the artificial cell.

Table 2: Arithmetic test conditions and cell output.

Time	Operation (binary)	Function	In 1	In 2	Output
0 ns → 1 ns	0001	Addition	1327	338	1710
1 ns → 2 ns	0010	Subtraction	84	3971	−3888
2 ns → 3 ns	0100	Accumulation	84	0	85
3 ns → 4 ns	0100	Multiplication	3419	363	1241097
4 ns → 5 ns	0100	Division	3419	363	9.419
5 ns → 6 ns	0000	Reset	0	0	0

detection, (3) local range detection, and (4) local standard deviation calculations [7, 8]. Although convolution is an analogue mathematical operator by its nature, in the recent past, much of the hardware implementation of convolution operator has been in the form of strict digital circuit logic, with little flexibility of generalization. Performing parallel convolution on the processing level is computationally intensive and expensive; however, it should be simpler and less expensive if performed at the sensing level. A digital implementation of convolution at the sensing level would require less number of data to be processed than the implementation at the processing level. This is because with local sensing and parallel processing of the circuits, the requirement for generalized memory allocations, data conversion and normalization circuits can be reduced. Although artificial-cell-inspired convolution may require a number of circuit elements, it has an advantage of guaranteed fault tolerance. This may result in a complex hardware structure with increasing image resolutions. On the other hand, a parallel convolution would ensure no compromise with speed.

Convolution architecture is constructed from cells that process in parallel to solve mathematical and logical functions. The cell computes arithmetic and logic operations in a synchronous and parallel manner. The architecture is tested on different picture sizes in a pixel-dedicated fashion and a sequential fashion. In the pixel-dedicated fashion, the cell architecture is applied to every pixel at a time to yield a true parallel convolution image processor. In contrast, in the sequential fashion, the cell architecture is applied to individual pixels at an expense of one clock cycle for every pixel. This section will show the cell design, the architectural design, and the test bench design for the sequential setup.

Figure 1 shows the overall Register transfer level of the cell architecture design. The cell takes the following 4 inputs: operation code (genome code), 2–32 bit inputs, and a clock signal for synchronous operations. The operation code is a 3 bit input that specifies the type of operation to be performed by the cell and enables either the arithmetic or the logic block. Table 1 shows the control codes that define the functionality of the cell. The cell for constructing the convolution operation is viewed as a composite of complex arithmetic or logical operations which is broken down to its simplest form as shown in Table 1.

The cell is based on the arithmetic and the logical blocks that perform the defined arithmetic and logical genome. The arithmetic block performs the simplest arithmetic operations like addition, subtraction, accumulations, multiplication, and division. The status bit is enabled high when the cell processes and completes the requested operation, resulting in a 32 bit cell output. The status bit is essential for communicating with neighboring cells if needed, for higher hierarchical level network implementations.

The cell is tested with different decimal values to verify the varieties of different outputs like signed, unsigned, and

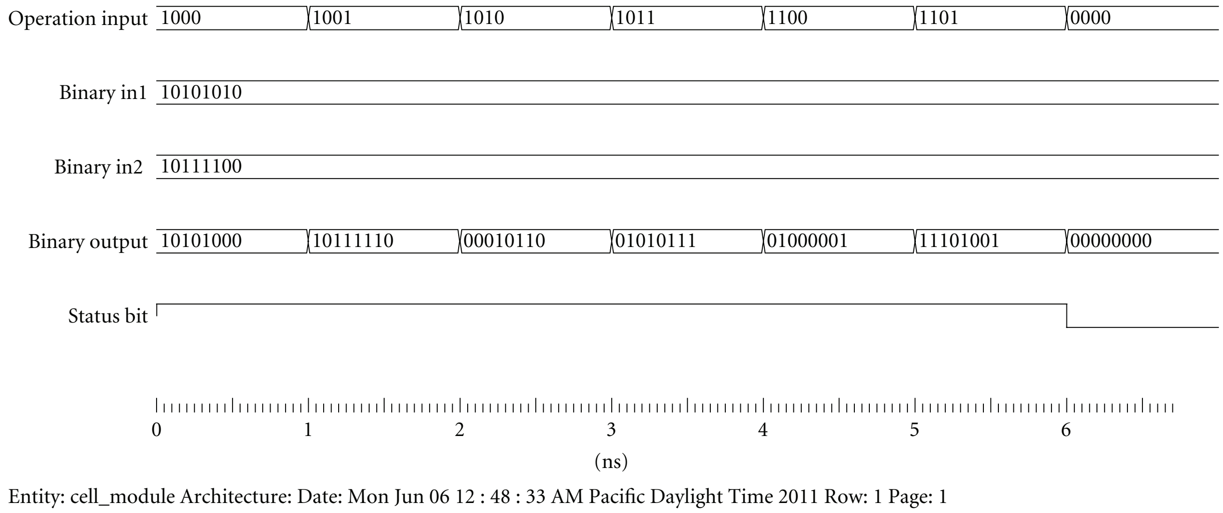


FIGURE 3: Timing diagrams representing outputs from "logic operation" of the artificial cell.

TABLE 3: Logic test conditions and cell output.

Time	Operation (binary)	Function	In 1	In 2	Output
0 ns → 1 ns	1000	AND	10101010	10111100	10101000
1 ns → 2 ns	1001	OR			10111110
2 ns → 3 ns	1010	XOR			00010110
3 ns → 4 ns	1011	NAND			01010111
4 ns → 5 ns	1100	NOR			01000001
5 ns → 6.5 ns	1101	XNOR			11101001
6.5 ns → 7 ns	0000	Reset			00000000

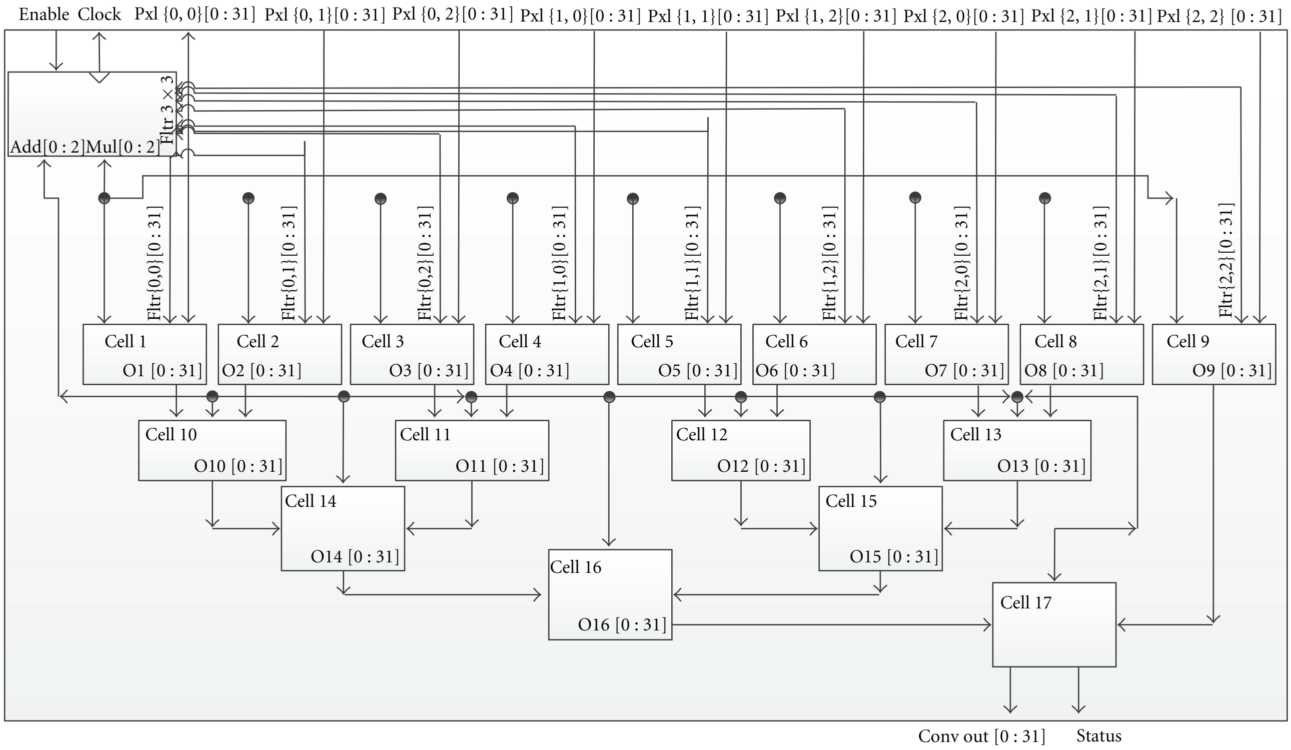
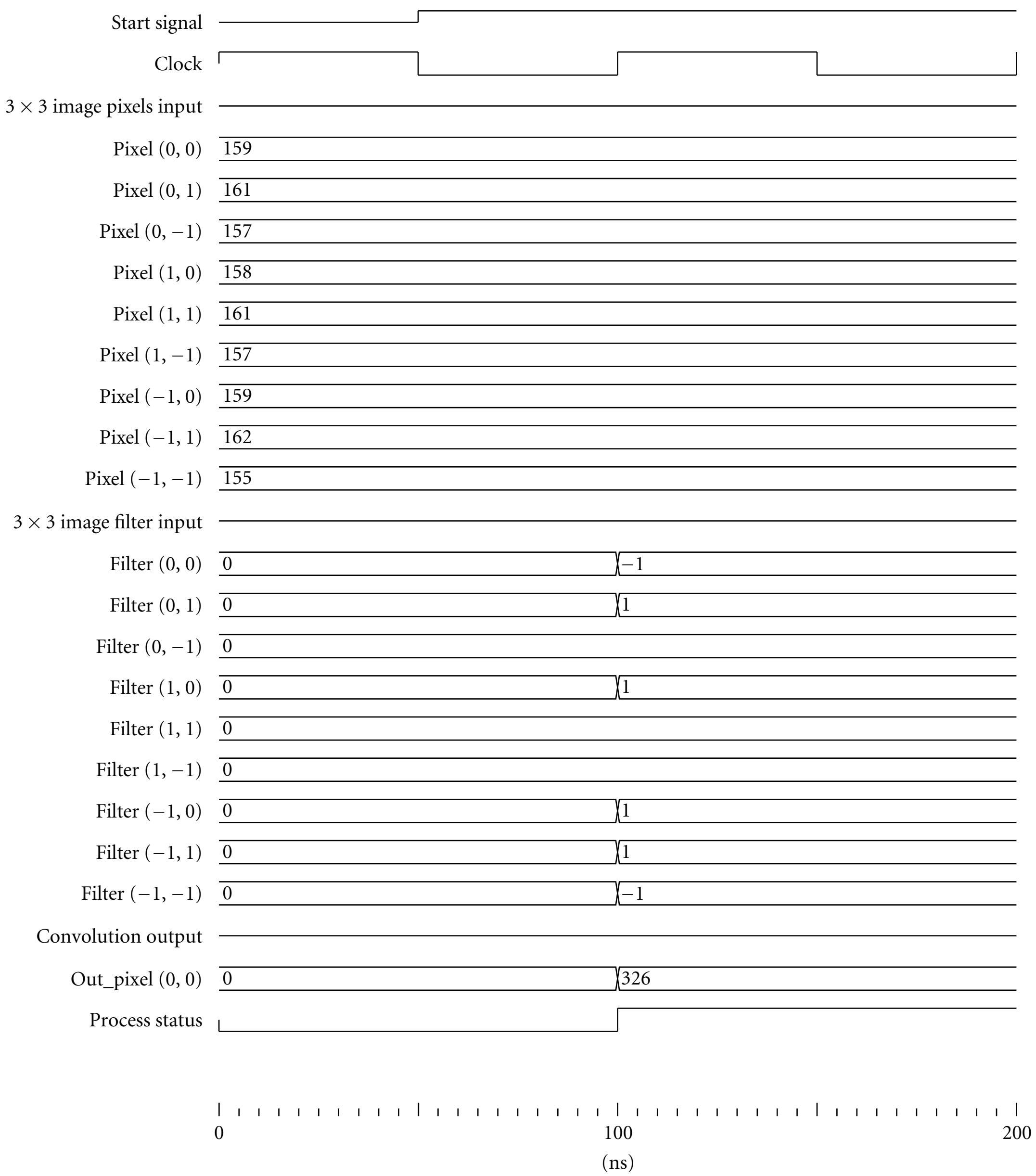


FIGURE 4: The hierarchical cell architecture for 2D convolution operator.



Entity: conv_top Architecture: Date: Fri Jun 10 9 : 14 : 39 PM Pacific Daylight Time 2011 Row: 1 Page: 1

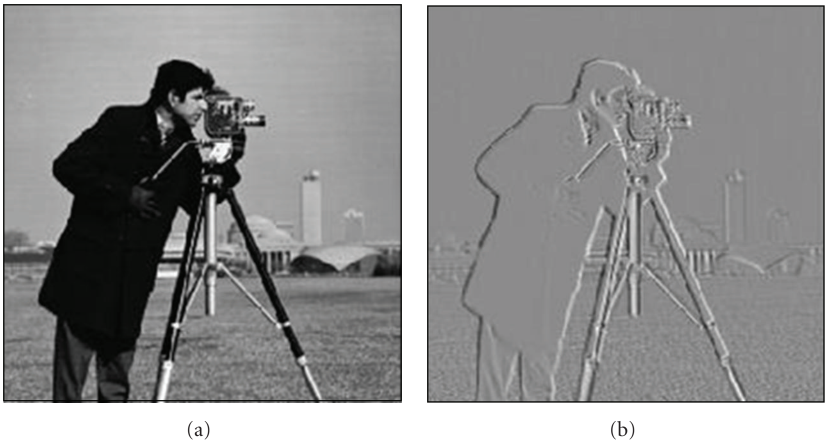
Figure 5: Convolution result of the convolution architecture performed on a single pixel.

float decimal values. Figure 2 shows the functional simulation results of the arithmetic block and Table 2 shows the corresponding numerical values of individual functional tests. When testing with an FPGA simulator, the obtained output does not support the float variables in the decimal scale, however, the 32 bit binary output value can be verified to be identical. The results in Figure 2 show that, within the duration of one clock cycle/1 ns, the cell is capable to perform the basic arithmetic operation independent of the size of the input binary representation.

Figure 3 shows the functional simulation results of the logic block and Table 3 shows its corresponding numerical values. The cell is tested with binary values to verify the different outputs of the logic operations stated in Table 3. The cell is verified to perform different logic operation within the execution time of one clock cycle (1 ns).

4. Convolution Architecture Design and Simulation

This section describes the overall design of the proposed artificial cell-based convolution architecture for image processing. Convolution in image processing is a linear operator over a matrix (image) by another one which is called "kernel." A 3 × 3 kernel is most widely used and can be used to implement various possible image operations [9]. Image filtering and normalization in image matching systems [10, 11] is the most common application of such convolution



(a)

(b)

Figure 6: An example of the input image (a) applied to the convolution processor. The image (b) shows the output of the convolution processor for the applied input image (b). Note that the image (b) is normalized for display purpose.

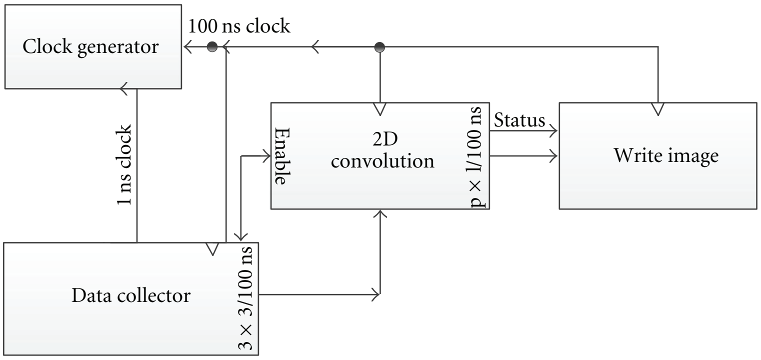


Figure 7: Function block diagram of ALU-based convolution.

operator. Equation (1) defines a convolution output C and shows how the convolution is performed, where G resembles the 3×3 kernel coefficients and F resembles the original image pixel with the surroundings (3×3 pixels and the original pixel is the middle of the matrix):

$$C(m \cdot n) = \sum_{u} \sum_{v} F[m+u][n+v] \cdot G[u][v]. \tag{1}$$

Figure 4 shows the register transfer level architecture of the convolution operation. The presented architecture has 9 pixel inputs from the image, and the architecture is set to convolute these inputs with a 3×3 window. It can be seen from Figure 4 that the cells are interconnected together in a hierarchical fashion to implement the convolution operation. The inputs of the cells are synchronized together through the input registers that are located in the blocks of individual cells. The first rows of the cells are set as multipliers that multiply the window coefficients with the corresponding input image pixels. The remaining cells are used for implementing the addition of the multiplied values. Overall, the architecture has 17 cells that implement the convolution of a single image pixel. It can be noted that the operator block governs the logical working of the cells by loading them with the relevant filter coefficients, and the relevant operation codes for addition and multiplication.

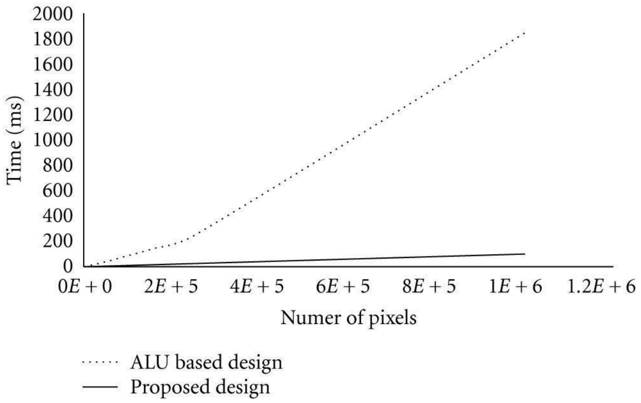


Figure 8: A comparison of convolution operator execution times of ALU design with that of proposed artificial-cell-based design.

Figure 5 shows the result of the convolution architecture. Here, the time required to perform the convolution operation over a single pixel is one clock cycle. The architecture can be applied in parallel to any arbitrary $n \times m$ image pixels so that the required processing time will remain at one clock cycle. For a demonstration of the proposed method, a 3×3 kernel matrix is randomly generated to take a value of $[-2\ 4\ 0,\ 0\ -4\ 1,\ -2\ 1\ 2]$. Figure 5 shows the out_pixel (0, 0) representing the filtered version of the center pixel (0, 0) in a 2D image convolution operation (as in (1)). Figure 6(a) shows the input image (all input pixels) applied to the convolution processor and Figure 6(b) shows the corresponding output image.

5. Comparison with ALU Implementation

An artificial cell-based 2D convolution processor is designed with a speed of 2 GHz without any negative slack. Figure 7 shows the functional block diagram of the test design and Figure 8 shows the time complexity results of proposed cell design and that of ALU-based convolution operation implemented in MATLAB run on an Intel(R) core TM 2 DUO CPU T6400@ 2.00 GHz-based computer with 2 GB of DDR RAM. It can be noted that the clock speeds of the proposed convolution processor and that of an ALU-based processor is kept same for fair comparison.

The design is composed of a clock generator that generates a 1 ns signal and a 100 ns signal. The data collector is synchronized to read 1 pixel every 1 ns. The data collector provides 3×3 pixel outputs with each pixel represented as 32 bits every 100 ns to the convolution architecture. The proposed architecture is set to compute the convolution every 100 ns, where the execution time is equal to one clock cycle. For every one clock cycle of 100 ns the architecture executes the pixel and writes it to a file.

Figure 8 shows the time complexity of executing the convolution operation when tested with different image sizes. The comparison in Figure 8 shows that the artificial cell-based design consistently outperforms the ALU-based design. Further, when using a parallel cellular architecture where all pixels are processed at the same time, the proposed approach resulted in a time complexity of one clock cycle per image. Although this parallel approach to implement convolution will add more number of circuit elements, the time complexity of the proposed design becomes independent of the size of the image.

6. Conclusions and Future Work

Inspired from the conceptual framework and properties of a cell, we form a generic artificial cell useful for implementing important machine vision function such as 2D image convolution. Using the concepts of cell hierarchy and multiplicity, we demonstrated the use of the proposed cell architecture in a practical application of image convolution. The proposed implementation of convolution operation when used in real time image processing system show lower time complexity than conventional ALU-based convolution implementation. Further, the parallel nature of the proposed architecture ensures the ease of scaling and robustness in functional implementation. Since convolution is a general operator that is used in almost all real-time digital image processing systems, the proposed system can be used to implement intelligent image processing cameras. The general idea of the artificial cell can be further exploited in designing high-speed object matching, tracking, image filtering, and recognition systems. In addition, the presented research can be extended to create more generalist cell architectures that can simulate and model complex processes such as protein-protein interactions, cell signaling, and high dimensional data processing.

References

[1] L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone, and A. Vespignani, "Artificial life X," in *Proceedings of the 10th International Conference on the Simulation and Synthesis of Living Systems*, MIT Press, 2006.

[2] R. A. Wilson and F. Keil, *The MIT Encyclopedia of the Cognitive Sciences*, The MIT Press, 1999.

[3] C. Adami, *Introduction to Artificial Life*, Springer, 1997.

[4] C. G. Langton, *Artificial Life*, Addison-Wesley, 1992.

[5] C. Szász and V. Chindriş, "Artificial life and communication strategy in bioinspired hardware systems with FPGA-based cell networks," in *Proceedings of the 11th International Conference on Intelligent Engineering Systems*, pp. 77–82, July 2007.

[6] V. I. Sobolev, "Convolution of functions," in *Encyclopaedia of Mathematics*, Kluwer Academic Publisher, 2001.

[7] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 26–33, 1986.

[8] V. Beiu, "A novel highly reliable low-power nano architecture when von Neumann augments Kolmogorov," in *Proceedings of 15th IEEE International Conference on Application-Specific Systems, Architectures and Processors*, pp. 167–177, September 2004.

[9] D. G. Bailey, *Design for Embedded Image Processing on FPGAs*, Wiley, 2011.

[10] A. P. James and S. Dimitrijev, "Inter-image outliers and their application to image classification," *Pattern Recognition*, vol. 43, no. 12, pp. 4101–4112, 2010.

[11] A. P. James and S. Dimitrijev, "Face recognition using local binary decisions," *IEEE Signal Processing Letters*, vol. 15, pp. 821–824, 2008.

The Scientific World Journal

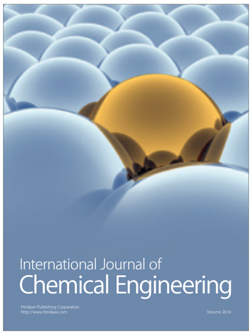
International Journal of Rotating Machinery

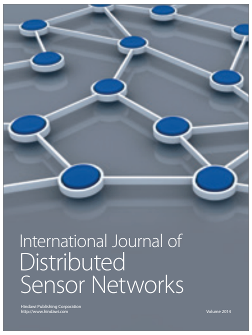
Journal of Robotics

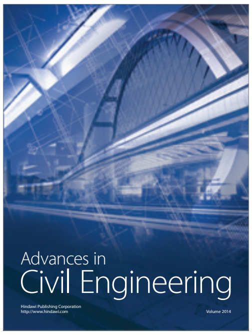
Journal of Sensors

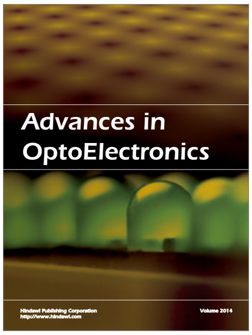
Advances in Mechanical Engineering

Journal of Engineering


International Journal of Chemical Engineering

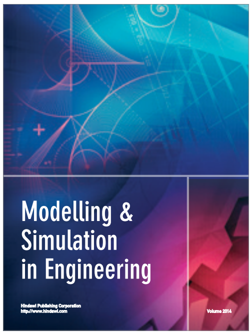

International Journal of Distributed Sensor Networks

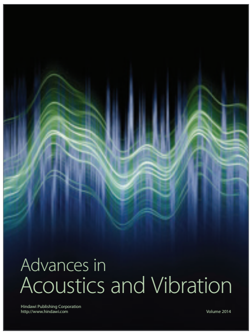

Advances in Civil Engineering


Advances in OptoElectronics

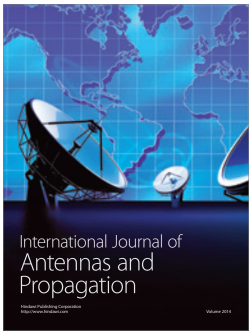
VLSI Design

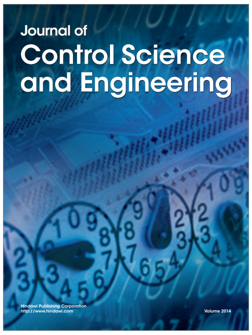
International Journal of Navigation and Observation

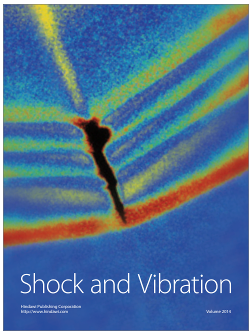

Modelling & Simulation in Engineering


Advances in Acoustics and Vibration

Active and Passive Electronic Components


International Journal of Antennas and Propagation


Journal of Control Science and Engineering


Shock and Vibration

Journal of Electrical and Computer Engineering