THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 16, Special Issue 2015, pp. 313-320

A PARALLEL ARCHITECTURE DESIGN FOR ULTRA-FAST IMAGE ENCRYPTION WITHIN WMSNS

Adrian-Viorel DIACONU

Lumina – The University of South-East Europe, IT&C Department, Romania
E-mail: adrian.diaconu@lumina.org

This paper aims to enhance performances of an image encryption algorithm by designing its parallel architecture, for the embedded computation in SoC (**S**ystem **o**n **C**hip) designs, using the means provided by CPA (**C**onnex **P**arallel **A**rchitecture). Complying with WMSNs' (**W**ireless **M**ultimedia **S**ensor **N**etworks) typical resources constraints and capabilities, *i.e.*, due to a faster and energy efficient design, algorithm's parallel architecture, as proposed and investigated, is proved to be suitable for usage within WMSNs.

Key words: ConnexArray^{TM}, image encryption, algorithm parallelization.

1. INTRODUCTION

With recent advances in wireless communication and embedded computation, large scales WMSNs are emerging in many applications, as shown in [1 – 5], facing with typical issues (*e.g.*, resources constraints and capabilities, *i.e.*, power consumption, QoS assurance, data storage etc.) and specific ones, more related to multimedia contents' processing techniques (*e.g.*, compression, authentication, encryption [6 – 9], respectively, parallel (or) distributed processing of multimedia contents [9 – 15]); and, with the amount of data stored and processed within, approaching or exceeding petabytes each year [10, 15], the problem of distributed storage and (or) parallel computing is addressed, being challenged both by typical WMSNs' constraints (*e.g.*, few storage and power resources, reduced computation capabilities etc.), *resp.*, parallel processing and storage system architecture design itself [10, 15 – 27, 50, 51].

Problem of parallel processing methods enjoys multiple views, *e.g.*, how to decompose a computationally intensive task into many 'small-sized' tasks which are executed on distributed sensors in parallel [10], in which case many similarities between the problems experienced in distributed WSNs and those of many-core systems are found [16], respectively, designing of new in-network processor architectures, tailored specifically to handling computationally intensive tasks at base station and even at sensor node level [28 – 32].

Current status of parallel computation, *„[...] a bad mixture between parallel structures and parallel algorithms [...]*" [33], *„[...] limited by absence of true parallel architectures [...]*" [33] has motivated the research towards designing of an integral parallel architectures for embedded computation [34, 35], making from resulted ConnexArrayTM SoCs, *i.e.*, CA1024 [36] and BA1024 [35 – 37], targeted competitive solution for multimedia content processing within WMSNs.

Among the recent proposals of in-network processor architectures, *e.g.*, [30 – 32], designed as to handle computationally intensive tasks, *e.g.*, image [32] and other types of multimedia content (pre) processing, the ConnexArray^{TM} (*i.e.*, cellular array which performs the intense part of computation, *e.g.*, CA1024 [36], a SoC design which integrates 1024 EUs, *i.e.*, Execution Units, running at 400MHz and with following measured operating performances: 120 GOPS/watt, respectively, 6.25 GOPS/mm^2 [34]) seemed to be the best current viable solution for designing of multimedia content processing algorithms' parallel architecture, *i.e.*, for high speed, quality processing, with respect to WMSNs typical constraints, such as limited power resources.

Block diagram of ConnexArray^{TM} is presented in Fig. 1. Here a linearly connected array of 1024 EUs receives the same instruction for each EU. The instruction is executed in each EU according to its own state. The reduction networks, designed for the most frequently used reduction functions (add, max etc.), sends back to the controller the requested data. An inner global loop, closed over the array, is used to classify the EUs according to the selected Boolean. The IO system works in parallel with and transparent to the main computation. The SoC CA 1024 contains besides the 1024 EUs (60% of the chip area) audio/video interfaces, a network of 4 MIPS and a time-parallel unit (8 16-bit processors) [34].

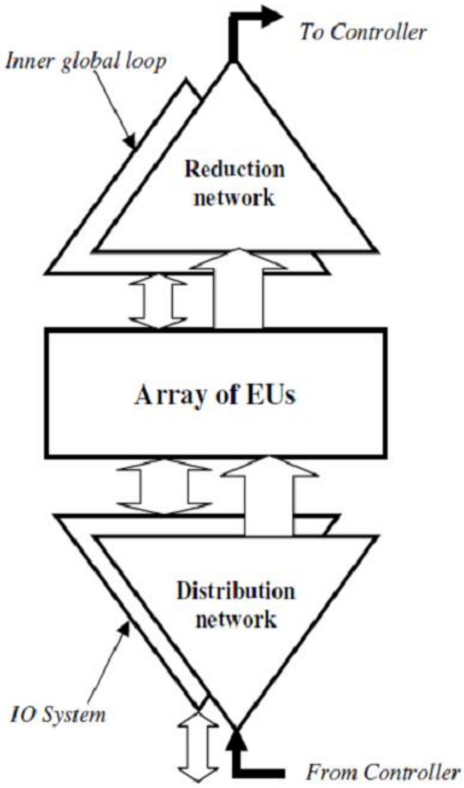


Fig. 1 – The ConnexArrayTM [34].

With the IPA of choice, image encryption algorithm based on Rubik's cube principle [38] was targeted in order to design its parallel architectures for embedded computation, *i.e.*, pursing to demonstrate its suitability for integration within WMSNs (that is, designed and implemented in such manner as to comply WMSNs' typical constraints, *e.g.*, limited power capabilities). Selection of this algorithm is doubly justified:

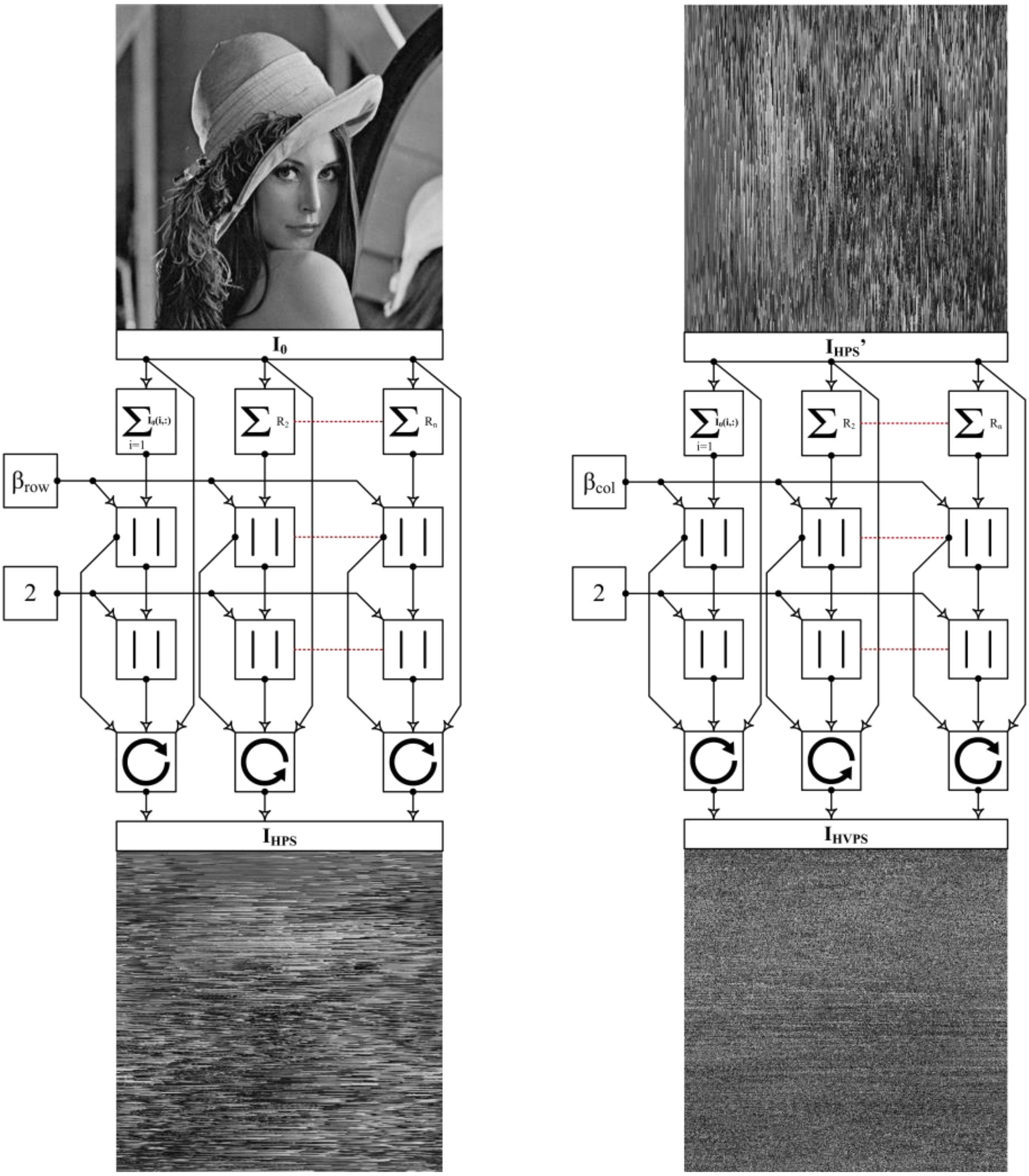
(i) it deals with a priority research interests within WMSNs, *i.e.*, ensuring secure communication mechanisms (since, due to the presence of sensitive multimedia data, *e.g.*, images, audio and (or) video streams containing certain information about individuals, in both an indirect or direct form, as in the case of telemedicine, *e.g.*, [39, 40], they are more vulnerable to security attacks);

(ii)due to its intrinsic design (*i.e.*, mode of operation, as shown in [38]), the selected algorithm is highly parallelizable (*i.e.*, in terms of computational implementation), in contrast with other newly proposed approaches.

Performances of resulted parallel designs were assessed by means of CPA simulator (developed using means offered by DrRacket platform [41]), as to highlight the benefits of using CPA in designing of parallel architecture for newly proposed, computationally intensive, multimedia content processing schemes; targeting the in-network integration, *i.e.*, at sensor node level, within WMSNs.

2. PARALLEL ARCHITECTURE DESIGN OF IMAGE ENCRYPTION ALGORITHM BASED ON RUBIK'S CUBE PRINCIPLE

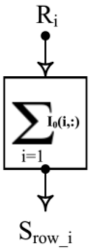
Assuming image's pixels' values pre-loaded in CPA's array, a number of m (where m is equal with image's dimensions, *e.g.*, $m = 512$ in our case) EUs (*i.e.*, Execution Units) are initialized. EUs will execute, simultaneously, a succession of basic operations (*i.e.*, sum of pixels' values, for an entire row R_i; two modulo n operations, where n has the value β_{row} or β_{col}, depending on pixels' shuffling direction, respectively, 2; and one row rotation, either to the left or to the right), in order to complete image's pixels shuffling procedure [38]), as shown in Fig. 2.



a b

Fig. 2 – Parallel architecture of the image encryption algorithm based on Rubik's cube principle – the scrambling procedure: a) first step; b) second step.

Preliminary information about each building block (*i.e.*, the input and output values, the associated CPA functions[1] etc.) are as follows:

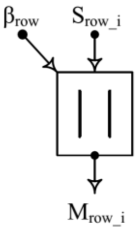


% computes the sum of all elements in row i

Input(s): CPA's array vector i, representing image's row$i$
Output:sum of all elements in row i, *i.e.*, $S_{row\_i} = \sum_{j=1}^{m} I_0(i,j)$; *(review [38])*

Associated CPA function:

(RedAddAll i)**OR**(ResetActive) followed by (RedAdd i).

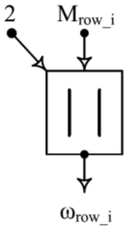


% computes modulo β_{row} of S_{row_i}

Input(s):sum of all elements in row i, *i.e.*, S_{row_i} and β_{row}
Output:modulo β_{row} of S_{row_i}, *i.e.*, $M_{row\_i} = S_{row\_i}(\mathrm{mod}\beta_{row})$; *(review [38])*

Associated CPA function:

(Remainder (RedAddAll i)β_{row}).

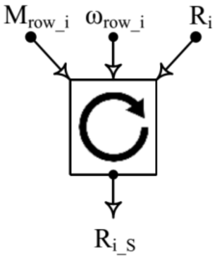


% computes modulo 2 of M_{row_i}

Input(s): modulo β_{row} of S_{row_i}, *i.e.*, M_{row_i} and 2
Output: modulo 2 of M_{row_i}, *i.e.*, $\omega_{row\_i} = M_{row\_i}(\mathrm{mod}\ 2)$; *(review [38])*

Associated CPA function:

(Remainder (Remainder (RedAddAll i) β_{row}) 2).



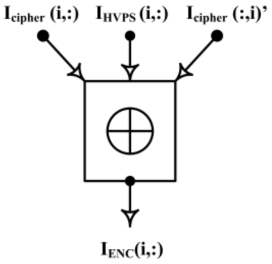
% circular-shift of row i, with M_{row_i}steps, either to the left or to the right, that is, depending on the value of ω_{row_i}; (review [38])

Input(s): ω_{row_i}, M_{row_i} and R_i

Associated CPA function:

```
(if (= (Remainder (Remainder (RedAddAll i) $\beta_{row}$) 2) 0)
    (RotateLeft  (Remainder (RedAddAll i) $\beta_{row}$)  i)
    (RotateRight (Remainder (RedAddAll i) $\beta_{row}$)  i)   ).
```

Fig. no. 3 showcases the parallel architecture of the module which handles the ciphering procedure. In this stage another building block appears (i.e., the bitwise XOR logical operator - based image ciphering block).



% image ciphering; (review [38])

Input(s): $I_{HVPS}(i,:)$, $I_{Cipher}(:,i)'$ and $I_{Cipher}(i,:)$, *i.e.*, rows' pixels from the shuffled image, associated matrix, *resp.*, from the ciphering matrix

Output:
ciphered image, *i.e.*, $I_{ENC}(i,:) = I_{HVPS}(i,:) \oplus I_{Cipher}(i,:) \oplus I_{Cipher}(:,i)'$

Associated CPA function:
(Xor (Xor (Vec i) $\mathrm{I}_{\mathrm{Cipher}}(\mathrm{i},:)$) $\mathrm{I}_{\mathrm{Cipher}}(:,\mathrm{i})'$).

[1] Full length articles and teaching materials about CPA (i.e., including description of the above functions' primitives) can be found at http://arh.pub.ro/gstefan/

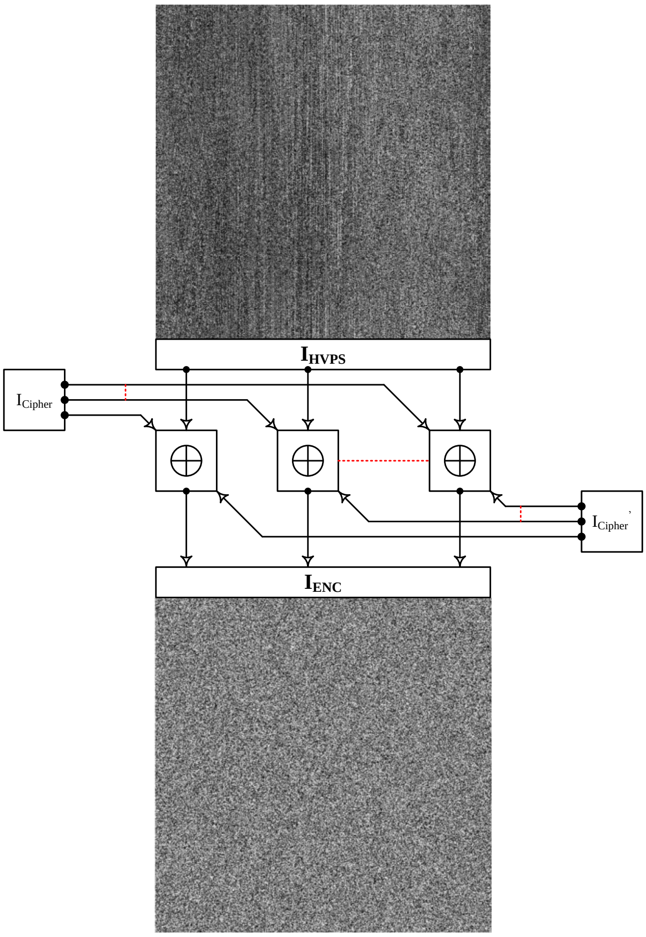


Fig. 3 – Parallel architecture of the image encryption algorithm based on Rubik's cube principle – the ciphering procedure.

3. PERFORMANCES ASSESSMENT

With EU's $Cycles_{Array} = 8$ and $Cycles_{Controler} = 99$ (*i.e.*, spent on the execution of basic operations mentioned above), maximum total number of cycles required to complete image's shuffling procedure can be evaluated using (1). To resulted number of cycles, extra 10% cycles are added (*i.e.*, being attributed to the transparent system's controller operations). Within (1), $Cycles_{Shift}$ is evaluated using (2). Here, l_{gray} represent image's maximum possible gray level's value, *i.e.*, 256, and thus evaluating the worst case scenario, that is, image's rows and columns are shifted with the maximum possible number of steps.

$$Total_{Cycles} = \alpha \cdot 2 \cdot \left(Cycles_{Array} + Cycles_{Controler} + Cycles_{Shift}\right) + 10\% \tag{1}$$

$$Cycles_{Shift} = \max\left(\left(\mathrm{l_{gray}} \cdot \mathrm{m}(\mathrm{mod}\,\beta_{row})\right), \left(\mathrm{l_{gray}} \cdot \mathrm{m}(\mathrm{mod}\,\beta_{col})\right)\right) \tag{2}$$

Knowing $Cycles_{Array} = 4$ (*i.e.*, supplementary number of machine cycles necessary to complete the image's ciphering procedure, that is, second stage of image encryption algorithm), the total number of cycles required to complete the encryption scheme is evaluated at the value of $6230\ cycles/image$.

Previously, it was assumed that image's pixels' values were pre-loaded in CPA's array. But, in order to make a complete and accurate evaluation of the proposed parallel design's performances, I/O bound hypothesis must be checked.

Considering I/O data transfers performed with a rate of 256 *bit/cycle*, *i.e.*, 32 *byte/cycle*, which implies 16 *cycles/vector* (where, vector's dimension is 512 *bytes i.e.*, image's dimension), a number of 8192 *cycles* are required for image's pre-loading stage. One can conclude that proposed parallel architecture is at I/O bound limit and, further, use this value for other performance's numerical evaluation.

In numbers, proposed parallel algorithm's design presents with following performances:

(i) knowing that processes of pre-loading a 512×512 image, respectively, of encrypting it are transparent one to each other (*i.e.*, they could be done in parallel, as also); with an average of 8192 *cycles/image*, by means of CPA, images are processed at a rate of 32 *pixels/cycle*.

(ii) considering use of the CA1024 SoC, with the system clock at 450 MHz (*i.e.*, with an equivalent time of 18.204 μs required to encrypt an image), a rate of 55 *Kframes/s* can be achieved.

Above improvements in the processing time, i.e., resulted encryption system is approximately 2000 times faster, in comparing with the original implementation [38], come with the following explanations:

(i) ConnexArray^{TM}, intrinsically designed to work with buffers (*i.e.*, data load is predictive and transparent) instead of cache memories, excludes, *e.g.*, the 'miss cache' possibility (*i.e.*, time consuming exceptional conditions);

(ii) as seen in Fig. no. 2 and Fig. no 3, algorithms' sequentially is replaced with multiple, parallel, blocks (*i.e.*, processing, at the same time, different image parts) and, by doing this, 'for' - type loops are avoided.

In terms of scalability, *i.e.*, how the input images' size influences the above performances (from throughput's rates point of view), taking into consideration that the actual images' processing speed is bounded by I/O data transfers, with the proposed parallel architecture for image encryption, the following rates can be achieved roughly: 3.46 Gbps (*i.e.*, for square images of the size *m=1024* pixels), 115.34 Gbps (*i.e.*, for square images of the size *m=512* pixels) and 344.45Gbps (*i.e.*, for square images of the size *m=256* pixels and fully pipelined implementation, that is, four images are simultaneously processed). Thus, with much less power consumption (*i.e.*, 160mW, on average), proposed implementation approaches comparable performances with other reported results (that is, of classical or new chaos-based algorithms, designed on different FPGA, multiple CPU or GPU based architectures), *i.e.*, [42-49].

4. CONCLUDING REMARKS

Nowadays designing of integral parallel architectures (targeting embedded computation) is a key research interest, motivated in the field of WMSNs by the amount of data stored and processed within. These IPAs, as all the algorithms implemented within, must be tailored specifically to handle computationally intensive tasks at base station and even at sensor node level, with respect to typical WMSNs' constraints (*i.e.*, power resources, computational capabilities etc.).

Whilst, in previous paper, research work was focused on designing of new multimedia contents' (*i.e.*, still images) processing schemes (*i.e.*, encryption), identifying ConnexArrayTM as an efficient and competitive IPA approach for the embedded computation, present paper has sought to design its parallel architecture. With the IPA of choice (*i.e.*, CA1024), in conjunction of CPA's CAD simulator, the previously proposed image processing algorithms was targeted, in order to design its parallel architecture, *i.e.*, image encryption algorithm based on Rubik's cube principle.

Performances' analysis highlighted that the mix between CPA (*i.e.*, due to its reduced power consumption, in comparison with computing capabilities, that is, 120 GOPS/watt) and proposed image processing schemes (*i.e.*, with highly parallelizable concept), is suitable for integration within WMSNs (due to fast time execution and low power consumption, *i.e.*, the two main operational constraints faced within WMSNs exploitation).

REFEREZNCES

1. I.F. AKYILDIZ, T. MELODIA, K.R. CHOWDHURY, *A survey on wireless multimedia sensor networks*, Computer Networks, **51**, *4*, pp. 921-960, 2007.
2. A. SHARIF, V. POTDAR, E. CHANG, *Wireless multimedia sensor network technology: A survey*, in Proc. of 7th Int. Conf. on *Industrial Informatics*, pp. 606-613, Cardiff, Wales, UK, 23-26 June 2009.
3. S. SORO, W. HEINZELMAN, *A survey of visual sensor networks*, Advances in Multimedia, **2009**, Article ID: 640386, pp. 1-21, 2014. DOI: 10.1155/2009/640386.
4. I.T. ALMALKAWI, M.G. ZAPATA, J.N. AL-KARAKI, J. MORILLO-POZO, *Wireless multimedia sensor networks: Current trend and future directions*, Sensors (Switzerland), **10**, *7*, pp. 6662-6717, 2010.
5. I.F. AKYILDIZ, T. MELODIA, K.R. CHOWDHURY, Wireless multimedia sensor networks: applications and test beds, Proc. of the IEEE, **96**, *10*, pp. 1588-1605, 2008.
6. M.G. ZAPATA, R. ZILAN, J.M. BARCELO-ORDINAS, K. BICAKCI, B. TAVLI, *The future of security in wireless multimedia sensor networks*, Telecommunication Systems, **45**, *1*, pp. 77-91, 2010.
7. L.A. GRIECO, G. BOGGIA, S. SICARI, P. COLOMBO, *Secure wireless multimedia sensor networks: a survey*, Proc. of 3rd IEEE Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies, pp. 194-201, Silema, Malta, 11-16 October 2009.
8. B. HARJITO, S. HAN, Wireless multimedia sensor networks applications and security challenges, Proc. of IEEE Int. Conf. on Broadband, Wireless Computing, Communication and Applications, pp. 842-846, Fukuoka, Japan, 4-6 November 2010.
9. A. MAMMERI, B. HADJOU, A. KHOUMSI, A survey of image compression algorithms for visual sensor networks, ISRN Sensor Networks, **2012**, Article ID: 760320, pp. 1-19, 201. DOI: 10.5402/2012/760320.
10. C. JARDAK, J. RIIHIJARYI, F. OLDEWURTEL, P. MAHONEN, *Parallel processing of data from very large scale* wireless sensor networks, Proc. of 19th ACM Int. Symp. on High Performance Distributed Computing, pp. 787-794, Chicago, IL, 21-25 June 2010.
11. A.T. ZIMMERMAN, M. SHIRAISHI, R.A. SWARTZ, J.P. LYNCH, *Automated modal parameter estimation by parallel processing within wireless monitoring systems*, Journal of Infrastructure Systems, **14**, *1*, 102-113, 2008.
12. A. MOTA, L.B. OLIVIERA, G.P. SAFE, F.F. ROCHA, R. RISERIO, A.A.F. LOUREIRO, C.J.N. COELHO, H.C. WONG, E. NAKAMURA, WISENEP: A network processor for wireless sensor networks, Proc. of 11th IEEE Symp. on *Computers and Communications*, pp. 8-14, Sardinia, Italy, 26-29 June 2006.
13. F.J. WU, Y.F. KAO, Y.C. TSENG, From wireless sensor networks towards cyber physical systems, Pervasive and Mobile Computing, **7**, *4*, pp. 397-413, 2011.
14. A. BAKSHI, V.K. PRASANNA, Algorithm design and synthesis for wireless sensor networks, Proc. of 33rd IEEE Int. Conf. on *Parallel Processing*, **1**, pp. 423-430, Quebec, Canada, 15-18 August 2004.
15. Y. WANG, Y. WANG, Distributed storage and parallel processing in large-scale wireless sensor networks, High Performance Computing Workshop, pp. 288-305, 2010.
16. J. TRAUE, R. KARNAPKE, J. NOLTE, From parallel systems to wireless sensor networks and back, Technical Report SEEMOO-TR-2012-03, TU Darmstadt, pp. 29-32, 2012.
17. K.-C. KIM, C.-S. KIM, Parallel processing of sensor network data using the column oriented databases, AASRI Procedia, **5**, pp. 2-8, 2010.
18. M. LI, Y. WANG, Y. WANG, Complexity of data collection, aggregation, and selection for wireless sensor networks, IEEE Trans. On Computers, **60**, 3, pp. 386-399, 2011.
19. S.R. MANGALWEDE, D.H. RAO, Performance analysis of Java-based approaches to distributed computing, Int. J. of Recent Trends in Engineering, **1**, *1*, pp. 556-559, 2009.
20. B. ZHANG, S. LI, Survey of network management protocols in wireless sensor networks, Proc. of IEEE Int. Conf. on e-Business and Information System Security, pp. 1-5, Wuhan, China, 23-24 May 2009.
21. W. ZHANG, G. GAO, T. LAPORTA, Data dissemination with ring-based index for wireless sensor networks, IEEE Trans. on Mobile Computing, **6**, *7*, pp. 832-847, 2007.
22. Y. WU, Y. LI, Distributed indexing and data dissemination in large scale wireless sensor networks, Proc. of 18th IEEE Int. Conf. on Computer Communications and Networks, pp. 1-6, San Francisco, CA, USA, 3-8 August 2009.
23. J. DEAN, S. GHEMAWAT, *MapReduce: simplified data processing in large clusters*, Communications of the ACM, **51**, *1*, pp. 107-113, 2008.
24. N.J. AL-KARAKI, A.E. KAMAL, *Routing techniques in wireless sensor networks: a survey*, IEEE Wireless Communications, **11**, *6*, pp. 6-28, 2004.
25. R. BOSE, *Sensor networks motes, smart space, and beyond*, IEEE Pervasive Computing, **8**, *3*, pp. 84-90, 2009.
26. M. KUORILEHTO, M. HA, T.D. HA, *A survey of application distribution in wireless sensor networks*, EURASIP J. on Wireless Communications and Networking, *5*, pp. 774-788, 2005.
27. E. FASOLO, M. ROSSI, WIDMER J., ZORZI M., *In-network aggregation techniques for wireless sensor networks: a survey*, IEEE Wireless Communications, **14**, *2*, pp. 70-78, 2007.
28. H. ZHANG, A. ARORA, GS3: scalable self-configuration and self-healing in wireless sensor networks, Computer Networks, **43**, 4, pp. 459-480, 2003.
29. D. GANESAN, D. ESTRIN, J. HEIDEMANN, *DIMENSIONS: Why do we need a new data handling architecture for sensor networks*, ACM Computer Communication Review, **33**, *1*, pp. 143-148, 2003.

30. C. WALRAVENS, W. DEHAENE, Low-power digital signal processor architecture for wireless sensor nodes, IEEE Trans. on Very Large Scale Integration (VLSI) Systems, **22**, 2, pp. 313-321, 2014.
31. I. GALPIN, C.Y. BRENNINKMEIJER, A.J. GRAY, F. JABEEN, A.A. FERNANDES, N.W. PATON, *SNEE: a query processor for wireless sensor networks*, Distributed and Parallel Databases, **29**, *1-2*, pp. 31-85, 2011.
32. D.M. PHAM, S.M. AZIZ, *FPGA-based image processor architecture for wireless multimedia sensor network*, Proc. of 19th IEEE Int. Conf. on Embedded and Ubiquitous Computing, pp. 100-105, Melbourne, VIC, 24-26 October 2011.
33. G.M. STEFAN, *Parallel architecturing starting from natural computational models*, Proc. of the Romanian Academy, Series A: Mathematics, Physics, Technical Sciences, Information Sciences, **1**, *3*, pp. 1-6, 2000.
34. G.M. STEFAN, *Integral parallel architecture in System-on-Chip designs*, Proc. of 6th Int. Workshop on Unique Chips and Systems, pp. 23-26, Atlanta, GA, USA, 4th December 2010.
35. M. MALITA, G.M. STEFAN, D. THIEBAUT, *Not multi-, but many-core: designing integral parallel architecture for embedded computation*, ACM Computer Architecture News, **35**, *5*, pp. 32-38, 2007.
36. L. BIVOLARSKI, B. MITU, A. SHEEL, G.M. STEFAN, T. THOMSON, D. TOMESCU, *The CA1024: A fully programmable System-on-Chip for cost-effective HDTV media processing*, Hot Chips: A Symposium on *High Performance Chips*, Stanford University, 20-22 August 2006.
37. I. LORENTZ, M. MALITA, R. ANDONIE, *Evolutionary computation on the Connex Architecture*, Proc. of the 22nd Midwest Artificial Intelligence and Cognitive Science Conference, pp. 146-153, Ohio, USA, 16-17 April 2011.
38. A.-V. DIACONU, K. LOUKHAOUKHA, *An improved secure image encryption algorithm based on Rubik's cube principle and digital chaotic cipher*, Math. Prob. Eng., **2013**, Article ID: 848392, pp. 1-10, 2013. doi: 10.1155/2013/848392.
39. F. HU, S. KUMAR, QoS considerations in wireless sensor networks for telemedicine, Proc. of SPIE, **5242**, pp. 217-227, 2003.
40. A.A. REEVES, Remote monitoring of patients suffering from early symptoms of Dementia, Int. Workshop on Wearable and Implantable Body Sensor Networks, London, UK, 12-13 April 2005.
41. M. FLATT, R.B. FINDLER, *Welcome to Racket* [http://docs.racket-lang.org/guide].
42. S.-M. YOO, D. KOTTURI, D.W. PAN, Blizzard J., *An AES crypto chip using a high-speed parallel pipelined architecture*, Microprocessors and Microsystems, **29**, *7*, pp. 317-326, 2005.
43. L. ALI, I. ARIS, F.S. HOSSAIN, N. ROY, *Design of an ultra-high speed AES processor for next generation IT security*, Computers and Electrical Engineering, **37**, *6*, pp. 1160-1170, 2011.
44. J.J. RODRIGUEZ-VAZQUEZ, S. ROMERO-SANCHEZ, M. CARDENAS-MONTES, *Speeding up a chaos-based image encryption algorithm using GPGPU*, Computer Aided Systems Theory – EUROCAST 2011, Lecture Notes in Computer Science, **6927**, pp. 592-599, 2012.
45. A. HODJAT, I. VERBAUWHEDE, *Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors*, IEEE Trans. on Computers, **55**, *4*, pp. 366-372, 2006.
46. Z. YANG, Y. ZHU, Y. PU, *Parallel image processing based on CUDA*, Proc. of IEEE Int. Conf. on *Science and Software Engineering*, **3**, pp. 198-201, Wuhan, China, 12-14 December 2008.
47. X. CAI, R. SUN, J. LIU, *An ultrahigh speed AES processor method based on FPGA*, Proc. of 5th IEEE Int. Conf. on *Intelligent Networking and Collaborative Systems*, pp. 633-636, Xi'an, China, 9-11 September 2013.
48. C. KACHRIS, N. NOURNAKIS, A. DOLLAS, *A reconfigurable logic-based processor for the SCAN image and video encryption algorithm*, International Journal of Parallel Programming, **31**, *6*, pp. 489-506, 2003.
49. C.W. HUANG, C.J. CHANG, M.Y. LIN, H.Y. TAI, *The FPGA implementation of 128-bits AES algorithm based on four 32-bits parallel operation*, Proc. of the 1st IEEE Int. Symp. on *Data, Privacy and e-commerce*, pp. 462-464, Chengdu, China, 1-3 November 2007.
50. V.M. IONESCU, I. LITA, D. VISAN, B. CIOC, *Architecture for adders in digital filters operating in mixed power modes*, Proc. of the 36th Int. Spring Seminar on *Electronics Technology*, pp. 397-400, Alba Iulia, Romania, 8-12 May 2013.
51. V.M. IONESCU, I.BOSTAN, L. IONESCU, *Systemic design for integrated digital circuit structures*, Proc. of the IEEE Int. Semiconductor Conf., **2**, pp. 467-470, Sinaia, Romania, 4-6 October 2004.