HARDWARE-BASED GENETIC ALGORITHM IMPLEMENTATION

IN FIELD PROGRAMMABLE GATE ARRAY

SATHIVELLU S/O BALAKRISHNAN

UNIVERSITI TEKNOLOGI MALAYSIA

HARDWARE-BASED GENETIC ALGORITHM IMPLEMENTATION IN FIELD PROGRAMMABLE GATE ARRAYS

SATHIVELLU S/O BALAKRISHNAN

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical Engineering)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

MARCH 2012

Dedicated, in thankful appreciation for support, and encouragement to my beloved family.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my research supervisor, Professor Dr. Haji Mohamed Khalil bin Haji Mohd Hani for his support and patient guidance throughout this research. His constant encouragement, critics and guidance is the key of bringing this research to a fruitful completion.

Besides, I also would like to take this opportunity to thank Dr. Muhammad Nadzir bin Marsono, Dr. Hau Yuan Wen, Ms. Rabia Bakhteri, Vishnu P. Nambiar, Eng Pei Chee, Mohamed Annuar, Liew Teik Yee, and Moganesh who provide me moral support, advices and guidance directly or indirectly in my research.

On a personal note, I thank my family and friends for their seamless caring, encouragement and moral support that enables me to complete this journey.

ABSTRACT

Genetic Algorithm (GA) is inspired by natural selection and evolution in a computer program. It has been shown to be effective in solving search and optimization problems. However, research has shown that software implementations of GA in complex problems usually lead to unacceptable optimization delays. Hence, hardware-based GA solutions are needed, especially in systems that require real-time performance. However, full hardware implementation of GA eliminates its flexibility to be reused in other applications. This is because some of the GA operations are highly problem-dependent. Consequently, this thesis proposes a hardware-based GA (HGA) that provides configurability, scalability and flexibility. The proposed reconfigurable HGA is implemented on Altera Stratix II EP2S60 FPGA prototyping board with a clock frequency running at 50 MHz. Hardware-software co-design technique is applied. The system partitioning is done based on the following aspects: (a) system constraints (b) compute-intensive operations (c) sequential operations (d) bottlenecks during system bus access (e) logic cost, and (f) ability to reconfigure. In this work, the HGA is deployed in a number of test case studies, which include optimization of a simple fitness function, a complex Michalewicz's function, and a real-world finger-vein image processing application. The real-world problem is to apply the GA to optimize the tuning of parameters in a finger-vein image processing biometric subsystem. Experimental results show that the proposed HGA achieves a good degree of configurability and flexibility in handling a variety of problems. The HGA is about three times faster compared to its software equivalent. The equal error rate of the finger-vein biometric system is improved from 1.004% to 0.101%. This shows that the proposed design is capable to optimize the tuning of the parameter set in this image processing application.

ABSTRAK

Algoritma Genetik (GA) berinspirasikan pemilih alam dan evolusi di dalam program komputer. Ianya telah terbukti berkesan dalam menyelesaikan masalah carian dan pengoptimuman. Walaubagaimanapun, kajian telah menunjukkan bahawa pelaksanaan perisian GA dalam masalah yang kompleks biasanya menyebabkan kelewatan dalam proses pengoptimuman yang tidak boleh diterima. Oleh kerana itu, penyelesaian GA yang berasaskan perkakasan diperlukan, terutamanya dalam sistem yang memerlukan prestasi masa nyata. Namun, pelaksanaan penuh GA dalam perkakasan, menyisihkan GA daripada digunakan semula dalam aplikasi lain. Hal ini kerana beberapa operasi dalam GA sangat bergantung kepada masalah yang disasarkan. Oleh itu, tesis ini mencadangkan satu reka bentuk perkakasan GA (HGA) yang menyediakan kemudahan untuk dikonfigurasikan, skalabilitas dan fleksibiliti. HGA yang dicadangkan dilaksanakan pada papan prototaip Altera FPGA II Stratix EP2S60 dengan frekuensi jam yang beroperasi pada 50 MHz. Perkakasan-perisian co-desain teknik diterapkan. Sistem dipartisikan berdasarkan kepada aspek-aspek berikut: (a) kekangan sistem (b) operasi mengira-intensif (c) operasi sekuensial (d) hambatan semasa mengakses bas sistem (e) kos untuk logik digunakan, dan (f) kemampuan untuk mengubah semula konfigurasi. Dalam tesis ini, HGA digunakan dalam beberapa kajian kes, ianya termasuk pengoptimasian fungsi kemampuan yang mudah, fungsi kompleks Michalewicz, dan satu aplikasi dunia nyata. Aplikasi dunia nyata ini adalah penerapan GA untuk mengoptimumkan penalaan parameter untuk pemprosesan gambar vaskular jari dalam subsistem biometrik. Keputusan kajian menunjukkan bahawa HGA yang dicadangkan mencapai tahap yang baik dalam menyediakan kemudahan untuk dikonfigurasikan dan fleksibiliti dalam menangani pelbagai masalah. HGA menghitung sekitar tiga kali ganda lebih cepat berbanding dengan perlaksanaan GA dalam perisian. Nilai *Equal Error Rate* untuk sistem biometrik vaskular jari dapat ditingkatkan dari nilai 1.004% kepada nilai 0.101%. Hal ini menunjukkan bahawa reka bentuk yang dicadangkan adalah cekap dalam mengoptimasikan penalaan parameter yang ditetapkan dalam aplikasi pemprosesan gambar.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENTS	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	x
	LIST OF FIGURES	xi
	LIST OF ABBREVIATIONS	xiv
	LIST OF SYMBOLS	xvi
	LIST OF LISTINGS	xvii
	LIST OF APPENDICES	xviii
1	**INTRODUCTION**	1
	1.1 Background	1
	1.2 Problem Statement	3
	1.3 Objective	4
	1.4 Scope of Work	4
	1.5 Research Contribution	6
	1.6 Thesis Organization	7

2	**LITERATURE REVIEW AND BACKGROUND**		**8**
	2.1	Previous Works on GA Hardware	8
	2.2	Related Work on GA Applied in Constrained Problems	11
	2.3	GA Applied in Parameter Optimization in Image Processing	12
	2.4	The Simple Genetic Algorithm	14
		2.4.1 Basic Behaviour and Configuration Issues	14
		2.4.2 Initial Population, Representation and Fitness Evaluation	16
		2.4.3 Selection, Reproduction and Replacement	17
		2.4.4 Random Number Generation	20
	2.5	Summary	21
3	**METHODOLOGY**		23
	3.1	Research Work Flow	23
	3.2	HW/SW Co-design Methodology	26
	3.3	Test Case Studies for GA Design Analysis	28
		3.3.1 Test Case Studies 1 and 2 - Unconstrained and Constrained Types	28
		3.3.2 Optimizing the Tuning of Parameters in a Finger-Vein Biometric Image Processing	30
		3.3.2.1 Image Processing Operation and Its Tuneable Parameters	33
	3.4	Summary	38
4	**DESIGN AND IMPLEMENTATION OF HARDWARE-BASED GA (HGA)**		40
	4.1	Software-based GA (SGA) Design	40
	4.2	Design of HGA – Architectural Issues	43
	4.3	Design of HGA - Configurability and Scalability	46

	4.4	RTL Design of GA Hardware Core	49
		4.4.1 GA Control Unit	56
		4.4.2 Data Path Unit (DPU)	58
		4.4.2.1 GA Memory Block	60
		4.4.2.2 *Find_Solution_Stopping* Routine	60
		4.4.2.3 Selection Module	62
		4.4.2.4 Crossover Module	63
		4.4.2.5 Mutation Module	66
		4.4.2.6 Replacement Module	69
	4.5	GA Coprocessor - Avalon Interface and Driver Firmware	69
5	**RESULT AND DISCUSSION**		74
	5.1	Performance Profiling of Software-based GA (SGA)	74
	5.2	Hardware - based GA (HGA) - Simulation Results	77
	5.3	HGA Applied in Test Case Study 1	79
		5.3.1 Timing Performance of HGA	79
		5.3.2 Behaviour of the GA for Test Case Study 1	80
	5.4	HGA Applied in Test Case Study 2	82
		5.4.1 Timing Performance of HGA	84
		5.4.2. Behaviour of the GA for Test Case Study 2	85
	5.5	Real-World Application (Finger-Vein Biometric System)	88
		5.5.1 Behaviour of the GA for Test Case Study 3	92
	5.6	Summary	95
6	**CONCLUSIONS AND RECOMMENDATIONS**		97
	6.1	Conclusion	97
	6.2	Recommendations for Future Work	98

REFERENCES 101

Appendices A-H 107-168

LIST OF TABLES

TABLE NO.	TITLE	PAGE
4.1	GA algorithm type in proposed HGA	48
4.2	RTL code of GA Hardware Core.	52
5.1	Performance profiling of SGA in test case 1 (worst-case processing).	75
5.2	Performance profiling of SGA in test case 2 (worst-case processing).	76
5.3	Resource usage for each GA hardware module in DPU	77
5.4	Speed comparison with previous hardware-based GA executed for 20 generations test with specified fitness function and genetic parameters.	79
5.5	Search space for each solution parameter, x_1 and x_2.	83
5.6	Speed comparison of the proposed HGA with previous work in [9].	84
5.7	GA implementation in Michalewicz function	87
5.8	Finger-vein biometric system's EER value	94

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Top-level architecture for the proposed system	6
2.1	Behavioural flow chart of a simple GA [19]	15
2.2	The examples of encoding. (a) Octal encoding (b) Hexadecimal encoding (c) Value encoding	17
2.3	Simple representation of tournament selection operation	18
2.4	The examples of crossover. (a) Single point crossover (b) Double point crossover (c) Multiple point crossover	19
2.5	The examples of mutation. (a) Single bit inversion. (b) Multiple bits inversion	19
2.6	Block diagram for RNG in GA	21
3.1	Research workflow	24
3.2	HW/SW co-design flow chart [41]	27
3.3	Graph for Michalewicz's function [9]	29
3.4	Process involves during training mode in finger-vein biometric system	31
3.5	Probability Density Function (PDF) of Genuine and Imposter Score in Dissimilarity Measurement System	32
3.6	Process involves during finger-vein verification operation	33

3.7	Image processing operation with identified tuneable parameters in adopted finger-vein biometric system.	37
4.1	Top-level behaviour of proposed HGA design	42
4.2	Architecture of HGA – System Partitioning	43
4.3	Generating initial population in constrained problem	44
4.4	General behavioural flow of mutation in constrained problem	45
4.5	Data structure for a chromosome in the proposed HGA.	49
4.6	System architecture of the proposed HGA	49
4.7	ASM chart modelling the RTL behaviour of the GA HW core	51
4.8	Functional block diagram of *GA_CU*.	57
4.9	Functional block diagram of DPU of GA HW core	59
4.10	Functional block diagram of GA memory block	60
4.11	Input/output block diagram of crossover module.	65
4.12	Functional block diagrams of crossover mask generator module.	65
4.13	Functional blocks diagrams of crossover module.	66
4.14	Block diagram of mutation module.	68
4.15	Functional block diagram of mutation operation	68
4.16	Functional block diagram of GA HW core	70
4.17	Avalon Interface Register	71
4.18	The source code for the newly defined macros.	72
4.19	Behavioural flow of the embedded software for the proposed hardware-based GA.	73
5.1	Results of profiling of SGA in test case study 1.	75
5.2	Results of profiling of SGA in test case study 2.	76
5.3	Timing simulation of GA HW core.	78
5.4	Behaviour of GA through the generations	81
5.5	Genotype and phenotype mapping	83
5.6	Simulation output for Michalewicz's function obtained using proposed GA. (a) Simulation output using multiple crossover, random parent selection	86

	and weak parent replacement (b) Simulation output using multiple crossover, tournament parent selection and weak parent replacement	
5.7	The behavioural model for automated optimization for image processing's tuneable parameters in finger-vein verification system.	89
5.8	Representation of each gene in chromosome in its solution parameters.	90
5.9	Process involve during fitness evaluation operation	91
5.10	Average fitness value against every 10 generations for setting 1.	93
5.11	The best EER value obtained from simulation output in Figure 5.10 plotted for the last 500th generations.	93
5.12	Output finger-vein image for each operation in image processing module and the extracted minutiae points using best parameters set obtained from the proposed system.	95
6.1	Proposed enhancement of the DPU in future HGA design.	99
6.2	Scheduling in the *Find_Solution_Stopping* function	100

LIST OF ABBREVIATIONS

ASM	-	Algorithmic state machine
bmp	-	bitmap
CAD	-	Computer-aided design
CST	-	Control signal table
CPU	-	Central processing unit
DFG	-	Data flow graph
EER	-	Equal error rate
FAR	-	False accept rate
FRR	-	False reject rate
FPGA	-	Field Programmable Gate Arrays (FPGA)
GA	-	Genetic algorithm
GCC	-	GNU Compiler Collection
GENOCOP	-	Genetic algorithm for numerical optimization for constrained problems
GPM	-	Genotype and phenotype mapping
HDL	-	Hardware description language
HGA	-	Hardware-based GA
HW	-	Hardware
IDE	-	Integrated development environment
MT19937	-	Mersenne Twisted
OS	-	Operating system
PC	-	Personal computer
PDF	-	Probability Density Function
RISC	-	Reduced instruction set computing

RNG	-	Random number generator
ROI	-	Region of interest
RTL	-	Register transfer level
SDL	-	Simple direct media layer
SDRAM	-	Synchronous dynamic random access memory
SGA	-	Software-based GA
SRAM	-	Static random-access memory
SOPC	-	System-on-a-Programmable-Chip
SW	-	Software
USB	-	Universal serial bus

LIST OF SYMBOLS

C_r	-	Crossover rate
g	-	Number of generations
H_T	-	High threshold
L_T	-	Low threshold
MHz	-	Mega Hertz
M_r	-	Mutation rate
ms	-	milliseconds
ns	-	nanoseconds
P_c	-	Crossover probability
ps	-	Population size
P_m	-	Mutation probability

LIST OF LISTINGS

LISTING NO.	TITLE	PAGE
4.1	Algorithm for *Find_Solution_Stopping* routine	61
4.2	Algorithm for selection module.	63
4.3	Algorithm for crossover operation	64
4.4	Algorithm for mutation operation.	67

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	C/C++ Source Code of the Implemented Software-based GA	108
B	C/C++ Source Code of Device Driver for the Proposed Hardware-based GA	119
C	Analysis of Processing Time for the Proposed Software and Hardware-based GA	123
D	Control Signal Table and RTL Code	131
E	Source Code for GA Hardware Modules in Verilog Hardware Description Language (HDL)	137
F	Functional Simulation Waveform for GA Hardware Module	168
G	Proposed Algorithm for GA with Elitism and Generational Replacement Type.	182
H	Parallel GA Architecture	183

CHAPTER 1

INTRODUCTION

1.1 Background

Genetic Algorithm (GA) was first introduced by Holland in 1975, GA is based on the idea of evolutionary computing which was introduced by Ingo Rechenberg in 1960 [1]. The GA methodology is about applying natural selection and genetic evolution to solve optimization and searching problem [2]. GA is a well known and a powerful tool for solving search and optimization problems [3]. Besides, GA has been theoretically and empirically proven to provide robust search in complex spaces [1].

GA obeys Darwin's evolutionary theory, "survival of the fittest" where the fitter individuals survive and the weaker individuals die [2]. GA imitates genetic evolution in the form of a computer program. In GA, a chromosome represents a possible solution, and the chromosome is subdivided into genes where genes represent parameters or characters of a problem's solution. Hence in a computer program, the chromosome is in the form of a binary string that have single or multiple genes where each gene has its own position in chromosome.

There are many other algorithms that are conventionally used to solve optimization and search problems such as random search, hill climbing and simulated annealing [4]. GA is different from the conventional problem-solving methods in very fundamental ways [2]:

REFERENCES

1. Holland, J. H. *Adaptation in Natural and Artificial Systems*, Ph.D thesis, University of Michigan Press. Ann Arbor, MI. 1975.
2. Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Massachusetts: Addison-Wesley Publishing Company. 1989.
3. Sivanandam S. N. and Deepa, S. N. *Introduction to Genetic Algorithms*, New York: Springer Berlin Heidelberg. 2008.
4. Weise, T. *Global Optimization Algorithms: Theory and Application*, Thomas Weise. 2008.
5. Nearchou, A.C. Path Planning of a Mobile Robot Using Genetic Heuristics, *Robotica.* Cambridge University Press. 2000. vol. 16: 575-588.
6. Tirumalai, V. *Hardware Implementation of a Parallelized Genetic Algorithm for Task Scheduling*. Master thesis. University of Alabama, Tuscaloosa, Alabama. 2006.
7. Jiménez, J.L. *Hardware/Software Architectures for Iris Biometrics*, Ph.D thesis. Departmento de Tecnología Electrónica, Universidad Carlos III de Madrid. 2009.
8. Zhu, Z., Mulvaney D. and Chouliaras, V. A Novel Genetic Algorithm Designed for Hardware Implementation, *International journal of Computational Intelligence*. vol. 3 (4). 2006.
9. Scott S. D., Samal A. and Seth, S. *A Hardware-Based Genetic Algorithms*, Technical Report UNL-CSE-97-001. University of Nebraska Lincoln. 1997.
10. De Jong, K.A. and Spears, W.M. Using Genetic Algorithms to Solve NP Complete Problems, *Proceedings of the Third International Conference on Genetic Algorithms*. J.D. Schaffer, Ed. June 1989. Morgan Kaufmann Publishers Inc. 1989. 124-132.

11. Nedjah, N. and Mourelle, L. M. An Efficient Problem-Independent Hardware Implementation of Genetic Algorithms, *Journal Neurocomputing, Dedicated Hardware Architectures for Intelligent Systems; Advances on Neural Networks for Speech and Audio Processing*. 2007, vol. 71 (1-3): 88-94.

12. Kanasugi A. and Tsukahara, A. A Processor for Genetic Algorithm Using Dynamically Reconfigurable Memory, *Journal of Convergence Information Technology*, 2007, vol. 2 (1): 4-15.

13. Scott S. D., Samal A. and Seth, S. HGA: A Hardware-Based Genetic Algorithm, *Proceedings of the ACM/SIGDA 3rd International Symposium in Field-Programmable Gate Array*, 1995, 53-59.

14. Schmitz, T., Hohmann, S.G., Meier, K., Schemmel, J. and Schürmann, F. Speeding up Hardware Evolution: A Coprocessor for Evolutionary Algorithms. *Proceedings of the 5th International Conference on Evolvable Systems lCES 2003,* 2003, 274-285.

15. Bland, I. M. and Megson, G. M. Implementing a Generic Systolic Array for Genetic Algorithms. *Proceedings of the 1st On-line Workshop on Soft Computing*, 1996, 891-905.

16. Megson, G. M. and Bland, I. M., Synthesis of a Systolic Array Genetic Algorithms. *Proceedings of the 12th. International Parallel Processing Sysmposium on International Parallel Processing Symposium*, 1998, 316-320.

17. Graham, P. and Nelson, B. Genetic Algorithms In Software and In Hardware – A Performance Analysis of Workstation and Custom Computing Machine Implementations, *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, 1996, 216-225.

18. Martin, P. A Hardware Implementation of a Genetic Programming System Using FPGA and Handel-C, *Journal of Genetic Programming and Evolvable Machines,* 2001, vol. 2 (4): 317-343.

19. Samarah, A. *Automatic Coverage Directed Test Generation Using a Cell-Based Genetic Algorithm*, Master Thesis, Department of Electrical and Computer Engineering, Concordia University, Montreal Québec, Canada. 2006.

20. Chambers, L. *The Practical Handbook* of *Genetic Algorithms, Volume 3: Complex Coding Systems*. CRC Press. 1998.

21. Thibault, D. *Hardware Structures for Genetic Algorithms*, Master Thesis, Faculty of the Department of Electrical Engineering, University of Massachusetts Lowell. 1996.

22. Joines, J.A. and Houck, C.R. On the Use of Non-Stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with GA's. *Proceedings of the IEEE Transactions on Evolutionary Computation*. 1994. vol. 7 (5): 445-455.

23. Soutar, C., Roberge, D., Stoianov, A., Gilroy, R. and Kumar, B. V. K. V. Biometric Encryption. *ICSA Guide to Cryptography*. R.K. Nichols, Ed. McGraw Hill. 1999.

24. Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin, Second Edition, 1994.

25. Schlüter M. and Gerdts, M. The Oracle Penalty Method, *Journal of Global Optimization*. 2010. vol. 47 (2): 293-325.

26. Juels A. and Wattenberg, M. A Fuzzy Commitment Scheme. *Proceedings of Sixth ACM Conference on Computer and Communications Security*. November 1999. Singapore: ACM Press. 1999. 28-36.

27. Jain, A. K., Bole R. and Pankanti, S. *Biometrics: Personal Identification in Networked Society*. Boston: Kluwer Academic Press. 1999.

28. Pankanti, S., Bolle R. and Jain, A.K. "Guest Editorial". *IEEE Computer (Special Issue on Biometrics)*. 2000. vol. 33(2): 46-49.

29. N. Goranin and A. Cenys, Evolutionary Algorithms Application Analysis in Biometric Systems, *Journal of Engineering Science and Technology Review 3 (1)*, 2010, 70-79.

30. Scheidat, T., Engel A. and Vielhauer, C. Parameter Optimization for Biometric Fingerprint Recognition using Genetic Algorithms. *Proceedings of ACM 2006 Multimedia & Security Workshop, Geneva, Schwitzerland*. 2006.

31. Jain, A. K., Prabhakar, S. and Ross, A. *Fingerprint Matching: Data Acquisition and Performance Evaluation*. MSU Technical Report TR99-14. 1999.

32. Scheidat, T., Vielhauer, C. and Dittmann, J. Distance-Level Fusion Strategies for Online Signature Verification. *Proceedings of the IEEE International*

Conference on Multimedia and Expo (ICME 2005). Amsterdam, Netherlands. 2005. 1294-1297.

33. Tajiri Y., Saito M. and Serikawa, S. Proposal of Optimization Method of Image Processing Parameter by Polytope Method. *International Symposium on Advanced Intelligent Systems*. Sokcho, Korea. 2007.
34. Khalil-Hani, M. and Eng, P.C. FPGA-based Embedded System Implementation of Finger Vein Biometrics. *2010 IEEE Symposium on Industrial Electronics & Applications (ISIEA 2010).* 2010. 700-705.
35. Spears, W.M., and De Jong, K. A. On the Vitues of Parameterized Uniform Crossover. *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufman. 1991. 230-236.
36. Martin, P. *An Analysis of Random Number Generators for a Hardware Implementation of Genetic Programming using FPGAs and Handel-C*. Technical Report CSM-358. Department of Computer Science University of Essex. 2002.
37. Altera Corporation. *Quartus II Version 7.2 Handbook. Volume 4: SOPC Builder*. Altera. 2010.
38. Vishnu Paramasivam. *An Embedded System for Networking Security Applying Cryptographic Acceleration in Field Programmable Gate Array Hardware*. Master Thesis. Universiti Teknologi Malaysia; 2009.
39. Micheli G.D. and Gupta, R.K. Hardware/Software Co-Design, *Proceedings of the IEEE*. 1997. vol. 85(3): 349-365.
40. Ernst, R. Codesign of Embedded Systems: Status and Trends. *IEEE Design & Test of Computers*. 1998. vol. 15 (2).
41. Dömer, R., Gajski, D. and Zhu. J. Specification and Design of Embedded Systems. *IT+TI Magazine*. 1998. vol. 3.
42. Axelsson, J. Hardware/Software Codesign for Automotive Applications, Challenges of the Architectural Level. *In ISORC, IEEE Computer Society*. 2001. 121.
43. Arató, P., Juhász, S., Mann, Z.Á., Orbán, A. and Papp, D. Hardware-Software Partitioning In Embedded System Design. *Proceedings of the 2003 IEEE International Symposium on Intelligent Signal Processing*. 2003. 197-202.

44. Lopez, M., Canto, E., Fons, M., Manuel, A. and Del Rio, J. Hardware Coprocesor Design for Fingerprint Image Enhancement. *Proceedings of the 49th IEEE International Midwest Symposium on Circuits and Systems MWCAS'06*. 2006. 520-524.

45. Gupta, R. *Embedded Processors Project Report for ICS 212*. 2000.

46. Hao, F., Anderson, R. and Daugman, J. Combining Crypto with Biometrics Effectively. *Proceedings of the IEEE Transactions on Computers*. 2006. vol 55(9): 1081-1088.

47. Harris, D.L., Oberman, S. F. and Horowitz, M. A. SRT Division Architectures and Implementations. *Proceedings of the 13th IEEE Symposium on Computer Arithmetic*. 1997. 18-25.

48. Vavouras, M., Papadimitriou, K., and Papaefstathiou, I. High-speed FPGA-based Implementations of a Genetic Algorithm. *Proceedings of the 9th International Conference on Systems, Architectures, Modelling and Simulation (SAMOS'09)*. Walid Najjar and Michael J. Schulte (Eds.). Piscataway, NJ, USA: IEEE Press. 2009. 9-16.

49. Jain, A. K. and Ross, A. *Introduction to Biometrics,* Handbook of Biometrics, A. K. Jain, P. J. Flynn and A. Ross (Eds.), Springer US. 2008. 1–22.

50. Tipton, H.F. and Krause, M. *Handbook of Information Security Management*. Auerbach Publications. CRC Press LLC. 1998.

51. Annemarie Nadort. *The Hand Vein Pattern Used as a Biometric Feature*. Master Thesis. Justitie Nederlands Forensisch Instituut, vrijie Universiteit Amsterdam. 2007.

52. Chopra S. and Marfurt, K. Gleaning Meaningful Information from Seismic Attributes. *First Break*. 2008. vol. 26(9): 43-53.

53. Ibáñez L., Schroeder W., Ng, L., Cates, J. and the Insight Software Consortium, *The ITK Software Guide*. version 2.4. 2005.

54. Wang, K., Gao, L., Pian, Z., Guo, L. and Wu, J. Edge Detection Combined Entropy Threshold and Self-Organizing Map (SOM). *Proceedings of the 4th international symposium on Neural Networks: Part II--Advances in Neural Networks (ISNN '07)*. Derong Liu, Shumin Fei, Zengguang Hou, Huaguang Zhang, and Changyin Sun (Eds.). Berlin, Heidelberg: Springer-Verlag. 2007. 931-937.

55. Shackleford, B., Snider, G., Carter, R. J., Okushi, E., Yasuda, M., Seo, K. and Yasuura, H., A High-Performance, Pipelined, FPGA-Based Genetic Algorithm Machine. *Journal of Genetic Programming and Evolvable Machine*. 2001. vol. 2(1): 33-60.

56. Meysenburg, M. M., Foster, J. A., Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M. H., Honavar, V., Jakiela, M. and Smith, R. E. Randomness and GA performance, revisited, *in Proceedings of the 4th Genetic and Evolutionary Computation Conferences*, 1999, vol. 1: 425-432.

57. Nahid Mahfuza Alam. *Implementation of Genetic Algorithms in FPGA-based Reconfigurable Computing Systems*. Master Thesis. Graduate School of Clemson University. 2009.

58. Knut, D. E. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Third Edition. Addison-Wesely. 1998.

59. Matsumoto M. and Nishimura, T. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Transactions on Modeling and Computer Simulation*. 1998. vol 8(1): 3-30.

60. Bedaux, J. *Mersenne Twister Pseudo-Random Number Generator in C++*. 2003.

61. Altera Corporation. *Avalon Interface Specification.* Altera. 2009.