ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 2, No.8 , March 2014
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

Multi-Objective Genetic Algorithms for computing Fast Fourier Transform for evolving Smart Sensors devices using Field Programmable gate arrays

Innocent Njini[1] Obeten O. Ekabua[2]
[1]Department of Computer Science, North-West University, Mafikeng Campus,
Private Bag X2046, Mmabatho 2735, South Africa
innocent.njini@gmail.com

[2]Department of Computer Science, North-West University, Mafikeng Campus,
Private Bag X2046, Mmabatho 2735, South Africa
obeten.ekabuao@nwu.ac.za

Abstract

This research explores the use of Fast Fourier Transform (FFT) in the design of smart sensors and various other portable telecommunication devices. Literature reveals that genetic algorithms play an increasingly important role in the computing FFT for the design and implementation of the foresaid devices. In this paper, a multi- objective genetic algorithm for computing Fast Fourier Transform for evolving smart sensor devices has been designed and simulated in software in a field programmable gateway array using Altera Quartus II 13.0. Programmable logic units, (Luts-representing genes in a chromosome) were designed using schematic /block diagrams and hardware description language and they were encoded into chromosomes and then optimized using field programmable gate arrays (FPGA) to achieve desired goals of portability and low energy consumption. The performance of our design was evaluated using Altera Quartus II Powerplay power analyser tool in terms of core static thermal power dissipation, core dynamic thermal power dissipation and input and output thermal power dissipation. The model was also evaluated in terms of functional correctness and portability. The results show significant improvements in all performance measures.

Key words : Genetic algorithm; Programmable logic unit (Luts); Fast Fourier Transform; Field programmable gate Array.

1. Introduction

An ad hoc wireless sensor network consist of a number of distributed micro electro mechanical devices (sensor nodes) deployed in some geographical region and they are capable of reorganizing themselves within the network. Each sensor node has wireless communication capabilities, usually in the form of radio signals and some level of intelligence for signal processing and networking data **[1].**

One of the most important constraints on sensor nodes is low power consumption requirement. Sensor nodes carry limited, generally irreplaceable power sources. Therefore, while traditional networks aim to achieve high quality of services (QoS) provisions, sensor network protocols must focus primarily on power conservation. They must have inbuilt trade-off mechanisms that give the end user the option of prolonging network lifetime at the cost of low power throughput or high transmission delay **[2].**

A sensor network design is influenced by many factors, which include fault tolerance, scalability, production costs, operating environment, sensor network topology, hardware constraints, transmission media and power consumption **[3].** These factors have been addressed by many researchers, but to my knowledge most of the researches have tended to concentrate on energy conservation in routing protocols, while this research looks achieving energy conservation through sensor collaboration and distributed processing. While less information is lost when communication is at a lower level (eg raw signals) being sent from source to the destination node (sink), this usually results in more bandwidth requirements and a lot of data redundancy, as duplicate information is sent. Thus, depending on the phenomenon under observation, the measurements collected by sensor nodes may need to be subjected to proper processing which might be performed either in a distributed manner by nodes, or centrally where information is collected **[4].**

Distributed processing, or collaborative signal and information processing where nodes in an ad hoc wireless sensor network collaborate to collect data and process it into useful information is a relatively new area of research. Processing data from more

sensors generally results in better performance but it also requires more communication resources (and thus energy). Energy efficient digital signal processors (DSP's) are becoming increasingly important with the growth of portable, wireless, battery-operated appliances such as wireless sensor networks, cellular phones, Personal Digital Assistants (PDAs) and laptops **[5].**

2. Genetic algorithms and FFT computing

Genetic algorithms are search procedure modeled on the Mechanism of Natural selection and evolution. They use the techniques found in nature such as reproduction, gene crossover and mutation to find optimal solutions to many real world problems. Genetic algorithms may be used to select the best signal measuring a phenomenon of interest so that only fewer signals are eventually transmitted from source to sink **[6].**

The concept of optimization, finding the extrema of a function that maps candidate 'solutions' to a scalar value of 'quality' is extremely general and useful concept that can be applied to innumerable problems in commerce, industry, and science. However, the majority of 'real' optimization problems, whatever their origin, comprise of more than one objective **[7]**. Multi-objective optimization takes seriously the fact that in most problems the different components that describe the quality of the solution cannot be lumped together into one representative overall measure, at least not easily and not before understanding of the possible 'tradeoff' available has been established **[7].**

The power of genetic algorithms comes from the fact that the algorithms converges to an optimal solution. Using Genetic algorithms, unlike other methods, it is possible to maintain infeasible and/or suboptimal solutions during the search process, which may be close to the global optimum while there are many local optima far away from the global optimum. They are resilient to being trapped in local optima **[8].** A further advantage of genetic algorithms is that they require no knowledge gradient information about the search space. Discontinuities in the response surface have little effect on the overall optimization performance. They perform very well for large- scale optimization problems. Further advantages are: Conceptual simplicity, Potential to use knowledge and hybridization, Parallelism, robust to dynamic changes, Capability for self-optimization and the ability to solve problems without known solutions **[9].**

2.1 Simulating Evolution

Genetic algorithms represent a special case of the more general class of evolutional computation algorithms (which also include methods such as evolutionary programming, and evolutionary strategies).

Evolutionary computing techniques, such as genetic algorithms, attempt to simulate biological process on the computer in order to solve difficult problems. Pioneered by John Holland , genetic algorithms have been employed by programmers for such diverse tasks as optimizing networks, Wireless sensor network deployment, calculating neural network weights, maximizing mathematical functions, scheduling resources more efficiently, minimizing costs in architectural designs while still meeting design constraints **[10].** Genetic algorithms apply when the elements are real discrete or complex valued. Thus genetic algorithms are suitable for digital signal processing and fast Fourier transform computations.

2.2 Genetic Operator

Signal encoding, representing the information to be optimized\searched is the most important step. In binary coding, which is the most popular approach, this entails representing a typical solution to be optimized in a binary string, referred to as a chromosome. Each chromosome has an associated objective function values called fitness function.

2.3 Fast Fourier Computing

Energy efficient digital signal processors (DSP's) are becoming increasingly important with the growth of portable, wireless, battery-operated appliances such as wireless sensor networks, cellular phones, Personal Digital Assistants (PDAs) and laptops

The Discrete Fourier Transform (DFT) and the Inverse Discrete Fourier Transform (IDFT) are computational tools that play a very important role in many digital signal processing applications, such as frequency analysis (spectrum analysis) of signals, power spectrum estimation, and linear filtering. The formula for transforming for a sequence {x(n) } of signals of length $\mathbf{L} \leq \mathbf{N}$ into a sequence of frequency of samples { $\boldsymbol{X(k)}$ } of length N is called the Discrete Fourier Transform (DFT) and its given by **[11]** :

$$\boldsymbol{X(k)} = \sum_{n=o}^{N-1} \boldsymbol{x(n)}\, \boldsymbol{e^{-j2\pi kn/N}} \quad , \qquad \textbf{(1)}$$

where n=0, 1,..., N-1.

The formula that allows us to recover the sequences of $x(n)$ from the frequency samples is called the Inverse Discrete Fourier Transform (IDFT) and it's given by:

$$x(n)=\frac{1}{N}\sum_{n=o}^{N-1} X(k)\, e^{j2\pi kn/N} \qquad (2)$$

where n = 0, 1,…, N-1 .

The importance of the DFT and IDFT in such practical applications is due to a large extent on the existence of computationally efficient algorithms, known collectively as Fast Fourier Transforms (FFT) algorithms for computing DFT and IDFT.

The computation of Discrete Fourier Transforms (DFT) is a significant component in many digital signal processing systems. Application of the fast Fourier transform (FFT) is increasingly gaining ground due to its ability to reduce the number of computations when computing discrete Fourier transforms. The wide spread use of FFT has led to a variety of implementations including custom FFT processors, digital signal processors, general purpose software and programmable hardware **[12].** With the advent of Silicon on Chip (Soc) technology, DSP algorithms such as FFT are increasingly becoming popular and are used in processor cores which are embedded within the Soc Platform . The increase in demand of high performance, low power applications like multicarrier systems, has lead to an increase in demand for FFT/IFFT cores, which provide high throughput while minimizing power consumption. It has been shown that the main source of power consumption in a typical CMOS circuit is due to switching power, P_{sw} by:

$$P_{sw}=\frac{1}{2}kC_{load}V^2_{dd}f \qquad (3)$$

where V_{dd} is the supply voltage , f is the clock frequency, C_{load} is load capacitance of the gate and k is switching activity factor, which is defined as the number of times the gate makes an active transition in a clock cycle **[13].**

Genetic algorithms play an important role in the design implementation of discrete time signal processing algorithms and systems. In **[14],** a genetic algorithm optimization for coefficient of FFT processor is reported. In **[15],** a multi-objective genetic algorithm for on-line adaptation a multi-carrier code deviation multiple access (MC –CDMA) receiver is presented. The approach uses a genetic algorithm to adapt the receiver, while dynamically optimizing the Fast Fourier Transform section of the receiver for both error value and power consumption. Figure 1 bellows shows the block diagram of the Multi-Carrier based Telephone Receiver. It consists of an FFT block that is used to demodulate the orthogonal frequency division multiple access (OFDMA) and a combiner block which equalizes signals and separates the coded users. The FFT processor is the most power consuming block, and its power consumption depends on the word length of data and the FFT coefficients. In **[16]** , a Multi-Objective Algorithm is used to find the optimal word length for input data and Fast Fourier Transforms coefficients while satisfying functionality constraints. The limitation of this approach is that the fitness function is based on one objective function derived from some error term.

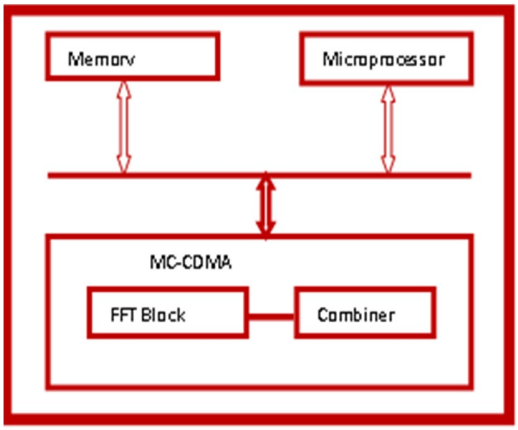

Figure 1: Block diagram for multi-carrier code deviation multiple access telecommunications receiver

Processors with larger word length will provide better performance in terms of accuracy. The accuracy of FFT processor is represented by Signal to Noise ratio (SNR), and a processor design with higher SNR value is desirable. On the other hand a high word length in FFT design will contribute to higher switching activities (SA). To resolve this problem, in **[17]** a single objective and a multi objective Genetic algorithm Approach is proposed to search for better SNR and SA value of the FFT processor.

3. Genetic algorithms in computing FFT for Smart Sensors

In **[18],** the idea of a Distributed digital signal Processor (DDSP) is considered, where a network is composed of multiple sensor nodes, each of which corresponds to a processor. The idea of divide- and-conquer approach is then applied, in which a DFT of size N, where is a composite number is reduced to the

computation of smaller DFTs from which the larger DFT is computed.

In **[19],** an Energy efficient algorithm for computing 1-D Fast Fourier Transform (FFT) over single and multi-hop wireless sensor network is proposed. The proposed algorithms reportedly reduces the number of transmissions, eliminates typical redundant computations in a distributed FFT algorithm and uniformly maps complex multiplications over all sensor nodes by introducing an extra bit – complement permutation stage after $(log_2 N)/2$ iterations.

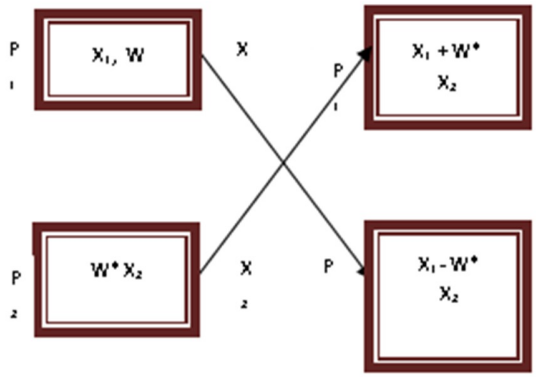

Figure 2: load balanced sensor network.

Figure 2 above shows two sensors in a traditional load balanced network.

An assumption is made that every sensor has a complex weight, w, required for the FFT for local computation. Between two consecutive stages, two sensors p1, and p2 have partial results, X1 and X2 available from the previous stage and they process information as shown in the diagram. The computations are as follows:

a) Sensor p_1 sends a copy X_1 to p_2

b) Sensor p_1 computes $X_1=X_1+w*X_2$

c) Sensor p_2 computes $X_2 = X_1-w*X_2$

It has been observed that at every stage each sensor is involved in transmitting and receiving one packet over the radio channel, and a complex multiplication and one addition and multiplication. Note that this leads to a load balanced distribution of work, but with redundancy of computation $w*X_2$ which is performed twice.

It has been shown in literature that transmission of a complex number over to an immediate neighbor consumes about 50 times more energy than the addition operation. Thus in order to reduce energy consumption, redundancy in complex multiplication should be avoided. There is also need to balance energy depletion among sensors.

In an attempt to avoid redundancy in computation, in **[20]** Canli etal proposed an approach that avoids redundancy at the expense of load balancing.

Figure3 bellow shows a diagrammatic representation of the approach.

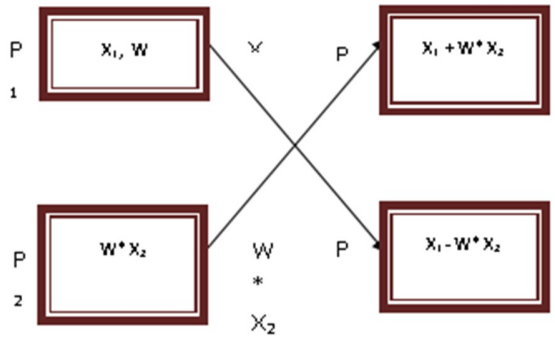

Figure3: Two Sensors without redundancy.

In this approach, p_1 and p_2 are computed as follows:

a) p_2 computes $w*X_2$

b) p_1 sends a copy of X_1 to p_2

c) p_2 sends a copy of $w*X_2$ to p_1

d) p_1 computes $X_1 = X1+ w*X_2$

e) p_2 computes $X_2 = X1-w*x_2$

In the above scenario, we have transmission of two packets and both sensors are involved in addition/subtraction but only one sensor is involved in complex multiplication. This approach removes the redundancy of complex multiplications compared to the convectional load balanced algorithms. The limitation of this approach is its lack of load balancing; hence some sensors will deplete their energy before others.

The load balancing problem is resolved by proposing an algorithm in which sensors are labeled p_1, p_2, …,p_n. Computation of data done as in one stage of the unbalanced algorithm, with even labeled sensors computing the complex multiplication and the odd labeled sensors using the results passed to them. In

the next stage, the roles are interchanged and the odd labeled sensors compute the complex multiplication while the even labeled use transmitted results of the computation. The approach helps to achieve both load balancing and reducing redundancy of complex multiplication **[21].** The limitation of this approach is the cost associated with switching as roles are interchanged at the end of each stage. This may lead to lager memory requirements.

4. low Power FFT Architecture

The evolution in semiconductor technology and the advert of SoC(System-on-Chip), has enabled prototyping of FFT algorithms as parameterizable cores which can be embedded within a SoC platform. The FFT processor can be classified into three main types: Pipelined FFT, Column FFT, and Fully parallel FFT. In applications that required high throughput and low power consumption, the Pipelined FFT processor is the preferred core. However, in real-time applications, where data is a sequential stream, the Pipelined architecture does not match the requirements of the FFT algorithms, as it requires temporal re-ordering of data **[22].** There is therefore need for a commutator, which is used to reorder the input data. One of the commonly used pipelined architectures is the Radix-4 Single path Delay commutator (R4SDC). This architecture is widely used due to its high utilization of multipliers, butterfly elements and memory blocks. While dedicated hardware can provide the highest processing performance, it suffers the drawback of lack of flexibility to changing processing requirements. While alterative systems based on software are flexible, they often suffer from insufficient processing capability.

Evolvable hardware is a new concept where the structure of hardware is designed to adapt to changes in task requirements or changes in the environment through its ability to reconfigure its own hardware structure online autonomously. In **[23],** a combination of genetic algorithms and software reconfigurable devices are reportedly used to design virtual reconfigurable hardware. The structure of reconfigurable devices is determined by downloading bit strings called architecture bits. The main drawback of this approach is its reliance on downloading and uploading of bit streams online **[24].** My proposed approach is a hybrid of the Radix-4 Single path Delay commutator (R4SDC) and Virtual Reconfigurable circuits implemented in Field programmable gateway arrays[**24][25].**

5. Multi-objective Genetic algorithms Design and Synthesis.

The multi-objective genetic algorithm we implement starts with a pool of chromosomes which might represent either feasible or non-feasible solutions to our optimization problem. In this implementation we started with feasible chromosomes and then moved towards optimality. Each chromosome is made up of several genes that define the chromosome length. Figure 4 bellow shows a typical chromosome and its constituent genes.

chromosome 1	inputs	Mux1	Lut1	Mux2	Lut2	----------	outputs
Gene codes	1101	1100	1010	1111	1111	xxxxx	

Figure 4: Chromosome with several genes (programmable logic units).

Each gene represents a programmable processing unit, generally referred to as a logic unit, which can be a multiplexer, or a combination of a number of primitive logic units that are connected together to perform a specific function.

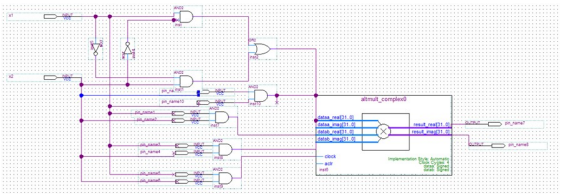

Figure 5: Sample programmable logic unit(Lut).

The Lut structures were incrementally built using Verilog VHDL and Quartus Schema/block diagrams interchangeably. The built-up of the Lut structures involved Compilation, Analysis and synthesis stages to verify the function correctness of the components. Figure 6 bellow shows compilation results for a sample Lut.

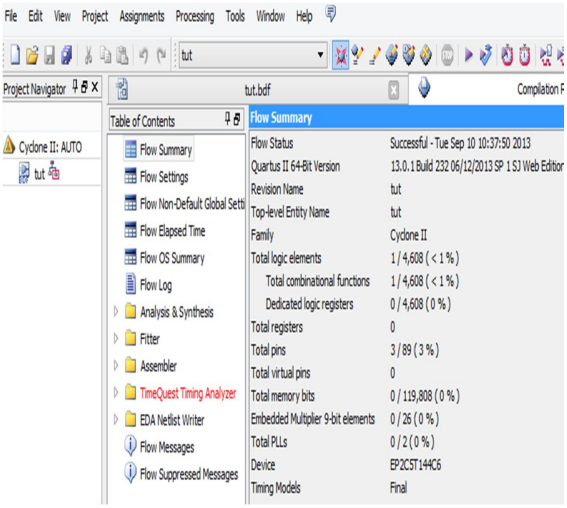

Figure 6: Compilation results for a sample Lut.

The Lut compilation above was performed using Quartus II 64-Bit version, device EP2C5T44C6. The compilation shows that there was no need for embedded multiplier elements, no need for dedicated registers and no need for virtual pins to process the Lut function. Only less than one percent of total logic elements were utilized using 3 pins. This compilation shows that the Lut structure under consideration is not resource wasteful and hence its inclusion within a reconfigurable circuit would energy conservative.

The gene codes represent the configuration of each logic unit. The processes of mutation (bit-inversion) are used to change gene codes, which results in reconfiguration of a logic unit such that the Lut behaves differently. This is achieved through opening and closing of circuits in practice. The process of mutation is used to crossover two parent chromosomes and generation a child which represents a new solution to the problem. The processes of mutation are carried out until the number of child chromosomes reaches a specified number, upon which the process is stop and selection takes place.

The algorithm we applied is outlined bellow:

1. Randomly select n chromosomes of Length N and fit them onto the FPGA.
2. Use the FPGA to compute the FFT equation 1, with N=16 and k = 0,1,2,…,15.
3. Evaluate fitness function

$$minimise\ f_n = \sum_1^N \mu\, V_d^2 \ + \rho S + \beta L$$

Where μ, β, and ρ are constants denoting relative importance of the respective variables.

V_d is voltage term relating to power consumption, S is slice space on FPGA relating to portability measure and L is the number of logical units used.

4. Perform crossover with random probability p.
5. Select new population of chromosomes and repeat steps 2 to 4.
6. Perform mutation with small probability ε.

6. Simulation Results

The power dissipation of FFT algorithm was determined using PowerPlay Analyser tool available in Quartus II systhesis tool. These tools perform power analysis on data obtained after synthesis using VCD file containing signal activities of FFT core. The functionality of individual components was verified by running test benches in ModelSim (Simulation tool for Mentographics Inc). The FFT processor algorithm as a complete system was functionally verified using system level test bench tools in Modelsim.

Figure 7 bellow show Powerplay Power analysis summary results for sample FPGA. The Powerplay Power Analysis tool enables us to estimate the power of a device accurately. Electrical energy supplied by external power sources is needed for proper externally and internally operation of FPGA devices. Designers need to understand the total power required from these supplies when implementing power supply solutions. Moreover, designers need to consider how much of that total power is actually dissipated within the device(referred to as “thermal power” or “dissipated power”) compared to the proportion of power that is dissipated outside the device, such as external output capacitive loads and balanced resistor termination networks.

Thermal power is the power that dissipates as heat from the FPGA. For a good device, this energy must be sufficiently lower to allow a cooling solution for the device. The total power consumed by a device is comprised of standby power, dynamic power and I/O power components. Standby power is from current in the device in standby mode. Core dynamic power is from internal switching within the device (changing and discharging capacitance on internal nodes). I/O power is from external switching (charging and

discharging external load capacitance connected to device pins), I/O drivers and external termination network **[26].**

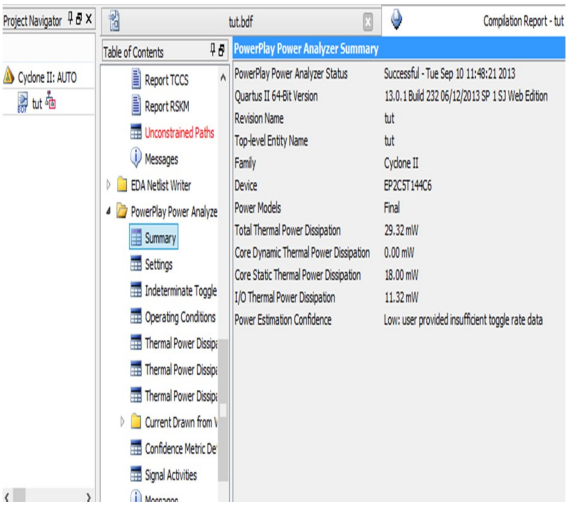

Figure 7: Sample Powerplay analysis summary for simple programmable logic units

The results of figure 7 show Total Thermal Power Dissipation of 29.32mW, Core Static Thermal Power Dissipation of 18mW, Input/output Thermal Power Dissipation of 11.32mW and zero Core Dynamic Thermal Power Dissipation.

An analysis of Total Thermal Power consumption with population evolution was carried out for our integrated reconfigurable circuit device. The results in figure 8 Show that the Total Thermal Power consumption generally decreased as the circuit evolved. This demonstrates the ability of our genetic algorithm to improving the performance of the circuit design under consideration.

Figure 9 shows an analysis of number of programmable logic units with population evolution for the integrated circuit design. The results show that our algorithms managed to significantly reduce the number of logic units require to perform the task at hand. The reduced number of logic units in turn leads to reduced energy consumption as reported above. Furthermore, this reduction in number of logic units in turn leads to reduced number of FPGA slices as indicated in figure 10. This reduction in FPGA is interpreted as a good measured of the portability of the physical device when implemented in hardware.

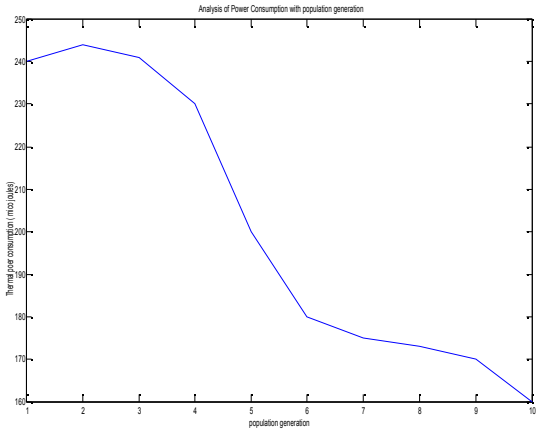

Figure 8: Thermal power consumption against population generation

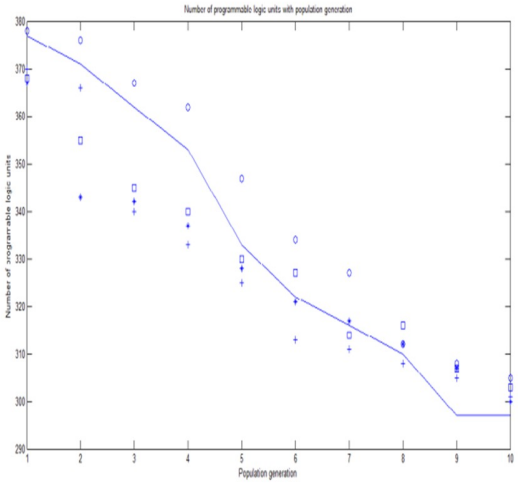

Figure 9: Number of programmable logic units with population generation

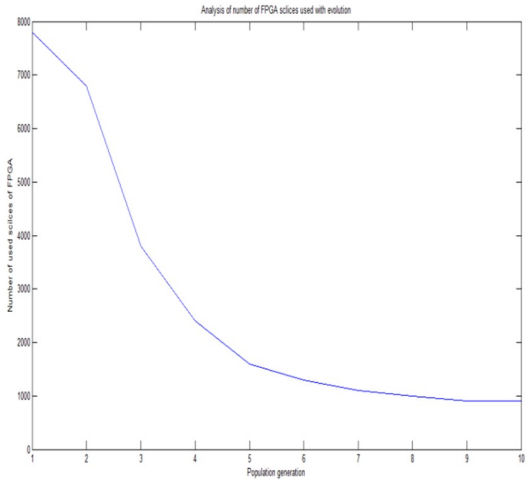

Figure 10: Number of FPGA slices with Evolution.

7. Conclusion.

In this paper, we present a Multi-objective Genetic algorithm for evolving smart Sensor devices using Field Programmable gate arrays. The Multi-objective genetic algorithm presented was used to evolve reconfigurable circuit designs in FPGA. Experimental results reveal that our genetic algorithms improved the design of the desired reconfigurable circuits in terms of energy consumption and device portability in a few generations.

Reference

[1] Akyildiz, Ian F., Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. "Wireless sensor networks: a survey." *Computer networks* 38, no. 4 (2002): 393-422.

[2] Basile, Claudio, Marc-Olivier Killijian, and David Powell. *A survey of dependability issues in mobile wireless networks*. Technical report, LAAS CNRS Toulouse, France, 2003.

[3] Akkaya, Kemal, and Mohamed Younis. "A survey on routing protocols for wireless sensor networks." *Ad hoc networks* 3, no. 3 (2005): 325-349.

[4] Arora, Anish, Prabal Dutta, Sandip Bapat, Vinod Kulathumani, Hongwei Zhang, Vinayak Naik, Vineet Mittal et al. "A line in the sand: a wireless sensor network for target detection, classification, and tracking." *Computer Networks* 46, no. 5 (2004): 605-634.

[5] Chong, Chee-Yee, and Srikanta P. Kumar. "Sensor networks: evolution, opportunities, and challenges." *Proceedings of the IEEE* 91, no. 8 (2003): 1247-1256.

[6] Michalewicz, Zbigniew. *Genetic algorithms+ data structures= evolution programs*. springer, 1996.

[7] Knowles, Joshua, and David Corne. "Memetic algorithms for multiobjective optimization: issues, methods and prospects." In *Recent advances in memetic algorithms*, pp. 313-352. Springer Berlin Heidelberg, 2005.

[8] Gen, Mitsuo, and Runwei Cheng. *Genetic algorithms and engineering optimization*. Vol. 7. John Wiley & Sons, 2000.

[9] Fonseca, Carlos Manuel Mira. *Multiobjective genetic algorithms with application to control engineering problems*. University of Sheffield, 1995.

[10] Bäck, Thomas, David B. Fogel, and Zbigniew Michalewicz, eds. *Evolutionary computation 2: advanced algorithms and operators*. Vol. 2. CRC Press, 2000.

[11] Ahmet T. Erdogan N Sulaiman, "A Multi-objective Genetic Algorithm for On-chip Real-time Adaptation of a," *Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 1-2, 2006.

[12] Richards, Mark A., and Gary A. Shaw. "Chips, architectures and algorithms: Reflections on the exponential growth of digital signal processing capability. "*Unpublished manuscript, Jan* 28 (2004).

[13] Malamas, Elias N., Euripides GM Petrakis, Michalis Zervakis, Laurent Petit, and Jean-Didier Legat. "A survey on industrial vision

systems, applications and tools." *Image and vision computing* 21, no. 2 (2003): 171-188.

[14] Hong, Pang Jia, and Nasri Sulaiman. "Genetic algorithm optimization for coefficient of FFT processor." *Australian Journal of Basic and Applied Sciences*4, no. 9 (2010): 4184-4192.

[15] Sulaiman, Nasri, and Ahmet T. Erdogan. "A multi-objective genetic algorithm for on-chip real-time adaptation of a multi-carrier based telecommunications receiver." In *Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on*, pp. 424-427. IEEE, 2006.

[16] Sulaiman, Nasri, and Tughrul Arslan. "A multi-objective genetic algorithm for on-chip real-time optimisation of word length and power consumption in a pipelined FFT processor targeting a MC-CDMA receiver." In *Evolvable Hardware, 2005. Proceedings. 2005 NASA/DoD Conference on*, pp. 154-159. IEEE, 2005.

[17] PALUMBO, Francesca. "Communication-Centric Approach to Multi-Processors System on Chip Design: Interconnection Networks Design and Evaluation." (2010).

[18] Chiasserini, Carla F. "On the concept of distributed digital signal processing in wireless sensor networks." In *MILCOM 2002. Proceedings*, vol. 1, pp. 260-264. IEEE, 2002.

[19] Yu, Yang, and Viktor K. Prasanna. "Energy-balanced task allocation for collaborative processing in wireless sensor networks." *Mobile Networks and Applications* 10, no. 1-2 (2005): 115-131.

[20] Canli, Turkmen, Ajay K. Gupta, and Ashfaq A. Khokhar. "Power Efficient Algorithms for Computing Fast Fourier Transform over Wireless Sensor Networks." In *AICCSA*, pp. 549-556. 2006.

[21] Ienne, Paolo, and Rainer Leupers. *Customizable embedded processors: design technologies and applications*. Access Online via Elsevier, 2006.

[22] PALUMBO, Francesca. "Communication-Centric Approach to Multi-Processors System on Chip Design: Interconnection Networks Design and Evaluation." (2010).

[23] Wang, Jin, Qiao Song Chen, and Chong Ho Lee. "Design and implementation of a virtual reconfigurable architecture for different applications of intrinsic evolvable hardware." *Computers & Digital Techniques, IET* 2, no. 5 (2008): 386-400.

[24] He, Shousheng, and Mats Torkelson. "Design and implementation of a 1024-point pipeline FFT processor." In *Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998*, pp. 131-134. IEEE, 1998.

[25] Murphy, Ciaron. "Virtual hardware using dynamic reconfigurable field programmable gate arrays." In *Engineering Development Centre, Liverpool John Moores University, UK, GERI Annual Research Symposium*. 2005.

[26] Pedram, Massoud. "Power minimization in IC design: principles and applications." *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 1, no. 1 (1996): 3-56.