International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-2 Issue-5, November 2013

Signal Processing Using Wavelet Transform On Sensor Nodes For Low Power Consumption

J. Hema shubraja, G. Vijayalakshmy

Abstract: ***Power consumption for different sensor interface utilized for wireless sensor is investigated. In that most sensor nodes used MCU to implement the main functionality in software. In other words such a unit consumes great deal of power for unused circuits. We propose the use of wavelet concepts to reduce the power consumption of small sensor node that has various sensors and wireless communication facilities, that were the result of an adaptive function specialization mechanism. Traditional sensor nodes must have had a powerful and multi functional Micro-Controller Unit (MCU) to satisfy the requirements for processing any kinds of application. However, most of these systems only use a part of the functions provided by an MCU. In other words, such a unit often consumes a great deal of power for unused circuits. To avoid this situation, we propose the use of wavelet transform processing technique instead of an MCU because this array dynamically changes the circuit to the optimal one that is just used for the calculation required by an application. Moreover, we implemented a prototype system to do a preliminary evaluation of our proposed mechanism. Here processing time and power consumption using Discrete wavelet transform (DWT) for testing the processing function on sensor nodes is proposed. Here the performance of proposed wavelet technique with the existing FFT model is compared and analysed. The performance in terms of noise and power are analysed. The experimental result shows that proposed mechanism reduces enough power of its sensor nodes to prolong the lifetime of nodes without decreasing the processing time.***

Index Terms: Wavelet, Images, FFT, Power

I. INTRODUCTION

Recently, many wireless sensor nodes have been proposed. Among them are MOTE [2] and Smart-Its [3]. Each node uses various types of sensors such as those for acceleration, geomagnetic, and temperature. Further, powerful middleware are provided for developers, such as an operating and a, database system, and a programming environment, which are useful in developing a variety of ubiquitous applications. One of the most significant issues in developing sensor nodes is to save the power consumption without decreasing functionalities. This is because its lifetime has a strong affect on the reliability of its service.

Revised Manuscript Received on 30 November 2013.
* Correspondence Author
J. Hema Shubraja, Electronics and Communication Engineering,, Kamban Engineering College, India.
G.Vijayalakshmy, Electronics and Communication Engineering, Perunthalaivar Kamarajar Institute of Engineering and Technology, India.

Downsizing a device is also an important issue in designing hardware, but there has been a trade-off between downsizing and power saving. To do both at the same time, the hardware has needed to be optimized to a specific application. In order to optimize hardware in existing architecture, however, a developer must carefully design it, and this is often a costly process. To reduce the cost of design, we propose the use of a new architecture.

Our basic idea is based on reconfigurable hardware architecture with a mechanism to adapt functions depending on the requirement of an application. We use FPGA based hardware instead of MCU, where FPGA is used for signal processing. Our FPGA-based processor can dynamically change the circuit to process functions required by an application. For example, when an application requires the spectrum analysis of sensor data, the system reconfigures the circuit to process the Discrete Fourier Transform (DFT). In another case, when an application requires only a simple analysis, it dynamically configures itself to process a Finite Impulse Response filter (FIR), which consumes less power than the DFT does. To develop such a sensor node, a developer using a traditional MCU-based device must make a sensor board that is independent from other devices; however, a developer using our FPGA-based device can dynamically change the circuits on the sensor node with software and thus not need to design hardware for each device. Hence this method can reduce the cost of developing various types of sensor nodes. A prototype device as an implementation with anIGLOO FPGA device produced by Actel Corporation is shown. Furthermore comparison of architecture with the prototype implementation with a traditional MCU-based device is evaluated. Previously they had used FFT for testing a function on sensor nodes [1]. In this work, discrete wavelet Transform circuit (DWT) is used for testing a function processed on the sensor nodes. In this work, power consumption (mW) and processing time (msec) were measured to compare the performance of each method. The results show that our architecture has an advantage in most cases. The rest of this paper is organized as follows. First, related works is mentioned in Section 2. Proposed approach and the core idea of the dynamic reconfigurable hardware architecture with the function specialization mechanism in given in Section 3. A prototype system and an implementation based on our architecture is described in Section 4. The evaluation results in our experiment are shown in Section 5. Finally, conclusion of the paper in Section 6.

II. RELATED WORKS

In this section, we mention related works on reconfigurable hardware architecture for developing small sensor devices.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

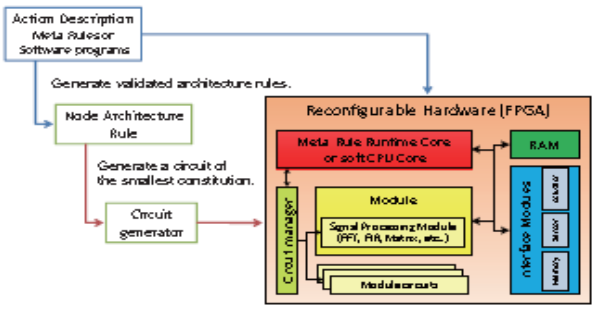

Fig. 1. Goal of Hardware and Software Generator

Bellis, et al. proposed a function specialization that uses an FPGA device on a sensor node at a module to control the RF device for efficient routing on sensor network [4]. In these studies, using an FPGA enables us to downsize and process at a high-speed at the same time with an FPGA-based function specialization mechanism. Most previous FPGA-based architecture brings us both the reduction of the device size and high-speed signal processing on a small sensor node. Moreover, in these architectures, the circuits cannot be dynamically changed to meet the requirements of applications because previous works focus on the case that a sensor device is only used in a specified application. In other words, a sensor node does not have to change functions dynamically because a developer of the application system fixes the required functions before developing the sensor node. Moreover a ubiquitous computing environment, many applications often require different functions to each sensor nodes to provide intelligent and convenient services to users. Because in this case the sensor node must calculate sensor data in a different way; the sensor needs to be dynamically reconfigured so it meets each requirement. Most of the previous sensor nodes do not support such dynamic reconfigurable mechanisms. On the other hand, there are some reconfigurable devices [5] [6] [7] [8], which are aimed at specializing the function on the basis of both the requirements of the application systems and the place where a sensor node is put. These approaches are similar to ours. While works focused on quite primitive calculation operations, such as summation, and division, here in this work more powerful calculation functions such as DWT is implemented. Thus, contribution in this paper is summarized by the following two points. First, dynamic reconfigurable hardware architecture with functions used in practical signal processing is implemented. Second, implementation of a prototype system using an FPGA device, and further, the advantages over traditional mechanisms with an evaluation of the performance evaluation of our architecture is performed.

III. PROPOSED ARCHITECTURE

In this section, we describe the architecture of our system. First, we describe the overall goals of our approach. The target of our approach is to reduce both the processing time and the power consumption of sensor nodes at the same time. For this purpose, we introduced a mechanism to configure the hardware dynamically to minimize the power consumption by specializing functions. To reconfigure the hardware, we used an FPGA instead of the MCU, which is a traditional device to process data on a sensor node. Our mechanism dynamically replaces circuits on the FPGA the set of functions required by application systems. That is, the FPGA only has just the minimum set of necessary circuits. By using this mechanism, a sensor node provides suitable a function set required by applications with the minimum power consumption without increasing the processing time. In the rest of this section, we describe how our goal can be achieved with our proposing mechanism. Overview of our mechanism is shown in Fig. 1. The key device of our mechanism is the FPGA that can dynamically change circuits. The controller has also a significant function, switching on the circuits on the FPGA. Our full implementation will use just an FPGA device as the controller but, to simplify the implementation, our prototype an MCU as the controller. The controller chooses a circuit set required by applications systems on the basis of previously described rules. A developer of sensor nodes describes available functions using the FPGA for application developers. When an application system needs to change the circuit sets, the controller dynamically replaces a set of circuits on the FPGA.

1) Hardware Architecture: Some of the traditional sensor nodes have a function that can change the software on altering demands of application [9]. However, the power consumption of the MCU changes little by changing the software in the main. In fact, although the traditional method cannot be used to reduce the power consumption but functions can be specialized it. It is necessary to replace circuits to reduce the power consumption. We can hold the power consumption to a minimum along the sensor nodes have that circuit to provide the function for required calculation. However, the hardware has a fixed architecture such an MCU, and it is impossible that the MCU cannot replace the circuits dynamically. Both the FPGA and the mechanism of dynamically replacing the circuit replacement can process faster and lower power consumption than a versatile CPU with dynamical replacement of software does. For these reasons, we designed sensor nodes with FPGA.

2) Controller: The controller, which is a significant module, has significant three functions: first, generating a set of circuits required by application systems using rules described by the circuit developers; second, ordering the replacement of the circuit set to the FPGA-based processor; and third, transferring data between an application system and the processor. Developers provide a set of circuits as a module and, moreover, describe the function of a module as a rule. The module includes the set of circuits, which provides a function to process simple summation, average, DFT, FIR, and so on.

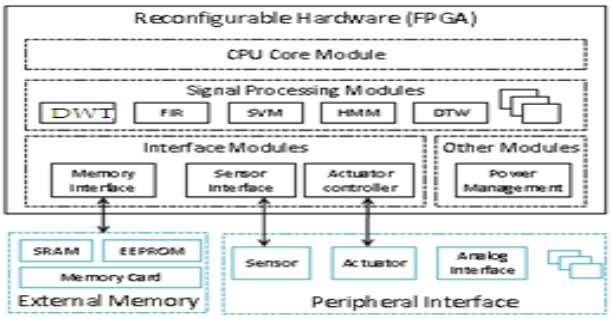

Fig. 2. Complete Design of Proposing Architecture

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Each module has the format both of input data and output data. The controller converts the format of data received from the application into a suitable format used in the module. Similarly, the controller returns the output data from the module with the format required by the application system. The mechanism allows many developers to make modules independently from each other, and they can also improve the module independently. We can add new functions to the sensor node by developing a module to provide the new function.

3) Software Layer: The reconfigurable hardware can calculate sensor data with dynamic changeable logical circuits. However, there is a trade-off when we design software on MCU or constitute a logical circuit because of scale of logical circuit have limit. The FPGA cannot process any tasks. It also does not well on sequential control. Note that both can be easily processed on an MCU. Therefore, we propose using a way to process complex calculations at a FPGA and others on software on an MCU. The complete version of the proposing system is shown in Fig. 2. In this version, both the circuits on the FPGA and the software on the MCU are automatically built using both a library and rules written by developers. This library includes logical circuit modules for the FPGA and software modules for the MCU, both of which have the same function. The controller automatically decides on the basis of the rule which modules must be used in the process at runtime.

IV. PROTOTYPE

In the previous section, we showed the complete version of our architecture which achieves our goal, but the essential idea of our proposed architecture is made by using the reconfigurable hardware architecture. Therefore, we also have a simple version for the current implementation in order to evaluate the power consumed when reconfigurable hardware is used. Here, we explain the prototype system used in our experiment. We extracted essential two mechanisms from our proposed architecture; that is the FPGA-based reconfigurable node that

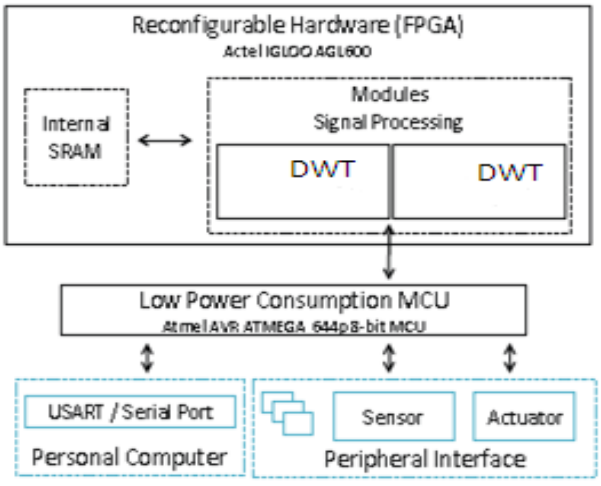

Fig. 3. Partial Design for Prototype System

can dynamically turn on/off the power provided to a circuit on the FPGA. In other words, this device cannot dynamically change the circuit but using the device can limit it, and consumes the power. This prototype is quite simple as it has only the core mechanism of our proposed architecture. In this work, both the power consumption and processing time with this prototype is measured. If the proposed prototype reduces the power consumption without decreasing the processing time, then our complete version also has the capability to reduce the power consumption because each system has the same mechanism used to turn on/off the circuit on a sensor node. To compare our mechanism with the traditional fixed hardware architecture, we also implemented an MCU-based device. An overview of the prototype is shown in Fig. 3. Here, we explain the design of our prototype using an FPGA and an MCU. The purpose of implementing a prototype is to compare performance of our proposed architecture with the traditional method in a real situation. For comparing austerely, we combined the greater part of our architecture with the proposed FPGA system and the MCU based one. In particular, we used the independent module as a controller in the MCU. The module has the function to make both composed a simple MCU processor and composed am MCU with a FPGA processor to run a task. The module can also be used to measure time. We got the processing time and the pure power consumption in processing by combined driving and time measuring. The prototype does not make use of sensor devices. We used random numbers or fixed number as calculating data as a substitute for sensor data. Our prototype system has the following two features. First, only DWT is implemented as a logical circuit. Second, it supports a 256 and 512 byte and also any type of byte format by writing code to the MCU in advance.

A Prototype using Wavelet Transform

Wavelet transformation (WT) is a tool that is widely used in nearly all applications nowadays. J. Morlet in the late 1970s introduced the concept that was the result of a drawback of Fourier transformation (FT). In FT, the information from the processed signal could only be extracted either in the time domain or in the frequency domain but not both at the same time. WT, on the other hand, will cut the signal into small pieces in the same size and will analyze this signal mutually. Wavelets are a powerful tool for the representation and analysis of ECG signal too. This is because wavelet has finite duration as compared to Fourier methods based on sinusoids of infinite duration. Hence wavelet transform is used to reduce the noise levels than other transforms (FFT). Since it is able to reduce noise the efficiency is increased and power consumption is reduced by its simplicity. Wavelet Transform involves the decomposition of signal into various components. They provide both time and frequency view. Unlike Fourier transform, they are very efficient for non-stationary signal. The Fourier Transform is a widely used tool for many scientific purposes but it is well suited for stationary signals. Gabor introduced a local Fourier analysis. He used the concept of a sliding window. This method, however, gives results when the coherence time is independent of frequency. Morlet introduced Wavelet Transform to have a coherence time proportional to the period. In Wavelet Transform, a fully scalable modulated window is used which solves the signal-cutting problem. The window is shifted along the signal. Spectrum is calculated for every position. This process is repeated by varying the length of the window. So we have a collection of representations, hence the name multi-resolution analysis.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

1. Continuous Wavelet Transform: Wavelet transforms are applied to decompose the signal into a set of coefficients that describe the signal frequency content at given times. The continuous wavelet transform of the signal, x(t), is defined as [3]:

$$F(a,b) = 1/\sqrt{a}\int_{-\infty}^{+\infty} x(t) * \Psi\left(\frac{t-b}{a}\right) dt \qquad (1)$$

Here Ψ(t) is the analyzing wavelet function a is the dilation parameter and b is the location parameter of the wavelet Actually the wavelets are generated from a single basic wavelet Ψ(t), the so called mother wavelet, by scaling and translation.

$$\Psi\, s,\tau\,(t) = \frac{1}{\sqrt{s}}\,\Psi(\frac{t-\tau}{s}) \qquad (2)$$

Here τ is the scaling factor and √ (1/s) is for normalization across the different scales. Due to the scaling and translation, Wavelet Transform is localized in both time and frequency. Several Mother Wavelets like Mexican-hat and Morlet have been used in signal analysis. The mother wavelet has a lot of significance for the efficiency of the process. In this paper we have used Haar Wavelet as the mother wavelet. We have gone for Haar wavelet because the oscillatory nature of other mother wavelets results in several ridges for each ECG component, while only one pair of ridges is generated via the Haar wavelet due to its configuration. Fig. 8 shows the Haar wavelet. The Haar wavelet's mother wavelet function Ψ(t) as shown in Fig.3.b can be described as

Ψ(t) = 1 $0 \le t < 1/2$
= -1 $1/2 \le t < 1$
= 0 otherwise

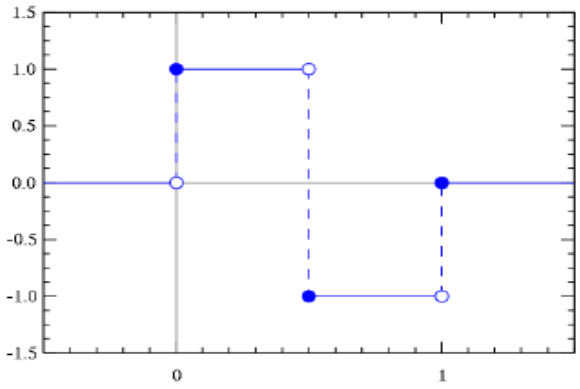

Fig. 3.b Haar mother wavelet

The mother wavelet used here is the Haar Mother wavelet. It is preferred here because the oscillatory nature of other mother wavelets results in several ridges for each signal component, while only one pair of ridges is generated via the Haar wavelet due to its configuration. The second step involves the use of a Threshold based detector.
Positive maximum peaks larger than a threshold are selected. The main threshold is chosen as a fraction of root mean square of the signal. We have chosen this to be around two times the root mean square of the signal after carrying out a series of experiments. The CWT method is found to have a good sensitivity. The average detection error rate is very low making this method highly lucrative. Moreover this method is much more evolved than others. The best feature of this method is that it is suitable for non-stationary signals.

V.EXPERIMENTAL RESULTS

Simulation results were computed for the power consumption and the computing time of reconfigurable hardware for analysing sensor data. Comparison results of experiments between FFT and DWT are shown. In particular, evaluated power consumption and processing time for processing the DWT in these systems were shown.

A.EXPERIMENT 1

Comparison of FFT and DWT coefficients

Comparison results between FFT and DWT coefficients were performed to analyse the coefficients of both FFT and DWT. Analysing FFT coefficients and DWT coefficients, we can prove that Coefficients of discrete wavelet transform are uniformly distributed leading to efficient spectrum. Since the DWT coefficients are uniform, noise can be effectively removed from the signal during signal transmission or in other words it is very difficult for the noise to be added with the original signal, because no irregular space between the wavelet coefficients as in FFT. Time delay for the signal transmission is also reduced consuming very less power in DWT. So we had proved along with power spectral density spectrum for DWT and FFT through simulation.

Fig. 4 shows the spectrum of FFT coefficients. Here it is clearly shown that the Fourier coefficients are irregularly spaced leads to ununiformed power distribution.

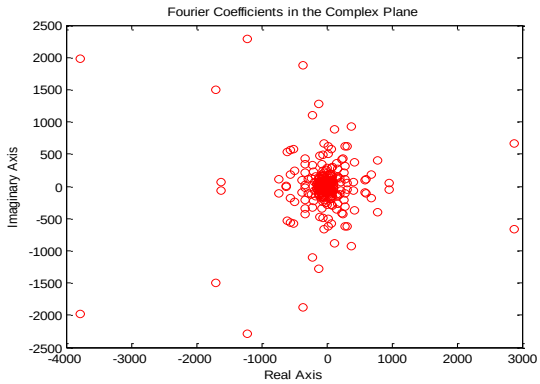

Fig. 4 FFT coefficients in complex plane

Fig. 5 shows the DWT coefficients which shows that the coefficients are uniformly spaced. So noise and the power can be reduced from analysing efficiency of the spectrum because of the uniformity of the wavelet packets. Next experiment is the continuation of previous experiment to prove noise is more in FFT than in DWT.

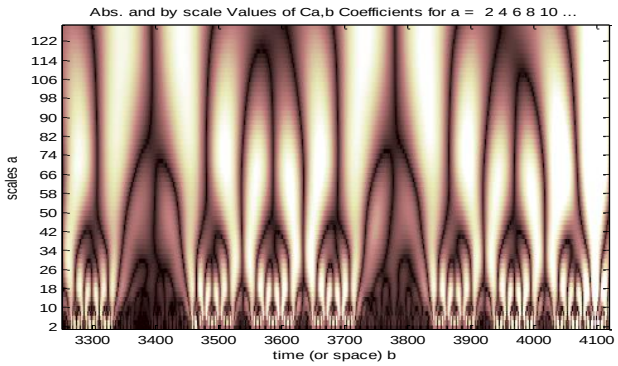

Fig. 5 DWT coefficients

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

B. EXPERIMENT 2

Comparison of noise signals in FFT and DWT

Simulation 2 is performed to analyse noise spectrum between FFT and DWT. Fig. 6 shows noisy signal in FFT plotted between frequency and magnitude of the signal. Here the amplitudes are not exactly at 0.7 and 1 is because of the noise. Several executions of this code (including recomputation of y) will produce different approximations to 0.7 and 1. The other reason is that you have a finite length signal. Increasing L from 1000 to 10000 in the example above will produce much better approximations on average. So noise is more in FFT.

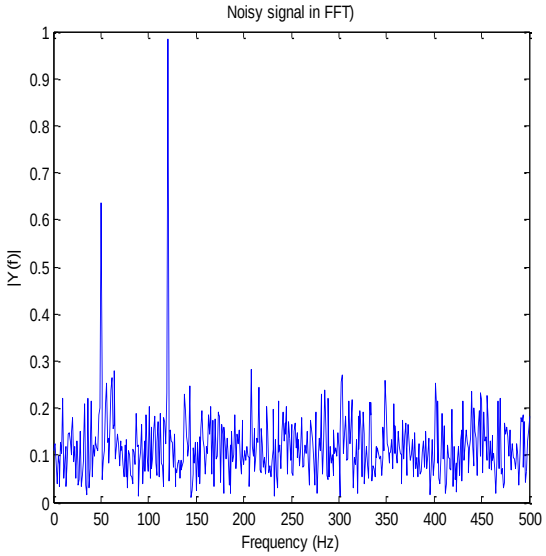

Fig.6 Noisy signal in FFT

Fig. 7 shows the denoised signal using wavelet packets. This experiment result clearly shows that noise can be effectively removed using DWT coefficients. Noise spectrum is highly reduced here when comparing to FFT.

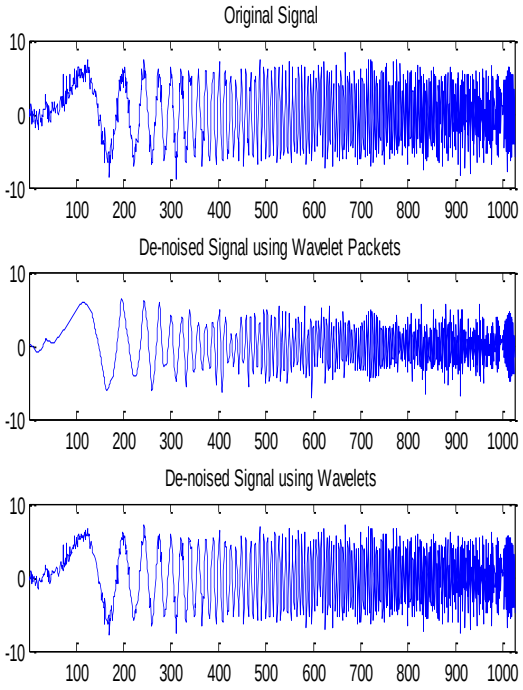

Fig. 7 Denoised signal using wavelet packets

Here the simulation result is plotted between noise and the amplitude. The retrieved signal and original signal are moreover equal. Here sudden amplitude change is not there as in FFt.

C. EXPERIMENT 3

Multi-signal processing with less processing time and reduced power consumption

Experiment 3 is performed to analyse multi-signal processing in DWT with reduced power and processing time. Since objective of the work is to prove power consumption is reduced. Fig.8 shows multi-signal processing in DWT. Here even for the multi-signals the noise is reduced highly where FFT is cannot be used for multi-valued signal transmission.

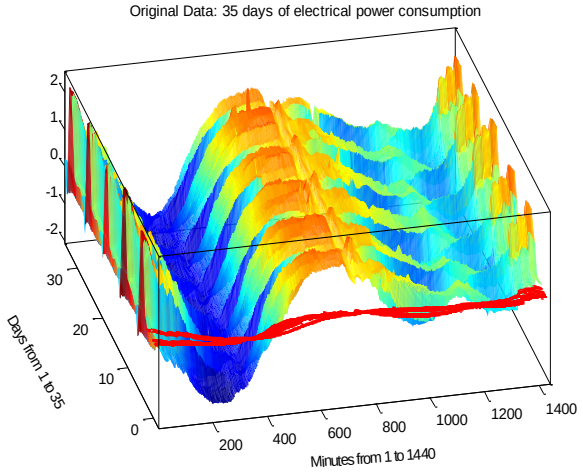

Fig. 8 Mutisignal processing in DWT

Processing time is also computed for signal processing nearly very less in nano seconds. It is also shown by following experiment by calculating retained energy and number of zeros using DWT calculations.

D. EXPERIMENT 4

Retained energy using DWT

Experiment 4 is performed to show the retained energy using DWT. Even after the original image is compressed, image can be easily retrieved with retaining original energy distribution using DWT. The retained energy is 98.65%. This shows DWT can consume less power leading to a less processing time. Fig. 9 shows original image and fig. 10 shows after performing DWT.

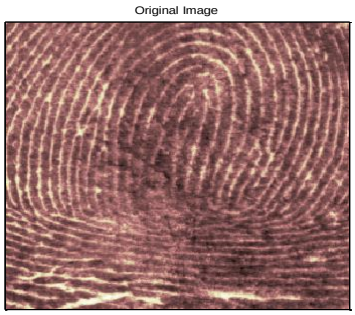

Fig. 9 Original image

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Fig. 10 After performing DWT

FFT can reduce the power consumption to 96.72 % and to 96.97 %. It is shown in previous research paper [1]. In DWT power consumption up to 98.65%.

E. EXPERIMENT 5

Power spectral density comparison between FFT and DWT

Experiment 5 shows the results for power spectral density for FFT and DWT. Here power spectral density of FFT and DWT is analysed because from power spectrum we can find out 3 parameters a. Power / Energy consumption b. Noise spectrum c. Efficiency of the spectrum. Fig. 11 shows the power spectral density of FFT. This clearly pictures that noise is larger so power consumed is also larger than DWT. Fig. 12 and 13 shows power spectral density of DWT

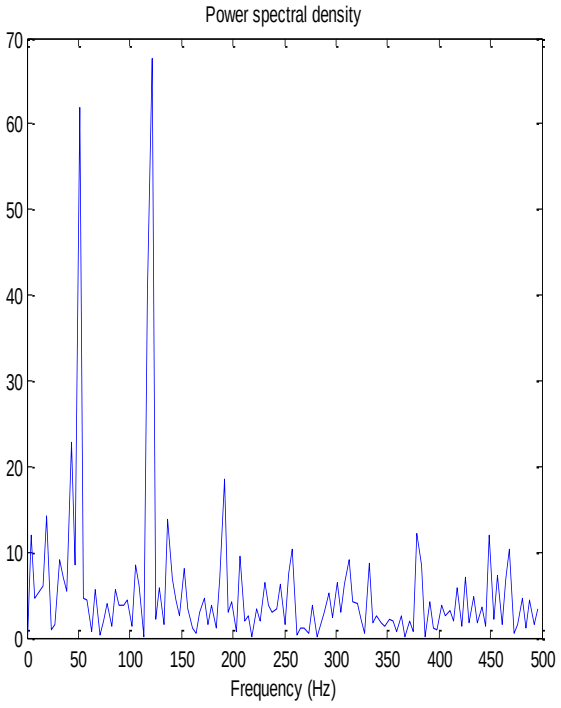

Fig. 11 Power spectral density of FFT

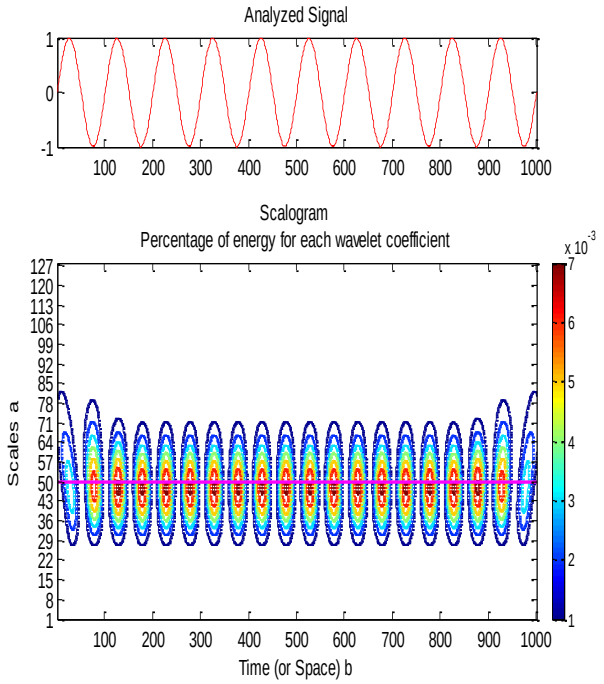

Fig.12 Power spectral density spectrum of DWT

In power spectral density plot of DWT percentage of energy for each wavelet coefficient is shown uniform and more levels of amplitudes can be plotted. Different plots have been plotted for power spectral density in order to prove power consumption is effectively reduced for DWT. The spectrum is plotted between Time and amplitude.

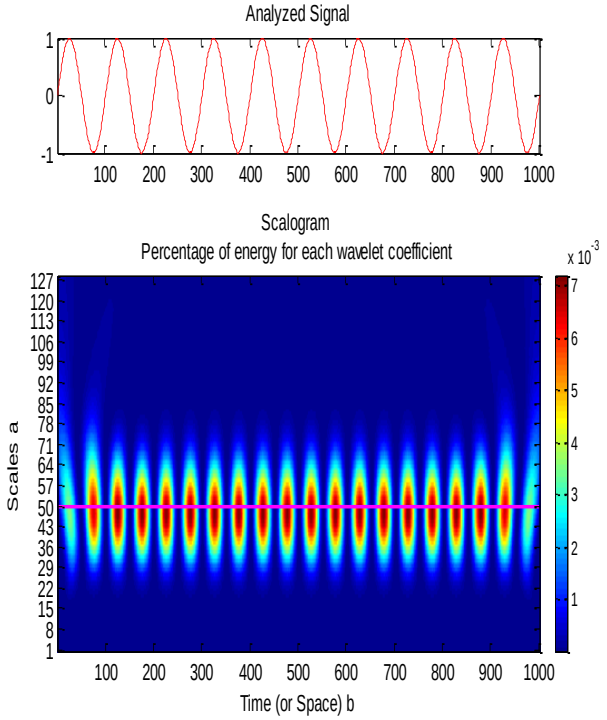

Fig.13 Power spectral density spectrum of DWT

In this experiment, we compared the amount of power consumption that depends on sensor data analysis in the active and the sleep mode of MCU and FPGA.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Therefore, we catch on the method of the proposed system that could reduce the power consumption to 98.65%. As a result, we were able to reduce the power consumption for calculations.

VI.CONCLUSION

The reconfigurable hardware architecture for a small sensor node is proposed. In general, existing sensor nodes, consume more power consumption for transmission wireless devices is much more than those used for calculation processing. This is because, in most traditional sensor node architectures, all the necessary sensor data must sent to a server. One way to reduce the amount of data sent to the server from a node is to compress and analyze them with signal processing functions on the node. Our method possibly enables nodes to process such calculation with both low power consumption and in a short processing time. **.** Performance results shows that in the proposed model noise is reduced, efficient power, and multi prshocessing are good in DWT, with efficient energy in the power spectral density than that of in FFT. The experimental result shows that proposed mechanism reduces enough power of its sensor nodes to prolong the lifetime of nodes without decreasing the processing time. The scope of our architecture to include the RF part and to thus reduce the power consumption of the sensor node even more is extended. In the future, we will implement several other calculation circuits such as the Support Vector Machine (SVM), the Hidden Markov Model (HMM), and the Dynamic Time Warping (DTW), and thus make analyzing sensor data more powerful. Moreover, we will archive other additional experiments on our system in realistic environments.

REFERENCES

1. Satoshi tanaka and Naotaka Fujita "Reconfigurable hardware architecture for saving power consumption on a sensor node" *Proceedings on international conference on networks*, 2008.
2. J. M. Kahn and R. H. Katz and K. S. J. Pister, "Next century challenges: mobile networking for "Smart Dust"", *Proc. of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 271-278, USA, 1999.
3. Hans W. Gellersen and Albercht Schmidt and Michael Beigl, Kafil M. Razeeb, Cian O'Mathuna, "Multi-sensor context-awareness in mobile devices and smart artifacts", *Mob. Netw. Appl.*, Vol. 7, pp. 341-351, 2002.
4. Stephen J. Bellis, Kieran Delaney, Brendan O'Flynn, John Barton, Kafil M. Razeeb, Cian"Development of field programmable modular wireless sensor network nodes for ambient systems", *Computer Communications*, Vol. 28, pp. 1531-1544, 2005.
5. Ali El Kateeb, L. Azzawi, "Hardware Reconfiguration Capability for Third Generation Sensor Nodes: Design and Challenges", *Proc. Of Workshops conjunction with 22nd International Conference on Advanced Information Networking and Applications*, Japan, pp. 675-680, 2008
6. Ali El Kateeb, Aiyappa Ramesh, L. Azzawi, "Wireless Sensor Nodes Processor Architecture and Design", *Proc. of Workshops conjunction with 22nd International Conference on Advanced Information Networking and Applications*, pp. 892-897, Japan, 2008.
7. H. Hinkelmann, P. Zipf, M. Glesner, "Design Concepts for a Dynamically ReconfigurableWireless Sensor Node", *Proc. of First NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 436-441, USA, 2006.
8. H. Hinkelmann, P. Zipf, M. Glesner, "A Domain-Specific Dynamically Reconfigurable Hardware Platform for Wireless Sensor Networks", *Proc. of International Conference on Field-Programmable Technology 2007*, pp. 313-316, Japan, 2007.
9. Ting Liu, Margaret Martonosi, "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems", *Proc. of the 9th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pp. 107-118 USA, 2003.
10. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *"Numerical Recipes 3rd Edition: The Art of Scientific Computing"*, 3-rd edition, Cambridge University Press, pp. 608-639, USA, 2007.
11. Kenji Kodama, Naotaka Fujita, Yutaka Yanagisawa, Masahiko Tsukamoto, Tomoki Yoshihisa, "A Rule Engine to Process Acceleration Data on Small Sensor Nodes", *Proc. of the 5th International Conference on Pervasive Services, Poster Session*, Italy, 2008.
12. V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, M. Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications", *Proc. of the 2nd ACM Conference on Embedded Networked Sensor Systems 2004*, pp. 188-200, USA, 200409

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)