# AN DYNAMIC ENERGY MANAGEMENT ON FPGA FOR WIRELESS SENSOR NETWORK

Alexandre Ingles Da Silva[1] and Fabio Dacencio Pereira[2]

University Center Euripides Marilia, Hygino Muzy Filho, Marilia, Brazil

## *ABSTRACT*

*Wireless sensor networks are occupying large space in the modern world. Technologies like FPGA appear with features to be explored as implementation of low-power system. In this project we built a low-power circuit applying the implementation of a technique to minimize the clocks number of the circuit and focusing on the dynamic power consumption of the architecture was evaluated energy and logical consumption module to module. The architecture built has all the necessary features to integrate, synchronize and perform communication according to the application. Different analyses the consumption of a general architecture is shown and an operation mode acceptable to dynamic energy consumption of circuit to wireless sensor network on FPGA is achieved.*

## *KEYWORDS*

*Wireless Sensor Network, Logical Architecture, Low Power e FPGA*

## 1. INTRODUCTION

Wireless sensor networks (WSNs) are increasingly gaining popularity worldwide for its various advantages such as embedded devices. Applications of Wireless Sensor Networks (WSNs) are spreading across many specific applications area for its low cost encouraging many studies in the area. According Arampatzis [1] there are a lot of implementations of wireless sensor networks and there is no a standard definite. Irrespective of the exact type of platform, already known applications can be categorized under some general headings: military applications, environmental monitoring, commercial or human centric applications and applications to robotics. With the development of embedded system and network technology, there has been growing interest in providing fine-grained metering and controlling of living environments using low power devices. A typical sensor node has several components: a radio transceiver with an antenna which has the ability to send or receive packets, a microcontroller which could process the data and schedule relative tasks, several kinds of sensors sensing the environment data, and batteries providing energy supply [2].

To implement reconfigurable circuits of wireless sensor networks several types of technologies that aims to meet your requirements can be exploited. Currently technology like FPGA has gained attention by provides the ability and flexibility to implement logical circuits, allowing include more functions or import from other components. According Shashank [15] substantial performance growth and reduced power consumption for both FPGA and ASIC technologies in comparison to the realization of the basic non-restoring iterative algorithm. In reference [4] summarizes the different works that have been carried out and various techniques used at system , device, and architecture and circuit level to reduce the power consumption of FPGAs. It describes many of the significant improvements which have been made to

improve power and energy efficiency of FPGAs that ranges from low level processes and circuit design techniques to high level techniques. Although significant improvements have already been made, many opportunities to further reduce power in FPGAs remain. In [3] circuit techniques for reducing field-programmable gate-array (FPGA) power consumption and propose a family of new FPGA routing switch designs that are programmable to operate in three different modes: high-speed, low-power, or sleep. This technique allows FPGA architectures operate in reprogrammable modes of energy consumption, thus it is possible to implement circuits that require low power as Wireless Sensor Networks. Circuits-on-FPGAs as (NOCs) as wireless sensor network can be built using the digital logic of the FPGA and your interface to general purpose I/O, being necessary connect the physical devices at the external interface of the FPGA.

According ValVerde [4] the use of FPGAs allows Partial and Dynamic Reconfiguration (DPR) which has many advantages when working with wireless sensor nodes. The possibility of adapting the system functionality loading new blocks at run time in a very fast way, when working with changeable environments, opens up new opportunities, however the implementation of these techniques may not be enough to reach acceptable values of power consumption. Even though the FPGA may be working in a very efficient way, the rest of the components will consume energy even when they are not needed. In this project, a logic architecture for wireless sensor network is implemented on FPGAs, targeting manage the physical device and reduce the dynamic power consumption of the circuit. The circuit has been validated in a real physical architecture, allowing logical modules built to adapt to different families of FPGAs.

With that a logical architecture that consists of various modules is proposed, using the advantages of FPGA architecture makes it possible to implement any combinational and sequential circuit, which can range from a simple logic function to a high-end soft-processor [10], the logical architecture built aims reduce the circuit dynamic energy consumption and manage the physical devices connected to external interface of FPGA after establish network synchronization. Also was analyzed in various ways the dynamic total power consumption of the circuit, where dynamic energy is consumed by activity in the part [9]. A technic is used as a frequency splitter module that through its clock input are generated different frequencies to other modules, these frequencies is distributed into the logical architecture. The analysis of the power consumption of the circuit is based on dynamic energy, considering that FPGA devices differ among themselves: across device families and vendors the resources and routing architecture on each FPGA vary greatly [6]. The logical architecture is composed by several logic modules that were implemented, each one of these modules have roles within of the architecture. Although the modules are separated, they are logically connected together and a sync logic module is used for setting the timing, this timing is used to integration and synchronization among logical architectures.

## 2. DYNAMIC ENERGY ON FPGA

Dynamic clock management (DCM) can have a strong impact on the reduction of power consumption in FPGAs. Most modern FPGA clocks have managers to troubleshoot high clock speeds. They include PLLs and / or Delay Locked Loops (DLLs) together with dividers and clock circuits interface. Dynamic power is consumed by toggling nodes as a function of voltage, frequency, and capacitance and is dissipated when capacitances are charged and discharged during the operation of the circuit and consumed during switching events in the core or I/O of FPGA [14]. Dynamic programmable clock divider (DPCD) can be implemented with the aim to multiply frequencies in a logical architecture distributing it to other logical modules through logic signals, reducing the amount of dynamic energy consumed by the circuit.

For this a simple clock divider can be implemented. The initial clock rate should be used as input, thus reducing the frequency in several phases; these phases characterized by a frequency can be distributed by logical modules the circuit. While the frequency of operation of the circuit is decreased but still meeting the computational requirements, significant amounts of energy consumed by the circuit can be saved, with a good opportunity to implement circuits which aims at low power consumption. With a Dynamic programmable clock divider (DPCD) that reaches all the modules of the architecture it is possible to suggest an opportunity to save energy by running the application in low frequency during idle mode. As in any case of integrated circuits FPGAs can dissipate energy for a few reasons. One it's dynamic power dissipation that is dissipated when capacitances are charged and discharged during the operation of the circuit [12].

There are several other used techniques that can reduce the dynamic power consumption of the circuit as Reducing Dynamic Power on Clock Scheme; Reducing Logic and Signal Power; Reducing Power RAM; Reducing I / O Power. Each one must be considered to reduce dynamic power consumption of the circuit, and thus achieve an acceptable consumption for applications that requiring low power consumption.

## 3. HARDWARE SPECIFICATION

To build the proposed logical architecture makes use of physical modules, which are the physical parts, used to build the logical architecture. In these modules are included sensors, radios transceiver and SRAM-based FPGA as processor. These are used to validate the circuit implemented in VHDL language. The following has the details and all features of the physical devices used.

### 3.1. FPGA device

We have focused on a single FPGA device, the Xilinx Nexys3. The Nexys3 is a complete, ready-to-use digital circuit development platform based on the Xilinx Spartan-6 LX16 FPGA. The Spartan-6 is optimized for high performance logic, and offers high capacity, high performance, and resources. Offers 2,278 slices each containing four 6-input LUTs and eight flip-flops 576Kbits of fast block RAM to utilization [7]. Through the external interface FPGA as Pmod connectors are used to connect the device as sensor and radio transceiver, a USB-UART is used to serial communication with a PC and Adept USB Port are used to power on the FPGA.

### 3.2. Radio Transceiver and Sensor Devices

To transmit and receive data packets is used XBee-PRO 900 embedded RF modules that combine fast point-to-multipoint networking with the RF range advantages of 900 MHz technology. Built upon a 156 Kbps RF platform, these module are ideal for applications requiring low latency and increased data throughput [5]. The EM-408 GPS engine board is low cost but maintains high reliability and accuracy making it an ideal choice for integration with OEM/ODM systems and LV-MaxSonar-EZ1 provides very short to long-range detection and ranging, in an incredibly small package [8]. These devices are used as the physical part of the architecture that accomplishes the sensoring, through it is possible to validate the logical communication architecture.

## 4. LOGICAL ARCHITECTURE

The logical architecture consists of several logical components connected. The modules are logically described using the VHDL language. For power-aware systems [11] the logical architecture have a model of the current context of operation and the set of active tasks in order to

support decision-making and processing and Logic to decide which components may be stopped or suspended in a given situation. Thus the modules are built in order to logically integrate suspending the modules that aren't necessary, the physical devices connected to the external interface of the FPGA are stopped or suspended either. The following has the logical architecture in blocks.

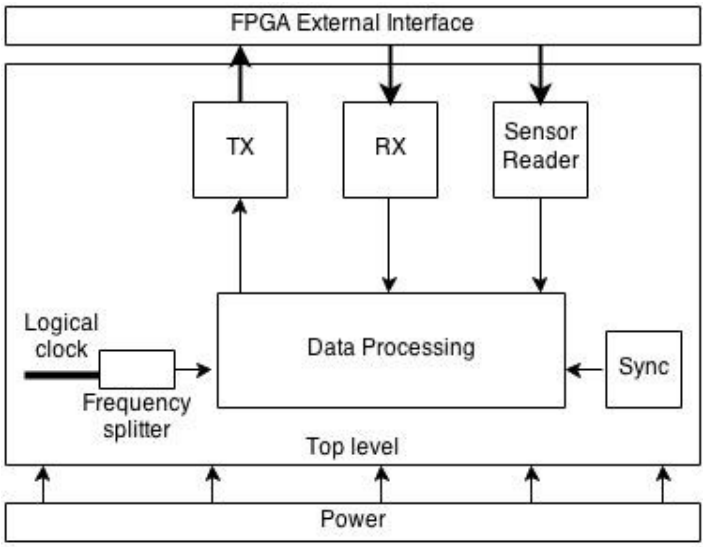


Figure 1 - Logical architecture in blocks

The modules are connected and their runtimes are managed by sync module, the frequencies of operation are managed by the splitter module while the other modules like TX, RX, data processing and a generic sensor reader are designed to accomplish the communication cycle.

## 4.1. Logical Modules

The architecture is composed of several logical modules; these modules are implemented in order to make the architecture autonomous, each module performing a function in order to realize the integration between architecture and accomplish the communication. In this session we present these modules implemented in the FPGA, logic consumption details by each one. Even the modules being separate some are dependent, logic signals are used to communicate each other. These modules are named as sensor readers, transceiver, receiver, data processing, frequency splitter, sync and finally a main module named top, this last connect all them. Below there are the details of the implementation of each logical modules.

### 4.1.1 Data Processing

This module performing the processing of data received and transmitted, any modules can be connected and reconfigurable. The received data by the receiver module and data read from sensor reader module are directly sent for processing, after processing these data are sent to the transmitter module, it performs transmission. In order to reduce the dynamic power consumption, the data processing module operate at B frequency, with minimum and maximum frequency rate the processing is affected, when operating at the minimum frequency is makes possible the minimum dynamic energy consumption by circuit module, but a larger amount of communication failure is identified and loss in performance when compared with the maximum frequency your

energy consumption is higher being identified better performance and greater reliability in communication.

The processing flow is implemented in VHDL language, making it possible to process data according to the need of communication. The data are processed in a generic form where only the size of the package is relevant to architecture. Only information necessary for processing of the package is included, the size of packets transmitted and received will be used to analyze the power consumption of the circuit in the session IV.B. A processing cycle is characterized from the receipt of the package to be processed until receipt of an acknowledgment packet of this packet transmitted. The following has the details of registers and LUTs used to build the module.

Table 1. Data Processing Logical Consumption.

| Used logic | Used | Available |
|---|---|---|
| Number of Slice Registers | 306 | 18224 |
| Number of Slice LUTs | 2861 | 9112 |

For the processing of data packets a large number of Slices LUTs and others registers are used, this happens by the need to process data packets received from the reader and receiver module.

### 4.1.2 Generic Sensor Reader Module (GRSM)

To validate the communication of architecture a generic sensor reader module was implemented, this module simply read the sensor data and after formed is performed processing. The following logical details of this module are presented.

Table 2. GSRM Logical Consumption.

| Used logic | Used | Available |
|---|---|---|
| Number of Slice Registers | 548 | 18224 |
| Number of Slice LUTs | 674 | 9112 |

Being a generic module consumes quite logic of the FPGA. Your use of LUTs is based on the need to form packets read from the sensor and pass for processing.

### 4.1.3 Transmitter Module (TX)

The transmitter module has a logic output connected to the external interface of the FPGA that connects at input (RX) of the radio transceiver. The logic module operates on several frequencies 1,200 bps to 230,400 bps, but has a size limit of the packet to be transmitted, the packet to be transmitted may contain different sizes, the size is identified at reception and then the transmitter is reported, having stipulated a size limit. The transmission is performed byte by byte, each byte is added stop and start bits. For a control power consumption of transmitter module circuit implementation is accomplished at predetermined times, thereby achieving the execution of the circuit only when required. This runtime control of transmitter circuit is taken based on a logically connected at a sync module. The transmitter module has logic control of the physical transceiver device activating, transmitting and disabling it.

Table 3. TX Logical Consumption

.

| Used logic | Used | Available |
|---|---|---|
| Number of Slice Registers | 177 | 18224 |
| Number of Slice LUTs | 298 | 9112 |

With the responsibility of transmitting data, this logical module requires to large amount of FPGA logic such as Slices Registers and LUTs. Its energy consumption is based on dynamic clock input, it's no possible use the A frequency as others by the need for a high clock rate to perform transmission of data.

### 4.1.4 Receiver Module (RX)

The receiver module has a logic input (RX) connected to the external interface of the FPGA that connects at output (TX) of the radio transceiver. The module operates with equal frequency settings of the transmitter module 1,200 bps to 230,400 bps. The package is formed of byte by byte where each byte is added start and stop bits. The receiver identifies the current size of the packet to be received, making it a dynamic receiver. As the transmitter module, executions of receiver circuit are at pre-defined time by the sync module logically connected. The receiver module is handled by sync module, after enable the receiver module for start execution the radio transceiver is enabled to accomplish the reception of data, when finished both modules are disabled in order to save energy.

Table 4. RX Logical Consumption.

| Used logic | Used | Available |
|---|---|---|
| Number of Slice Registers | 316 | 18224 |
| Number of Slice LUTs | 623 | 9112 |

Making use of registers for storing data received, a lot of logic in the FPGA is used, since the module has the characteristic dynamic data reception. As this module makes connections to physical devices connected to the FPGA, it uses the B frequency for execution of the circuit.

### 4.1.5 Sync Module

The logical Sync module was implemented to establish the integration and synchronize architecture with architecture. This module is responsible for keeping updated synchronization of time of receipt, reading sensors and execution of modules of the circuit; the times are updated every communication cycle after receiving the acknowledgment packet. It's used as reference for the operating times of the modules, since each module starts operation and out of operation is based on this module. To the synchronization in the architecture is used Reference Broadcast Synchronization RBS [13], where a receiver with receiver, instead of sender with receiver. The external physical devices connected to the I/O FPGA as sensors and radio transmitter are handled in accordance with the times set out for this module. These times are distributed between sleep and wake up.

*Sleep*

After completing the communication cycle the physical devices connected to FPGA with except FPGA are disabled and the circuit is not executed, only the module sync continues to run. The time sleep of the architecture is configured according to the application needs, after accomplish the integration of the network the logic modules has knowledge of these times.

*Wake-up*

In this case where the FPGA has not been power off even when in sleep the circuit does not lose its configuration, logical modules and physical devices are awake to accomplish the communication cycle.

Table 5. Sync Module Logical Consumption.

| Used logic | Used | Available |
|---|---|---|
| Number of Slice Registers | 38 | 18224 |
| Number of Slice LUTs | 89 | 9112 |

This module uses little logic of the FPGA, since it only marks the time of operation to other modules. Its dynamic energy consumption is mainly based on the B frequency used.

### 4.1.6 Splitter Module

With a logic clock input of 10MHz based on 100 MHz CMOS oscillator of FPGA the divider module generate A and B frequencies and it's distributed to the logic modules that composes the logical architecture. This distribution is applied in order to decrease the accumulation of clock's used by the circuit. The logic modules that are connected to the I/O external interface are configured according to the physical devices operation frequency, while the others have its frequency in 100 KHz to 1MHz.

Table 6. Splitter Module Logical Consumption.

| Lógica Utilizada | Used | Available |
|---|---|---|
| Number of Slice Registers | 26 | 18224 |
| Number of Slice LUTs | 63 | 9112 |

For this logic module the minimum FPGA logic was used, it is just a simple frequency splitter, having as input a logically clock of 10MHz.

### 4.1.7 Top Level Module

This module connects all the modules that is part of the architecture, logic signals and other types of connection that allows modules to communicate with each other are logically connected through this module. Also through this module the logic module and physical devices are connected using FPGA IOs, where I/O power depends on the amount used. To simulate the architecture a total of eight I/Os were used. The table VII show the logical consumption used to construct this module.

Table 7. Top Level Module Logical Consumption.

| Used logic | Used | Available |
|---|---|---|
| Number of Slice Registers | 2 | 18224 |
| Number of Slice LUTs | 1 | 9112 |

Little logic of the FPGA is used to build this module, it only connects the modules to form the architecture.

## 5. PERFORMANCE AND POWER CONSUMPTION ANALYSIS

To achieve good performance and acceptable dynamic power consumption of the circuit some techniques as splitter module for decrease clock's consumption and a rigid control of energy consumption in the circuit trough a sync module for establish synchronization is applied. The

different ways to analyze the dynamic energy consumption of the logical architecture is shown in this session.

## 5.1. Clock Frequency

The FPGA physical processor operates with a 100 MHz CMOS oscillator, but logically a clock rate of 10MHz is used at most. This logical clock of 10MHz is used as reference to generate A and B frequencies. These two frequency of operation are distributed by the circuit, one of the frequencies defined as A frequency is used to connect the logic modules to I/O interface, except transmitter module that make direct connection to clock input and can be configured according operation rate of the physical device. These I/Os have a direct connection with the physical devices used to validate the architecture, devices such as sensors and radio transceiver make use of this port configured with their relative frequencies of operation. Another frequency defined as B frequency is connected to the modules that make no connection to the I/O interface FPGA and can be configured according to the desired rate of operation.

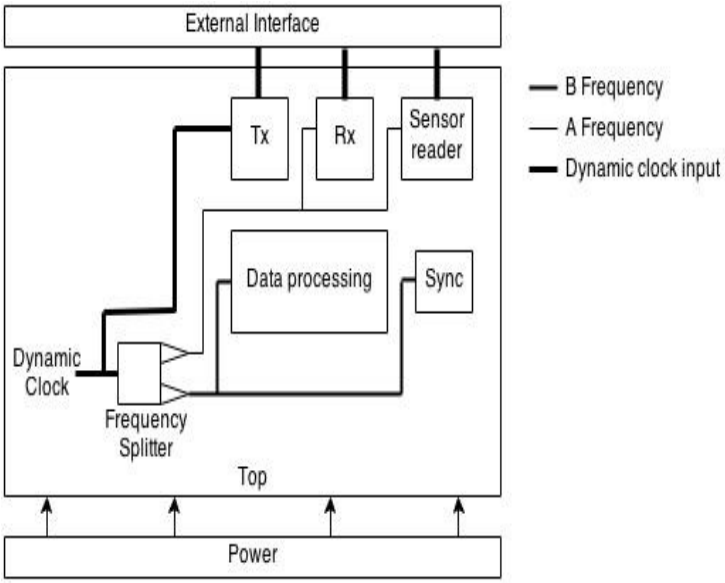


Figure 2 - Diagram logic in block of logical architecture

Figure 2 shows the logical architecture built in blocks, each of these modules are connected logically, is also shown the distribution of frequencies by the circuit. In the following table the consumption of frequencies generated by clock input is presented.

Table 8. Logical Clock and Frequencies Consumption.

| Clock Frequency | Consumption Average |
|---|---|
| Clock of 100KHz to 10MHz | 0.48mW to 0,81mW |
| Frequency A of 600 to 115200 bps | 0.4mW to 0.38mW |
| Frequency B of 100KHz to 1MHz | 0.54mW to 0.60mW |

In Table 8 it's displayed the power consumption average of the dynamic clocks that are distributed at various frequencies throughout the circuit. A logical clock ranging from 100kHz to 10MHz and it's responsible to generate A and B frequencies, the dynamic power consumption of each one is presented. The dynamic power consumption of each module is defined as:

$C = F_u + S_p + L_p$ where C it's the logical power consumption, $F_u$ it's the used frequency, $S_p$ signal power, $L_p$ logic power. (1)

This technique allows any logic module be integrated into the architecture, using one of two frequencies only their consumption of logic and signal power are added to the total dynamic energy consumption of architecture.

## 5.2. Dynamic energy consumption analysis

To analyze the dynamics consumption of circuit energy XPower Analyzer tool in the packet WebPack of the Xilinx ISE 13.4 is used. It is possible to check the consumption of the logical architecture in several ways; the following table shows logical and signal power of each module and following has the dynamic power consumption average of the circuit in different operational modes.

Table 9. Average of Logical and Signal Power for each Module.

| **Module Name** | **Signal Power Average** | **Logical Power Average** |
|---|---|---|
| Data processing | **<8µW** | **<8µW** |
| Sensor Reader generic | **<5µW** | **5µW** |
| Transmitter(TX) | **<12µW** | **<5µW** |
| Receiver(RX) | **<10µW** | **<10µW** |
| Sync | **<2µW** | **<1µW** |
| Splitter | **<1µW** | **<0.5µW** |
| Top Level | **<11µW** | **<11µW** |

These are logical and signal power consumption of each module; these modules are designed to accomplish the execution of the circuit with the minimum signal and logical power consumption.

### 5.2.1 Operation Modes

The operation modes of the circuit are divided in different analyzes; such as modules only operate at certain times defined after logical architecture integration, thus this analysis is the entire operation to realize a communication cycle and it's used a sensor reader generic for analyses. In the following table there are the operating modes of the circuit and its power consumption average. When the frequency of operation of the module data processing is less than 100kHz, it is not possible to perform data processing, then the analysis of dynamic power consumption of the circuit is performed with greater frequencies than 100kHz distributed among the logical modules.

The first mode of operation of the circuit is quite important because the circuit is at rest, in other words no other logic module is running, only the sync module is enabled. The mode 2 show the power consumption of circuit when operating in its normal mode, receiver and sensor reader modules are configured to operate with A frequency in a bade rate of 4800bps, transmitter operate in 4800bps and the other at 100KHz frequency B, while size of transmitted and received packets are 127 bits.

Table 10. Average of Circuit Energy Current Consumption.

| **Modes** | **Description of circuit operation** | **Dynamic Consumption Average** |
|---|---|---|
| Mode 1 | only the sync module is enabled, in this case no other logical module is executed. | 480μW |
| Mode 2 | Circuit operation, transmitter, sensor reader and receiver with a bade rate of 4800bps and others operate with a clock frequency of 100KHz, whereas packet sizes transmitted and received with a maximum size of 127 bits. | 1500μW |
| Mode 3 | Equal to 2, but data processing module operates with a frequency of 1MHz. | 2000μW |
| Mode 4 | Equal to 2, but packet sizes transmitted and received with a maximum size of 190 bits. | 1800μW |
| Mode 5 | Equal to 3, but packet sizes transmitted and received with a maximum size of 190 bits. | 2300μW |

In mode 3 the transmitter, sensor reader and receiver modules operate equal mode 2 while the others module data operates at 1MHz frequency B. The tests showed that this can be a good solution for a good performance, reliable in the communication and dynamic power consumption of the circuit. In case 4 the circuit is simulated with a maximum packet size of 190 bits, transmitter, sensor reader and receiver modules and others operate equal 2. Already in the mode 5 is simulated with the circuit package of 190 bits, transmitter, sensor reader and receiver modules operate equal 2, but with the others modules operate equal 3. In both cases, the circuit has increased consumption, this occurs by the need for transceiver and receiver modules use more clock to transmit and receive these packets.

## 6. CONCLUSIONS

There are several challenges in Wireless Sensor Networks that even with its limitations, increasingly gaining ground in our world. This project was built an architecture composed of several logical components to simulate the communication between logical sensor nodes in different configurations in order to manage the power consumption of the physical devices connected to the FPGA and running the circuit on schedule timers defined in the integration. The architecture provided the necessary conditions performing the integration, synchronization and communication between architectures establishing an autonomous connection. Although FPGAs consume a rather high amount of energy dynamics by using a high amount of clock, this project was possible to minimize this consumption by applying a frequency splitter module, thus saving a lot of dynamics energy in the circuit. It was also shown the dynamic energy consumption of the logical architecture in different configurations to perform communication.

## REFERENCES


[1] Th. Arampatzis, J. Lygeros, Senior Member, IEEE, and S. Manesis, Member, IEEE: A Survey of Applications of Wireless Sensors and Wireless Sensor Networks, 13th Mediterranean Conference on Control and Automation Limassol, Cyprus, 2005.

[2] Fei Yu, Raj Jain: A Survey of Wireless Sensor Network Simulation Tools: I.J. Information Engineering and Electronic Business, 2011.

[3] Jason H. Anderson, Member, IEEE, and Farid N. Najm, Fellow, IEEE: Low-Power Programmable FPGA Routing Circuitry, IEEE transactions on very large scale integration (VLSI) systems, 2009.

[4] Juan Valverde , Andres Otero, Miguel Lopez, Jorge Portilla, Eduardode la Torre, Teresa Riesgo, Using SRAM Based FPGAs for Power Aware High Performance Wireless Sensor Networks: Sensors 2012.

[5] XBee-PRO 900, Digilent, 2014. [Online]. Available: https://www.digilentinc.com

[6] Peter Yiannacouras, J. Gregory Steffan, Jonathan Rose, Exploration and Customization of FPGA-Based Soft Processors: Computer-Aided Design of Integrated Circuits and Systems, 2006 IEEE.

[7] Spartan 6 LX16 Nexys3, Xilinx, 2014. [Online]. Available: http://www.xilinx.com

[8] GPS Engine Board EM-408 , Digilent, 2014. [Online]. Available: https://www.digilentinc.com

[9] Reduce FPGA Power Consumption, eASIC corporation, 2014. [Online]. Available: http://www.easic.com

[10] Kuon, I.; Tessier, R.; Rose, J. FPGA Architecture: Survey and Challenges. Found. Trends Electron. Des. Autom. 2007, 2, 135–253

[11] Andy Nisbet, Simon Dobson: A systems architecture for sensor networks based on hardware/software co-design, 1st IFIP Workshop on Autonomic Communications, 2004.

[12] Naresh Grover, Dr. M.K.Soni: Reduction of Power Consumption in FPGAs - An Overview, Information Engineering and Electronic Business, 2012.

[13] J. Elson, L. Girod, and D. Estrin, “Fine-Grained Time Synchronization using Reference Broadcasts, " Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, MA, December 2002.

[14] Naresh Grover, Dr. M.K.Soni: Reduction of Power Consumption in FPGAs - An Overview: I.J. Information Engineering and Electronic Business, 2012.

[15] Shashank Suresh, Spiridon F. Beldianu and Sotirios G. Ziavras: FPGA and ASIC S quare for high Performance and Power Efficiency: 24th IEEE International Conferenceon Applicatio-specific Systems, Architectures and Processors,2013.