Research Journal of Applied Sciences, Engineering and Technology 8(17): 1855-1862, 2014
DOI:10.19026/rjaset.8.1173
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 20, 2014 Accepted: August 19, 2014 Published: November 05, 2014

Research Article

Hardware-software Reconfigurable Techniques for Wireless Sensor Network

[1]C. Rajasekaran, [2]R. Jeyabharath and [2]P. Veena
[1]Department of ECE, K. S. Rangasamy College of Technology, Tiruchengode, India
[2]Department of EEE, KSR Institute for Engineering and Technology, Tiruchengode, India

Abstract: Now-a-days, the industrial based real time embedded applications are highly developed and they act as a major role in the production cost and real world safety environment. In that, one of the advanced technique is that reconfigurable techniques. This technique plays a major role with wireless sensor networks for the efficient data transmissions. In recent days, most of the industrial applications are works to minimize the size and cost of device. The foremost improvement of the reconfigurable technique is that it circumvents the unnecessary hang and deferral in the device performance. In modern biosphere, Field Programmable Gate Array (FPGA) is one of the supreme operative reconfigurable devices and generally used for most of the hardware and software reconfiguration applications. In this study, the exertion pacts with whatever going to make changes in the hardware and software during runtime, it should not disturb the present successively process. This is the main impartial of the study that the changes to be done in an analogous manner at the same time concentrating the cost and power transmission problems during data trans-receiving and also the results to be seen with the help of online manner via the Ethernet cable.

Keywords: Advanced RISC Machine (ARM), Field Programmable Gate Array (FPGAs), runtime reconfigurable techniques, Wireless Sensor Networks (WSNs)

INTRODUCTION

VLSI and Embedded techniques are amalgamated and plays a key role for the modern technologies and at the same time exploration based on wireless network also difficult one due to the nodes in the environment which want to be work properly in every time with less power consumption. These are the major tasks want to concentrate before the research going on with WSNs (Zhuang *et al*., 2007; Willig, 2008). In the modern world applications, it has millions and trillions of sensor nodes revolve in an environment which makes a multifaceted task for the network researches. The major part of the industrial wireless sensor is an ultra-low power wireless sensor nodes that collects the processing signals from wherever the industrial sensor be present in the particular location and then it transmits the information to a base station through wireless networks for controlling and monitoring. If the interfaces and data processing to be done between the sensors, the communications are very difficult from one sensor to another and it also vary from one application to another application, so the designers are trying to overcome these problems in their day to day environment for the devices wanted to programmed and reprogrammed with application-specific hardware designs. In olden days, FPGAs were limited to performing the tasks that required only the fixed-point arithmetic, but in the recent advances, hardware's have made floating-point arithmetic on FPGA. The FPGAs are used to accelerate computationally intensive scientific computing applications that require floating-point arithmetic operations to be done in an efficient manner (Horta and Lockwood, 2004).

For all this major things, the main reasons are nodes and their power utilization which based on two major scenario namely the cost and memory utilization of the device which is increased so that the recent upcoming inventions are mainly based on the recent inventions which are mostly based on VLSI orientated the major advantage of this technology (Polastre *et al*., 2005). Runtime re-configurability is one of the up growing topics within the reconfigurable computing area and the changes to be done in the FPGAs configurations are at runtime, even the input/output device is inactive and any other logic is kept active. These powerful aspects are developed in Atmel FPGAs and Xilinx that permits to perform hardware updates at runtime also compared with memory space and programming time to occupy the FPGA reconfiguration.

This study mainly deals with a reconfiguration system in wireless sensor networks. FPGA is very difficult to give the analog input directly, so the to give the analog input directly, for that purpose the controller

Corresponding Author: C. Rajasekaran, Department of ECE, K. S. Rangasamy College of Technology, Tiruchengode, India

be used here as the major part of the controller used as the major part to convert the getting analog input into digital output through the FPGAs which would be done the reconfigurable process (Krasteva *et al*., 2011, 2008). In this method, ARM act as a part of microcontroller and it is very efficient one when compare to the previous microcontroller device and also it works under the Reduced Instruction Set (RISC) in the manner of instruction wise user friendly device which has 40 pin configuration, 6 ports and inbuilt analog to digital conversion units are the special things of the ARM microcontroller.

MATERIALS AND METHODS

In this section, an over-all part of the processing layer unit and the WSNs node parts are offered. Both the engineering based and speculative related approaches are presented here. ARM and FPGAs be act as a processing layer unit.

ARM LX2378: In the previous papers, the research is based on 8051 microcontroller. In the analog operation, the designer wants to make the Analog to Digital Converter (ADC) part separately so that it attain some amount of power consumption and also make delay in the design process and also want to make separate interface which are coming under ARM based design.

ADC part act as a major role in this paper, it should be the input part of the controller unit. In the ARM, the ADC is one of the special units compare to the previous microcontroller families which avoids the unwanted power consumption.

Power supply layer unit: This study deals with a Ethernet module for the data transmitting and receiving part where this module is controlled by ARM through the serial communication of Universal Asynchronous Receiver Transmitter (UART) and in future try with Wi-Fi or Bluetooth suppose if the node distance need to be increased.

Communication layer unit: This study deals with a Zigbee module for the data transmitting and receiving part and this module is controlled by PIC through the serial communication of Universal Asynchronous Receiver Transmitter (UART) and in future try with Wi-Fi or Bluetooth if the node distance need to be increased.

Sensor layer unit: The sensor layer has one of the additional features that is the interfacing done with both analog and digital sensors. The analog sensor (Temperature) to be connected with ARM microcontroller and the digital sensors to be connected with FPGAs. Here the digital sensors are considered as generalized manner (Fig. 1) which depending on the output from the microcontroller. The FPGA control the digital sensors and the information passed through the serial cable if there is a changes occur in the environment. The controller automatically passes the trigger to the FPGA depend on the triggering, then it check the current status of the digital sensors and again it reply back the strategy to the microcontroller. This event is being as a Runtime Reconfiguration.

During this process whatever the changes may arise in both hardware and software it does not affect the whole process, it only make changes in the particular partial reconfiguration things that depend on the coding and real time environment that changes the remaining process to be run in a parallel manner and this is the main specialty of the FPGA due to the interconnection between the blocks and wires so that it save most of the power loss and at the same time it reduce the designer work time in a single coding monitoring. In recent days, Scrofano *et al*. (2007) there are lot of myriad sensors are available in the market with many different interfaces. Most of them are digital sensors with various categories (Chaoui *et al*., 2006) such as I2C, SPI, 1-Wire, etc. First the signals have to be processed when the problems risen related to timing and processor using a microcontroller. Some manufacturers provide Hardware Description Language (HDL) code to implement the sensor interfaces in a coprocessor.

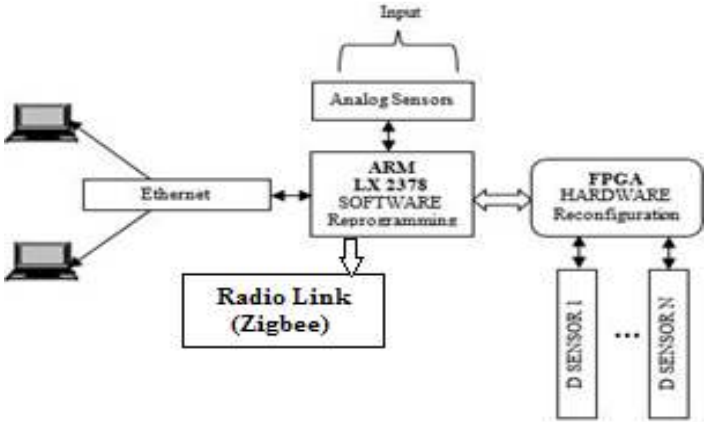


Fig. 1: The architecture of a sensor node system (A sensor means the analog sensor and D sensor means digital sensor)

A digital sensor is an electrochemical sensor, where data transmission and data conversions are done in a digital manner. Sensors are commonly used for analytical measurements, i.e., the measurement of physical and chemical properties of liquids. Some of the measured parameters are pH value, oxygen, conductivity, redox potentials and others.

RECONFIGURATION SCENARIOS OF NODE AND CONTROL FLOW

Network level based reconfiguration is the first reconfiguration scenario. This is one of the most required process during deployment and the response due to the response of final function of each node is defined here, this layer includes the used sensors and data processing otherwise the network may get any defect or the function is changed as an emergency situation. Substantially this scenario may put-on both, the hardware reconfiguration technique and software reprogramming. In the context of node platform, hardware reconfiguration is compulsory when modifications affect digital sensors (mentioned with "D" in the Fig. 1), as they are directly fastened with the FPGA. Opposite, software reprogramming is required for analog sensors (mentioned with "A" in the Fig. 1) due to fastened process in the ARM ADCs. Reconfiguration at Node Level is the second scenario, it mainly involves in the hardware configuration. In this scenario, a reconfigurable array blocks act as a reconfigurable coprocessor where the repetitive tasks lighten by the node Platform hardware device (FPGA) in parallel manner. This parallel task is done with the help of reconfigurable array blocks (Hinkelmann *et al*., 2006; Lymberopoulos *et al*., 2007). Here the FPGA has two main functionalities one deals with the analog sensor it fully act as a coprocessor another functionality is when it deals with the digital sensors it been acting as both coprocessor and digital sensor control. According to this both scenario, a partial runtime reconfiguration technique has been built on the top of the node. This partial reconfiguration and FPGA be act as a reconfiguration technique. The runtime reconfiguration resulting allows to modify the data processing, digital sensor interface or the interface between ARM and FPGA process them even the system in running time. The ARM is the core element for controlling all the reconfigurable process which acts as a position of receiving the software reprogramming and hardware configuration changes and manage with the FPGA reconfiguration. Therefore, all the reconfigurable process is controlled by the simple software coding parts done by the designer. The top of the node (Fig. 2) have abstraction layer part which has various platforms that consist of both hardware file and software library

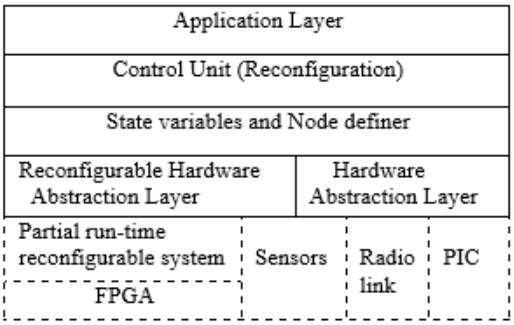


Fig. 2: Software stacks of reconfigurable node

files. The one unit of the abstraction layer is traditional hardware abstraction layer and the remaining one is reconfigurable hardware abstraction layer. The software programmer making the FPGA blocks depends on the functionality. After the abstraction layer, the above layer is node definer and state variable unit part that indicates the future work of the node and also it have some compressed messages like radio communication, sensor working unit, sensors working part and the variable defines the current using state of the resource node. Another layer is controlling unit part which directly maintains the node decision part. In that the node should take their decision in an independent manner during the time when the reconfiguration is needed for the node and another one is if the reconfiguration is not available at the time then the node going to decide, these are the two main works taken care by this unit The final top most layer is application layer where this layer decides the transmitting, receiving, sensing, data processing, etc.

ROLE OF PARTIAL RUNTIME SYSTEM IN WSN

The partial runtime system is the most efficient technique for runtime reconfiguration, but in recent days most of the parallel processing industrial applications are based on these techniques. The file based on partial configuration is in a compact manner. During the runtime or in normal working condition, it attains only less amount of energy and bandwidth. The designer needs to make the partial reconfiguration to allocate the Configurable Logic Blocks (CLB) for the specific operations in a correct manner. In this study the FPGA reconfigurable blocks are arranged based on Virtual Architectures (Vas) (Portilla *et al*., 2006). In this the virtual architecture has been divided into three major functionality, the first one is between microcontroller and FPGA, which is fixed one and it only do the interfacing operations (left side), the second one is a digital sensor accessing part, which deals the sensors in a generalized manner (right side) and the final one is in between this two major functionality

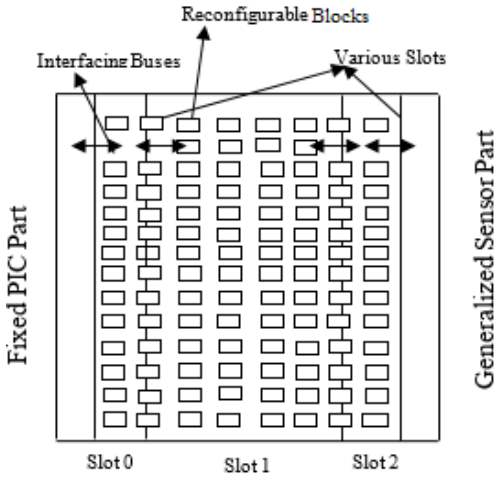


Fig. 3: FPGA virtual architecture view and slot allotments for node and hardcore

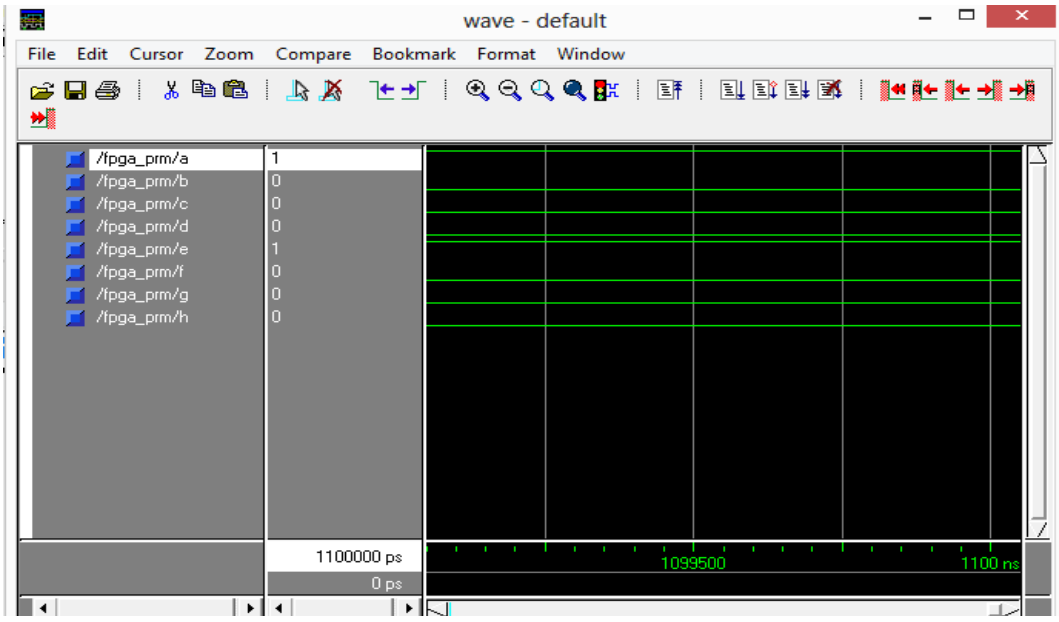
Fig. 4: MBC2300 board processing layer unit

which is the reconfigurable (FPGA) part of configurable logic block columns. The each CLB is a combination of four Lookup Tables. The Spartan 3 FPGA has 16 CLB columns and 24 CLB rows and each have some special slot functions (Steiger *et al*., 2004). In this study the reconfigurable FPGA has three major functions under that the first one is sensor interfacing part, the second one is reconfigurable part (Co processing unit) and the final one is FPGA and PIC interfacing part (Fig. 3). Thus, all functions to be split as three slots, the first slot 0 for PIC communication with the FPGA, then slot 1 for reconfigurable functions and slot 2 for sensor interfacing. These all interfacing work done with the help of Buses. In this both unidirectional and bidirectional buses are used for data transferring between the slots. By the use of bidirectional bus the data's be moved from left to right (4 bits) at the same time the data transfer from right to left (4 bits) and also pipelining technique are used in the architecture concept (Nolet *et al*., 2008), it reduces most of the delay in the data transmission apart from this (environmental condition, cable's, working mode) and it also decide the data transmission speed the following paragraphs explain detailing about the above delay things (Section below).

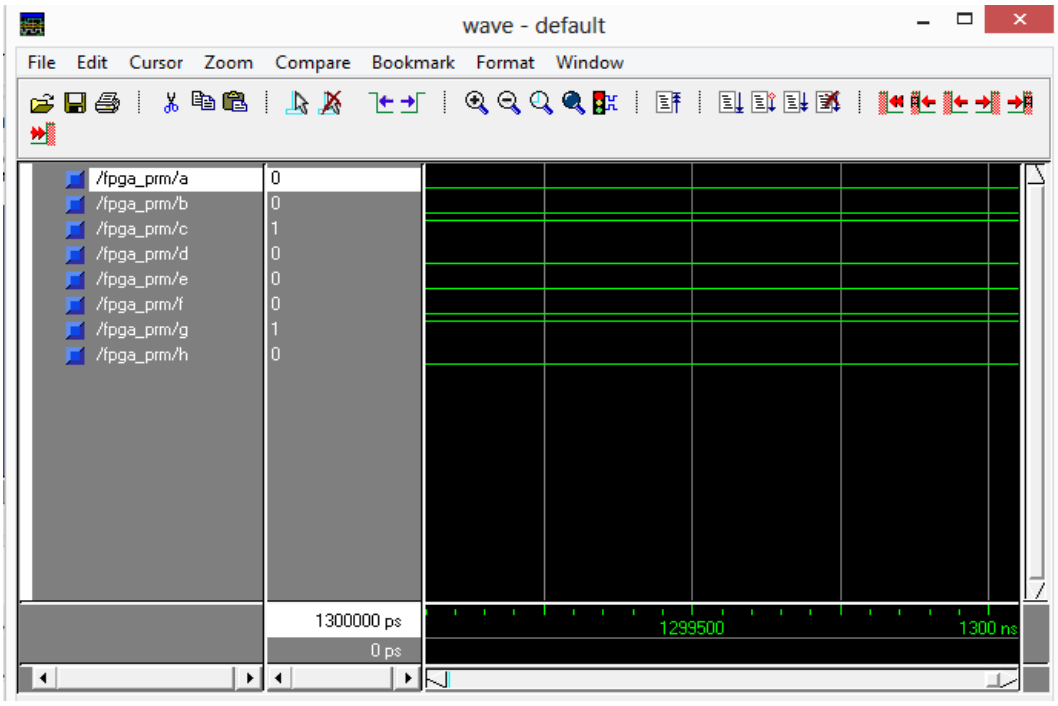
RESULTS AND DISCUSSION

In the result and parameter analysis part, it mainly concentrate on memory and energy. These two things related to the term of cost. These two parameters analyze in:

- The reconfigurable systems, which is use in wireless sensor network
- The changes of new hardware configurations and software programs along with wireless sensor network, Linke (2012)



(a)



(b)

Fig. 5: Reconfigurable simulation results of FPGA to the microcontroller from D. sensor 1 and D sensor 2

Hardware part: In this study hardware part design with the help of MCB2300 board (Fig. 4) processing the layer which built with the help of active and passive components and the ARM act as a heart of the nodes for all the controlling functions and reconfigurable function which would be carried out by the microcontroller and the sending unit is at one end (analog sensor) which always analyze the environment condition that depend on the reconfigurable function that be carried out in wireless sensor networks where the node be increased depending on the application.

Simulation results: The below simulation results (Fig. 5 and 6) clearly shows that the exact reconfigurable process of the FPGAs depend upon the receiving inputs from the ARM to FPGAs. The FPGAs did the transmitting part from FPGA to ARM depend on the environmental condition and then the reconfiguration will occur.

The process is done in the FPGA side to check the digital sensors conditions continuously if they get any changes in the interior side of the valve it intimate to the processor through the bidirectional bus.

This result shows that the simulation output of a reconfigurable process between the controller and FPGA kit through the medicine tool. Fatherly wants to analyze the exact automation process that makes the practical setup for viewing the exact range of output through the data transmission range which increased with the help of Zigbee.

- Transmission Energy:

$$Etrans = Ttrans * Ptrmode \quad (1)$$

where,
Ttrans : The time needed to transmit a configuration
Ptrmode : The power consumption in this working

- Reception energy:

$$Erec = Trec * P\ recmode \quad (2)$$

- Reconfiguration energy is the energy needed for a single reconfiguration:

$$Erecnf = Treconf * Preconfmode \quad (3)$$

- Transmission cost:

$$Ctrans = Etrans/Enode \quad (4)$$

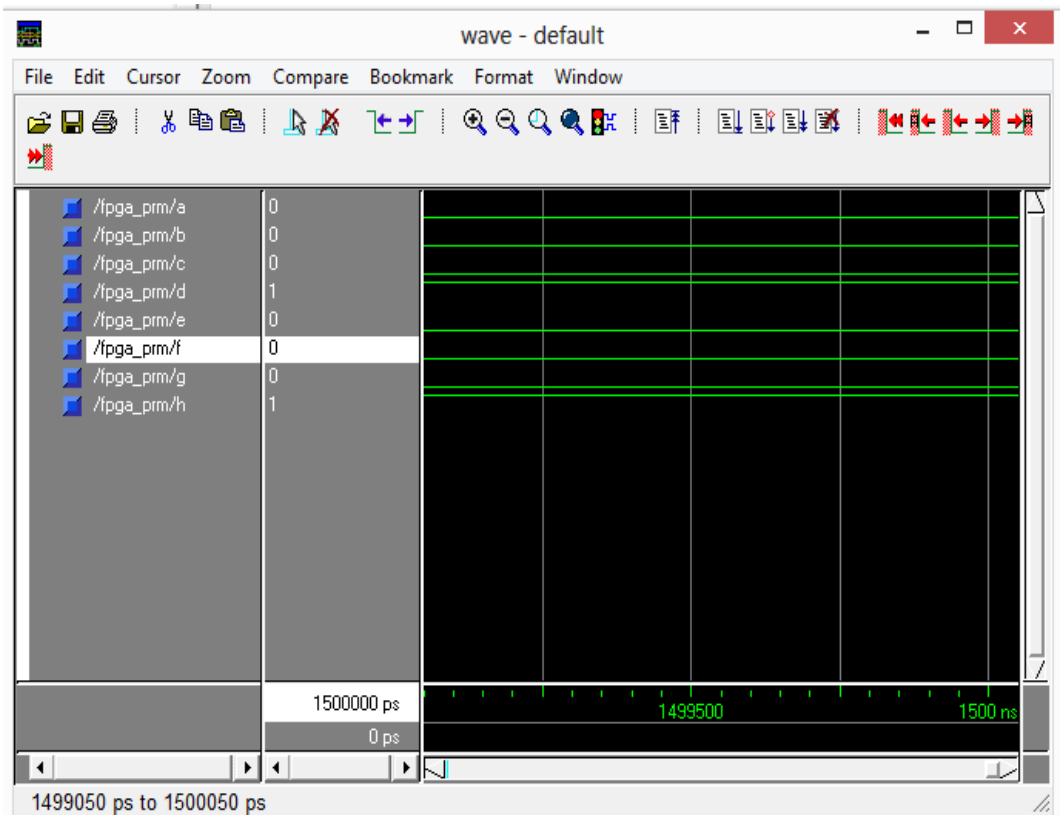
- Reception cost:

$$Crec = Erec/Enode \quad (5)$$

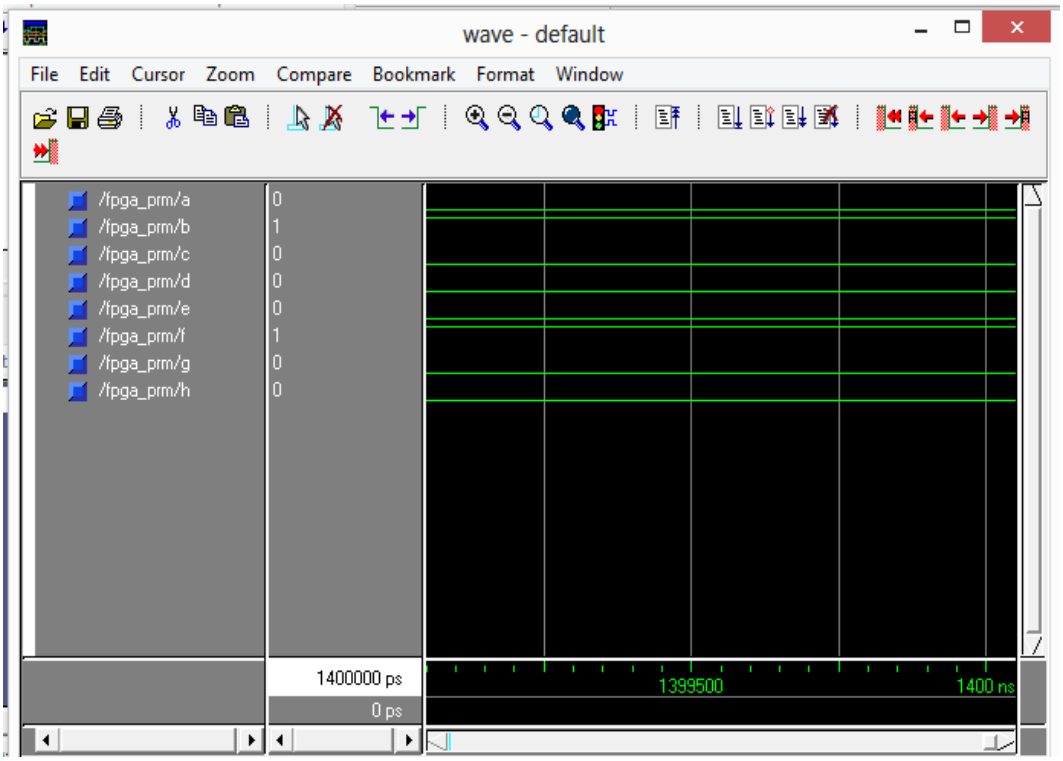
- Retransmission cost:

$$Cret = Eret/Enode \quad (6)$$

- Reconfiguration cost:



(a)



(b)

Fig. 6: Reconfigurable simulation results of FPGA to the microcontroller from D. sensor 3 and D sensor 4

(a)

(b)

Fig. 7: Online webpage result through the ethernet cable

Table 1: Data transmission rate

Desired baud rate	Calculated baud rate	X	Error rate
9600	9615	25	0.16
2400	2377	104	0.96
4800	4717	52	1.73
19200	17857	13	6.99

$$Creconf = Ereconf/Enode \quad (7)$$

Baud rate analysis: In normal form of the signal, it has 8 data bits and also has start and stop bit. The time period of bit is set by the baud rate. The typical value is 9600 baud which is about 10 k bits per second. In this study 16 MHz crystal oscillator are used as clock for this kind of mid-range PIC and this clock rate gives better resolution and attain baud rate is 9600 compared to previous reconfigurable analysis work which reduces more delay in the data transmission path. For a device with FOSC of 16 MHz, desired baud rate is about 9600. The following equations referred from Microchip.com. Asynchronous mode 8 bit BRG. The term BRG means Baud Rate Generator:

Desired Baud rate = FOSC/64 ([SPBRGH: SPBRG] +1)
Solving for SPBRGH: SPBRG
X = ((FOSC/Desired Baud rate) /64) /1 = ((1600 0000/9600)/64) /1, = 25.042 = 25
Calculate baud rate = 16000000/ (64 (25+1)) = 9615

Error = ((Calculate baud rate - Desired baud rate) /desired baud rate) *100 = ((9615-9600) /9600) * 100 = 0.16%

Another formula for calculating the baud rate depend on the bit rate:

X = ((FOSC/baud rate) - 16) /16

Formula for producing 8 or 16 bit SPBRG solutions: SPBRG = INT (Fosc/Baud Rate/Divisor - 1). The whole analysis part detail about the different baud rate medium in PIC and also explain how much the data transmission is varied in different baud rate condition (Table 1).

Webpage output through ethernet: The following (Fig. 7) shows the results of the sensors condition through the online by the way of Ethernet where one of the wired network in Local Area Network works with the help of the advanced technology the results be seen through anywhere by using particular IP address and

also it avoids the huge amount of problems even if anyone of the Zigbee node not worked.

CONCLUSION

The work deal in this study is trying to improve the efficiency of data transmission in runtime reconfigurable techniques in wireless sensor network. From the help of the Advanced Microprocessor (ARM), lot of work time is reduced for the designer at the same time the resolution and the data transmitting efficiency also be improved. In future work, the processors like ARM 9, 10 series may improve the data transmission rate by using various transmitting protocols. The most efficient thing is whatever the task may allocate in dynamic manner, the changes be done in the both hardware and software configuration in a runtime condition. This kind of real time application will act as a major role in the industrial automation and maintenance function.

REFERENCES

Chaoui, H., M.C.E. Yagoub and P. Sicard, 2006. FPGA implementation of a fuzzy controller for neural network based adaptive control of a flexible joint with hard nonlinearities. Proceeding of IEEE International Symposium on Industrial Electrons, 4: 3124-3129.

Hinkelmann, H., P. Zipf and M. Glesner, 2006. Design concepts for a dynamically reconfigurable wireless sensor node. Proceeding of 1st NASA/ESA Conferences on Adaptive Hardware and Systems (AHS'06), pp: 436-441.

Horta, E.L. and J.W. Lockwood, 2004. Automated method to generate bitstream intellectual property cores for Virtex FPGAs. Proceeding of 14th Field-Programmable Logic and Applications (FPL'04), pp: 975-979.

Krasteva, Y.E., J. Portilla, J.M. Carnicer, E. de la Torre and T. Riesgo, 2008. Wireless sensor networks node with remote HW/SW reconfiguration capabilities. Proceeding of IEEE Annual Conferences on IEEE Industrial Electrons Soc. (IECON'08). Orlando, FL, pp: 2483-2488.

Krasteva, Y.E., J. Portilla, E. de la Toree and T. Riesgo, 2011. Embedded runtime reconfigurable nodes for wireless sensor networks applications. IEEE Sens. J., 11(9).

Linke, B., 2012. Understanding the DS1WM Synthesizable 1-wire Bus Master. Maxim Integrated Products, Inc., Datasheet.

Lymberopoulos, D., N.B. Priyantha and F. Zhao, 2007. mPlatform: Arecon-figurable architecture and efficient data sharing mechanism for mod-ular sensor nodes. Proceeding of 5th IEEE/ACM International Conferences on Information Process. Sensor Networks (IPSN'07), pp: 128-137.

Nolet, V., P. Avasare, H. Eeckhaut, D. Verkest and H. Corporaal, 2008. Run-time management of a MPSoC containing FPGA fabric tiles. IEEE T. VLSI Syst., 16(1): 24-33.

Polastre, J., R. Szewczyk and D. Culler, 2005. Telos: Enabling ultra-low power wireless research. Proceeding of 4th International Symposium on Information Processing in Sensor Networks (IPSN'05), pp: 364-369.

Portilla, J., J.L. Buron, A. de Castro and T. Riesgo, 2006. A hardware library for sensors/actuators interfaces in sensor networks. Proceeding of IEEE International Conferences on Electrons Circuits and Systems (ICECS'06). Niza, France, pp: 1244-1247.

Scrofano, R., M.B. Gokhale, F. Trouw, V.K.E. Monmasson and M.N. Cirstea, 2007. FPGA design methodology for industrial control systems-A review. IEEE T. Ind. Electron, 54(4): 1824-1842.

Steiger, C., H. Walder and M. Platzner, 2004. Operating systems for reconfigurable embedded platforms: Online scheduling of real-time tasks. IEEE T. Comput., 53(11): 1393-1407.

Willig, A., 2008. Recent and emerging topics in wireless Industrial communications: A selection. IEEE T. Ind. Inform., 4(2): 102-124.

Zhuang, L.Q., K.M. Goh and J.B. Zhang, 2007. The wireless sensor networks for factory automation: Issues and challenges. Proceeding of IEEE Conference on Emerging Technologies and Factory Automation (ETFA'07), pp: 141-148.