IMPLEMENTATION OF EFFICIENT WIRELESS SENSOR NETWORKS

Mercy Subaraman

(*Electronics, Sir MVIT/VTU, India)*

Abstract—*Sensor networks offer a powerful combination of distributed sensing, computing and communication. They lend themselves to countless applications and, at the same time, offer numerous challenges due to their peculiarities, primarily the stringent energy constraints to which sensing nodes are typically subjected. The sensor nodes in a wireless sensor networks are normally microcontroller based which are having limited computational capability related to various applications.FPGAs offer flexibility,reconfigurability and cost advantages for hardware-in-the-loop simulation of sensors .This paper emphasizes the implementation of wireless sensor network using FPGAs with real time processing cores for safety applications. The usage of FPGA systems in real time domain is a very fruitful proposition as the FPGA devices are coming with processing cores for Real Time data. In order to add safety,RSA Algorithm has been integrated into the FPGA based wireless sensor network.*

Index Terms*: Communication, FPGA, RSA, WirelessModules, ADC, RTOS*

IJSER

I. INTRODUCTION

Wireless sensor nodes imply a large number of self power nodes which collect the information, detect the special events and communicate it in a wireless manner. The main goal is to handle the processed data to a base station safely. The main three elements which make them to be contributed to various applications:sensing,processing and communication. Recent advances in Low power VLSI, Embedded computing are making this emerging technology as a reality.

1.1 Application of sensor networks:

Possible applications of sensor networks are of interest to the most diverse fields. Environmental monitoring, warfare, child education, surveillance, micro-surgery, and agriculture are only a few examples. Intel's Wireless Vineyard [B11] is an example of using ubiquitous computing for agricultural monitoring. In this application, the network is expected not only to collect and interpret data, but also to use such data to make decisions aimed at detecting the presence of parasites and enabling the use of the appropriate kind of insecticide. Data collection relies on data mules, small devices carried by people (or dogs) that communicate with the nodesand collect data.

Military applications are plentiful. An intriguing example is DARPA's self-healing minefield a self organizing sensor network where peer-to-peer communication between anti-tank mines is used to respond to attacks and redistribute the mines in order to heal breaches, complicating the progress of enemy troops.Urban warfare is another application that distributed sensing lends itself to. An ensemble of nodes could be deployed in a urban landscape to detect chemical attacks, or track enemy movements. PinPtr is an ad hoc acoustic sensor network for sniper localization developed at Vanderbilt University. The network detects the muzzle blast and the acoustic shock wave that originate from the sound of gunfire. The arrival times of the acoustic events at different sensor nodes are used to estimate the position of the sniper and send it to the base station with a special data aggregation and routing service.

1.2Characteristics of Sensor networks.

i) **Lifetime**

Lifetime is extremely critical for most applications, and its primary limiting factor is the energy consumption of the nodes, which need to be self-powering. Many researchers suggest that energy consumption could be reduced by

considering the existing interdependencies between individual layers in the network protocol stack. Routing and channel access protocols, for instance, could greatly benefit from an information exchange with the physical layer.

ii) **Flexibility**

Sensor networks should be scalable, and they should be able to dynamically adapt to changes in node density and topology, like in the case of the self-healing minefields. In surveillance applications, most nodes may remain quiescent as long as nothing interesting happens. However, they must be able to respond to special events that the network intends to study with some degree of granularity. In a self-healing minefield, a number of sensing mines may sleep as long as none of their peers explodes, but need to quickly become operational in the case of an enemy attack. Response time is also very critical in control applications in which the network is to provide a delay-guaranteed service.

iii) **Maintenance**

The only desired form of maintenance in a sensor network is the complete or partial update of the program The only desired form of maintenance in a sensor network is the complete or partial update of the program code in the sensor nodes over the wireless channel. code in the sensor nodes over the wireless channel. Time synchronization is advantageous in promoting cooperation among nodes, such as data fusion, channel access, coordination of sleep modi , or security-related interaction.

iv) **Data Collection**

Data collection is related to network connectivity and coverage. An interesting solution is the use of ubiquitous mobile agents that randomly move around to gather data bridging sensor nodes and access points, whimsically named data MULEs (Mobile Ubiquitous LAN Extensions). The predictable mobility of the data sink can be used to save power as nodes can learn its schedule.

2. FPGA

Sensor Network (WSN) protocol implementation is a challenging issue when one has to deal with several objectives such as durability, portability, and evolutivity. These are difficult to satisfy while rapid development of prototype is an overall requirement. Implementing advanced and sophisticated WSN protocol in a flexible, durable hardware platform using Multiple FPGAs which allows time-saving in terms of qualification and long term exploitation of WSN.

3.. DESIGN METHODOLOGY

3.1. Real Time Operating System (Xi/kernel)

Real time embedded systems are typically designed for various purposes such as to control or to process data, meeting certain deadlines at the right time. To achieve this purpose, real-time operating systems (R TOS) are often used.

RTOS can be defmed as: "a program that schedules execution in a timely manner, manages system resources, and provides a consistent foundation for developing application code. An RTOS is a piece of software with a set of APIs for users to develop applications. RTOSes are typically differentiated from generic OSes regarding the following criteria i.e. Preemptive or prioritybased scheduling, Predictability in task synchronization, deterministic behaviors . Multitasking, other key features of RTOS which usually means that the software is divided into tasks, or smaller subsets of the total problem and at run-time, creating an environment that provides each task with its own processor.There are various RTOSes are available for for FPGA based design ie. VxWorks, QNX, eCos, LynxOS, and RTLinux. Here we have chosen Xilkernel as our RTOS which isprovided by Xilinx. Xilkernel is a small, robust, and modular kernel. It is highly integrated with the Platform Studio Frame .It allows a very high degree of customization . It supports the core features required in a lightweight embedded kernel, with a POSIX API. Xilkernel works on both the MicroBlaze and PowerPC 405 processors. Xilkernel has verylow memory footprint, it uses 7-16 kb of BRAM in a multithreaded program, which is much smaller than the RTOS used in microcontroller. Now scheduling is also a major issue in the environment of multitasking, Xilkernel supports priority driven, preemptive scheduling with time slicing or simple round-robin scheduling.Xilkernel is structured as a library. The user application source files must link with Xilkernel to access Xilkernel functionality. In Xilkernel a thread is the unit of execution and is analogous to a process. Threads are coded like functions.

IJSER

3.2. Hardware based modular exponential operation

Modular exponentiation is a type of exponentiation performed over a modulus. It is particularly useful in computer science, especially in the field of cryptography. In this paper the cryptography algorithm that we used is RSA. The exponential heuristics developed for computing M are applicable for computing M (mod n) .

C=Mmod n

Doing a "modular exponentiation" means calculating the remainder when dividing by a positive integer n (called the modulus) a positive integer M (called the base) raised to the

e-th power (e is called the exponent). The first rule of modular exponential is that we do not compute M because both M and e may be very large in case RSA algorithm. If we going to store M a large no of memory space needed. The temporary result must be reduced modulo n at each step of exponentiation this is because of space requirement of M is enormous . If M and e have 256 bit each we need 1080 bits to store M . This number is approximately equal to the number of particle of universe. In order to compute this total bit capacity all computers in the world we can make an assumption there are 512 million computers, each of which has 512 Mbytes of

memory. Then total number of bits available on all computers

will be 1018 . So we have no way to compute M. We have many hardware compatible algorithms to implement RSA; i.e. Binary Method, M array Method etc For our implementation the main concern was about the resources usage and execution time. Due to these two merits processor should have minimum number of modular multiplication to compute M mod n. According to the reference binary method is one of the fastest methods to compute modulus exponent without computing M. In reference execution time is depending on the value of exponent term ('e').

3.3. Algorithm

We present the RSA algorithm in Fig. 1, where both the encryption and decryption algorithms have the same steps, but only differ in the exponent term.
Thisproposedalgorithm is aModified version of that in [1] , the modification is done to handle the exponential operation in a better way. The real time scenario of the system is analyzed and its hardware utilization proves that it's better compared to the related works.

Proposed RSA Algorithm for encryption	Proposed RSA Algorithm for decryption
Input: M,e,n *Output* $C = M^e \bmod n$ *1.* $e_{k-1} = 1$ *then C:=M else C:=1* *2. For i=k-2 down to 0* *2a. C=C*C (mod n)* *2b. if* $e_i = 1$ *then C=C*M(mod n)* *3. return C*	*Input: M,e,n* *Output* $M = C^e \bmod n$ *1.* $e_{k-1} = 1$ *then M:=C else M:=1* *2. For i=k-2 down to 0* *2a. M=M*M (mod n)* *2b.if* $e_i = 1$ *then M=M*C(mod n)* *3. return M*

Fig. 1: Proposed Algorithm

In our algorithm we have 3 inputs M, n and e where all the inputs have k number of bits. This binary method checks each of the bits of the exponent term from left to right. Depending on the scan bit value a squaring and a subsequent multiplication operation is performed for each step (2a and 2b). As an example; let e=01111011. Which implies k=8. Since ek_1=O, we take C=1. The binary method proceeds as shown in Table II, which shows Mi23 is 7+5=12. It is very obvious the number of clocks to execute modulus exponent only depends on the number of modulus multiplications rather than the value of exponent term.

TABLEI
STEPWISE RESULT OF MOD CALCULATION ALGORITHM

i	e_i	Step 2a	Step 2b
6	1	1	M
5	1	M^2	M^3
4	1	M^6	M^7
3	1	M^{14}	M^{15}
2	0	M^{30}	M^{30}
1	1	M^{60}	M^{61}
0	1	M^{122}	M^{123}

IJSER

4.PROPOSED DESIGN

In the proposed design, temperature will be sensed by the thermistor and is given to Analog to Digital Convertor which is directly provided to input pin of FPGA Board (Cyclone II FPGA).Encryption part of RSA algorithm is implemented using VHDL in the first FPGA board. Due to the presence of RTOS, large number of processing can be done upon the data obtained from the sensor.Wireless module transmitter connected with the first FPGA transmits the encrypted data in a wireless fashion. And the wireless module receiver will receive the original data and will provide it to the second FPGA Board (Cyclone II FPGA) after implementing decryption algorithm in the second FPGA Board.Digital output can be verified in the second FPGA Board.

5. CONCLUSION

On-FPGA communication architectures play a crucial role in determining the performance and energy consumption of platform-FPGAs containing embedded coarse-grain modules. The design of efficient and reliable communication architecture is a challenging multi objective Optimization problem. The paper proposes the implementation of wireless sensor network using FPGAs with real time processing cores for safety applications. With the implementation of Real Time Operating System in FPGA,Sensor networks will be contributable to a vast variety of applications with high efficiency.

REFERENCES

[1] W. Wolf, "FPGA-Based System Design", Prentice Hall, New Jersey 2004

[2] S. Kilts, "Advanced FPGA Design: Architecture, Implementation, and
Optimization", Wiley & Sons, Wiley-IEEE Press, August 2007

[3] J. Boercsoek, A. Hayek, M. Umar, "Implementation of a 1oo2-RISCArchitecture
on FPGA for Safety Systems", 6th ACS/IEEE International Conference on Computer Systems and Applications, Doha, 2008

[4] J. Boercsoek, A. Hayek, B. Machmur, M. Umar, "Design and Implementation of an IP-based Safety-related Architecture on FPGA", XXII International Symposium on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 2009

[5] Actel Inc., "SmartFusion and Fusion Mixed-Signal FPGAs", http://www.actel.com, 2011

[6] Xilinx Inc., http://www.xilinx.com, 2011

[7] J. Boercsoek, "Functional Safety: Basic Principles of Safety-related Systems", Huethig, Heidelberg, 2006

[8] J. Boercsoek, "Electronic Safety Systems: Hardware Concepts, Models
and Calculations", Huethig, Heidelberg, 2004

[9] International Electrotechnical Commission, "IEC/EN 61508: International standard 61508 functional safety: safety related systems, Geneva; 2005

[10] A. Birolini, "Reliability Engineering: Theory and Practice", Springer
Verlag, Heidelberg, 2007

[11] Xilinx Inc., "Spartan-3E FPGA Family: Data Sheet", DS312, August 2009

[12] Connectblue, http://www.connectblue.se, 2011

[13] Bosch Sensortec, "BMA180: Digital, triaxial acceleration sensor", Reutlingen, 2010

[14] J. Blanchette, M. Summerfeld, "C++ GUI Programming with Qt 4", Prentice Hall International, New Jersey, 2006

[15] H. Asadi, M.B. Tahoori, B. Mullins, D. Kaeli, K. Granlund, "Soft Error

IJSER

Susceptibility Analysis of SRAM-Based FPGAs in High-Performance
Information Systems" IEEE Transactions on Nuclear Science, Vol. 54
Issue 6, Snowmass Village, CO, USA, 2007

[16] MISRA, "The Motor Industry Software Reliability Association",
http://www.misra.org.uk,2011

IJSER

IJSER