Sensor Node Processor for Security Applications

Goran Panić, Thomas Basmer, Oliver Schrape, Steffen Peter, Frank Vater, Klaus Tittelbach-Helmrich
IHP, Im Technologiepark 25, D-15236 Frankfurt (Oder), Germany
panic, basmer, schrape, peter, vater, tittelbach@ihp-microelectronics.com

***Abstract*— In this paper we present a sensor node processor designed to support complex data encryption/decryption operations. The system is developed around an asynchronous processor core supported by AES, ECC and SHA-1 crypto accelerators. The paper describes the chip architecture and its components and gives the chip implementation details. Finally, the power and performance of the chip have been discussed and analyzed.**

I. INTRODUCTION

A wireless sensor network (WSN) consists of a number of sensor nodes deployed randomly in the area. Many applications require high level of confidentiality during the data transfer. That makes the security a major issue when designing dedicated sensor node hardware. Due to limited battery resources, a long time operation of a sensor node is only possible when power consumption is low. Therefore, the primary goal of this work was to design a sensor node processor architecture that supports a variety of secure data protocols by maintaining low power consumption.

Security in wireless sensor networks has been widely investigated. The problem of data security has been addressed at two abstraction levels, software and hardware. The software-centric approach tries to optimize security algorithms making them applicable with standard sensor node solutions. This approach requires sensor nodes that provide sufficient memory storage for variables and software of current secure algorithms. In many cases, the available resources of traditional sensor nodes turned out to be insufficient to support complex crypto operations. Additionally, due to the limited computing power of general-purpose microcontrollers, the performance of traditional sensor nodes remains affected by the software implementation of encryption algorithms. Hardware-based solutions promise high performance and efficiency at the cost of chip area, flexibility and power consumption. The hardware solutions incorporate dedicated accelerators or coprocessors to support high-speed data cryptography [1].

The power reduction still remains a critical task when designing sensor node hardware. At the software level, the power is addressed by novel communication protocols and algorithms optimized to reduce overall traffic in the network. Consequently, to support implemented software solutions, the target hardware must be optimally tuned. At the physical level, power optimization includes implementation of advanced power saving methodologies such as dynamic frequency and voltage scaling, multi-voltage design and power gating. Also, some works propose usage of dedicated sensor node processors or application of special power-optimized system architectures.

To support low-power operation, our solution utilizes an asynchronous nature of the main processor that wakes up peripherals only when they have been addressed. Otherwise, all peripherals sleep with their clocks inactive. The support for cryptographic operation is provided via hardware accelerators tuned for low power. The rest of the paper is organized as follows: Section II discusses security issues related to sensor networks. Section III describes the architecture of the implemented sensor node processor system. Section IV gives the implementation results, Section V presents results and Section VI concludes the paper.

II. SECURITY IN WIRELESS SENSOR NETWORKS

Sensor nodes are usually deployed randomly in the area where they can be easily compromised. The most common approach to face security challenges in a sensor network relies on secret key cryptography. There are two families of algorithms that are commonly used: asymmetric (public-key cryptography) and symmetric (shared-key cryptography). Applying public-key cryptography in a traditional microcontroller-based sensor node was generally considered to be infeasible for WSN applications [2]. However, recent works show that with certain simplification, public key cryptography can be successfully implemented in embedded systems [3]. The techniques based on symmetric cryptography are generally faster and more power efficient compared to asynchronous. That made them the preferred solution for WSN applications. Some of the protocol-based symmetric algorithms are specially developed for implementation in sensor networks [4-6]. The major problem of symmetric cryptography remains the key establishment and distribution, since centralized distribution sites are not feasible for WSN. To solve this problem, some works propose key predistribution before deployment or having a random key distribution [7].

The problem of key distribution can be addressed by using asymmetric algorithm only for the key exchange. Once the key is available, it is used to implement a symmetric algorithm for data encryption. Additionally, to increase the security and ensure data integrity, one can apply a secure hash algorithm to

the payload. To support such behaviour, we designed a processor system that implements dedicated hardware accelerators for symmetric, asymmetric and secure hash crypto algorithms. The rest of the paper presents system architecture of the implemented processor and discusses implementation details.

III. PROCESSOR ARCHITECTURE

The designed sensor node processor is built around an asynchronous 16-bit processor core developed by Fraunhofer IPMS [8]. The processor connects a number of peripherals: three digital IO ports, a timer, a SPI, two UARTs, a baseband processing unit and hardware accelerators for crypto operations (ECC, AES and SHA-1). The architecture of the system is given in Figure 1.

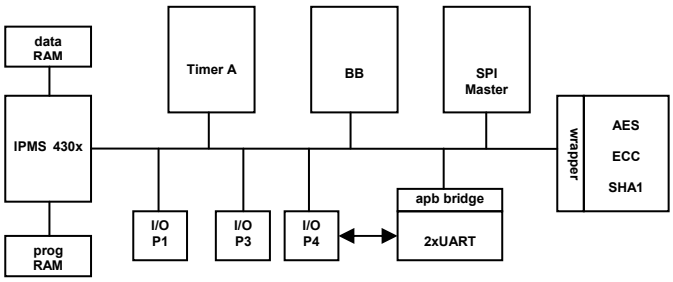

Figure 1. Crypto Processor Architecture

A. Processor Core

IPMS430x is an asynchronous 16-bit processor core fully compatible to Texas Instruments MSP430x architecture. The core is a latch-based design that applies poly-phase clocking in its control unit. The processor supports up to 1MB of shared program and data memory (20-bit address bus) and implements interrupt controller logic.

B. Digital IO Ports

The system implements three 8-bit digital IO ports (P1, P3 and P4) that are fully compatible to TI implementation of digital IO in MSP430x family of processors. Port P1 provides interrupt capability and can be programmed to generate interrupts either on rising or falling edge of input signal. Port P2 is not implemented. Ports P3 and P4 are without interrupt capability. All IO Port signals can be individually programmed to connect to external ports or to internal signals. We used this feature to connect one of the UARTs through P4 in order to reduce pad number.

C. UARTs

Two asynchronous UARTs (TI 16550 compatible) are provided for serial communications. The UARTs support 5 to 8 data bits, one optional parity bit and one or two stop bits. To generate the bit-rate, the both UARTs include a programmable 12-bits clock divider. The hardware flow-control is supported through RTS/CTS handshake signals. Error-detection in the receiver detects parity, framing, break and overrun errors.

D. Timer

The implemented timer is fully compatible to 'Timer A' specification of TI MSP430x architecture. It's a 16-bit timer/counter with three capture/compare registers. The timer supports multiple capture/compare, PWM (pulse width modulated) outputs and interval timing. It provides an interrupt generation when the counter makes an overflow or when capture/compare event occurs. The capture unit supports both synchronous and asynchronous capture, where capture event is selectable between rising, falling or both edges. The timer supports up to three different external clock sources and it can be programmed to use system clock as well. Additionally, the input clock can be divided with ratio of 2, 4 or 8. When the timer is off, the internal clock is stopped.

E. SPI

The SPI master core is to be used for control of external radio modules or to attach sensor logic. The core supports active edge selection for slave select signals and provides an option for burst mode operation. When operating in burst mode, SPI requires that one of IO port signals provides the function of slave select signal. The SPI also implements the clock prescaler having programmable division ratios from 1:4 to 1:32.

F. Baseband Controller

The implemented baseband controller complies with the DIN EN 13757-4 standard. EN 13757-4 is a communication standard for meters and remote reading of meters specified for short range devices running at frequencies 868-870MHz with data rates ranging from 2.4 to 66.6 kb/s. The baseband supports 3-out-of-6 and Manchester coding defined by standard and is extended to support DSSS (direct sequence spread spectrum) to improve communication in highly jammed environment (Barker 7 and Barker 11 modulation). Additionally, clock gating and wake-up support is implemented for maximum power efficiency at low duty cycle operation. Baseband can process input data stream with clock offset higher that 2% and compensate it during frame receiving. That can improve communication with devices, which have large clock drift caused by any reason.

G. Crypto Cores

To support both symmetric and asymmetric cryptographic operations we implemented hardware accelerators for AES and ECC, respectively. Additionally, to provide the means for user authentification and data integrity, we implemented the hardware accelerator for SHA-1 hash algorithm. All crypto cores are 32-bit wide and they share a single interrupt signal to the processor. A wrapper has been designed to provide 16-to-32 bit communication between processor and crypto logic. The crypto logic implements clock gating to put up the cores into the sleep mode when those are inactive.

1) AES: The Advanced Encryption Standard (AES) is the replacement for the insecure DES-Algorithm. It was standardized in 2001 by the NIST. It is a symmetric cipher algorithm, which uses the same key for encryption and decryption. In our implementation, AES has key length of 256 bits with 4-bit address and 32-bit data memory-like interface

to the processor. The performance improvement of the hardware-based AES over software implementation is given in [9].

2) ECC: ECC stands for Elliptic Curve Cryptography and it is an asymmetric cipher algorithm that uses algebraic operations on elliptic curves (EC) over finite fields. System's security is based on the complexity of finding the discrete logarithm of a random elliptic curve element. Due to the higher difficulty of calculating a discrete logarithm for points of elliptic curves, the algorithm is able to provide a very high level of security even with relatively short key sizes. For example, the level of security provided by a 1024 bit RSA key can be achieved by a 160 bit EC key. In our ECC implementation, the elliptic curve point multiplication (ECPM) is performed by the Lopez-Dahab algorithm. The control unit also manages the access to the eight 233 bit registers. One of these registers can additionally be written from the external bus. The ALU combines the functionalities of addition, squaring and allows bit manipulations. The multiplier is an Iterative Karatsuba Multiplier and it requires 9 cycles for each 233 bit operation. The core is optimized for low power and does not support operation on keys with sizes other than 233. The details of the implemented ECC core are given in [10].

3) SHA-1: SHA-1 (Secure Hash Algorithm) is the most widely used cryptographic hash function of the existing SHA hash functions, and is employed in several widely-used security applications and protocols. SHA-1 produces a 160-bit message digest based on principles similar to those used in MD4 and MD5 message digest algorithms, but has a more conservative design.

H. Memory Subsystem

The IPMS430x provides up to 1MB of available data/program memory space (von-Neumann architecture). The memory is byte-organized. We implemented 16kB of data RAM and an interface to external Flash. Additionally, we integrated a 2kB RAM block that can be used as a boot memory. This RAM block is also used for debugging purposes.

IV. Implementation

The chip has been implemented in a standard design flow (RTL, synthesis, layout) where the functionality has been proven after each implementation step. The chip has been designed for IHP 0.25um CMOS technology. The results of power analysis have been presented.

A. RTL

Many of the details related to RTL implementation of the asynchronous IPMS430x processor were already given in [11]. Therefore, they will not be discussed here. We will only emphasize the most important differences in the implementation approach used for this chip. Unlike our previous work, in this chip we implemented explicit synchronization of all preset/clear signals avoiding the need for glitch filtering. Also, the peripheral clock generation logic has been modified. Now, the clocks are generated only during write access to a peripheral. Those changes provided an additional robustness of the design during later implementation steps.

B. Synthesis

We performed the bottom-up synthesis of the design using Synopsys Design Compiler. The cell area of the complete chip was estimated to 5mm^2, and for crypto cores 2mm^2. The estimated power of the chip was 8.1mW and for crypto cores around 0.8mW. The power results are based on the static analysis and can only serve as an indicator to overall system parameters and relation between different components. The system has been synthesized for target frequency of 20 MHz.

C. Layout

The layout of the chip is implemented with Cadence SOC Encounter tool. Total area of the chip is 15.75mm^2 (pad-limited). The chip contains 104 pads (80 signal pads, 24 power/ground) and it fits into a 128-pin package. The chip is currently in the production. The layout photo is given in Figure 2.

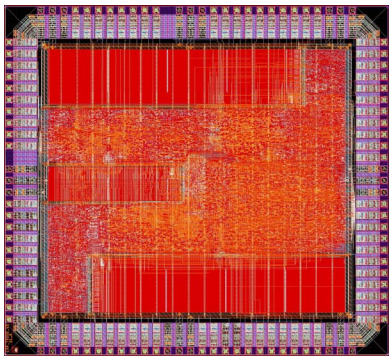

Figure 2. Crypto Processor Layout

D. Design Verification

The functionality of the chip has been tested thoroughly after each implementation step. We've created separate testbenches targeting each specific function of the chip. The tests were implemented using three different test approaches. The first approach, used to test processor itself, was to emulate debug functionality in the testbench by feeding the processor with instructions via I2C debug port. Thus, we've tested the complete instruction-set of the processor. The second approach, used to test peripherals, was to execute a specific program from the external Flash model. The tests have been created to cover the majority of peripheral functions and to generate reports for all the relevant test information. The third approach was to use I2C debug port to load internal memory with the test program. After the load process is completed, a program is executed from the memory. We used the same programs as for testing peripherals.

V. Results

A. PowerEstimation

The initial power estimation performed after synthesis was based on the premise that 20% of input ports switch at the

time. In this static analysis, the tool propagates specified switching activity through the design in order to estimate the total power consumption. This approach gives satisfactory results only with purely synchronous designs. In the case of an asynchronous design, the signals can not be accurately propagated, resulting in the results that are generally too optimistic.

Too get an accurate power estimation of the chip we performed after-layout vector-based dynamic power estimation using Cadence SoC Encounter Power System. This approach uses cell activity saved during post-layout simulation to annotate switching power of the complete chip. The layout model of the chip contains extracted parasitic values for all the cells and wires in the design. Also, the effect of voltage drop in the supply lines is modelled. The switching activity vectors have been profiled in order to determine simulation ranges with maximum activity. Then the power analysis for the specific range is performed. The results for power consumption estimated for typical conditions (Vdd=2.5V, T=25C, f=12.5MHz) are summarized in Table I.

TABLE I. POST-LAYOUT POWER ESTIMATION

Chip Function	Total Power (mW)
SHA-1 Calculation	24.04
AES	27.66
ECC Point Multiplication	53.49
ECC First Point Inversion	46.64
ECC Second Point Inversion	47.27
Tx Mode	11.98
Rx Mode	11.90
Timer On, CPU Off	3.755
Timer Off, CPU Off	3.647

B. Result Analysis

To analyze the power, we compared ECC performance of our chip with different software implementations. The results are summarized in Table II.

TABLE II. POWER FOR DIFFERENT ECC IMPLEMENTATIONS

CPU	Key Size (bit)	Frequency (MHz)	Energy (mWs)	Comment
MSP430	163	16	340.3	[12]
MSP430	283	16	831.2	[12]
MSP430	571	16	833.2	[12]
MSP430 (periph. sleep)	163	16	180.7	[12]
MSP430 (periph. sleep)	283	16	441.5	[12]
MSP430 (periph. sleep)	571	16	442.6	[12]
MSP430/FPGA	571	16	0.831	[12]
MIPS R4000	163	33	7.0	[10]
Atmel ATmega128	163	7.4	120	[10]
Motorola Dragonball	163	16	135	[10]
StrongARM	163	206	3.6	[10]
Pentium II	163	400	90	[10]
MIPS 4kEP	233	33	16.49	[10]
IHP CRYPTO	233	12.5	0.055	our chip

In [12] authors presented the software implementation of ECC on TI MSP430F1611 microcontroller and compared it to combined processor/FPGA solution, where FPGA is used for ECC computation and the processor for data feed. The reported power improvement of FPGA solution over software is around 1000x. Our system shows improvement of 6000x compared to TI MSP430 software implementation and accordingly 6x improvement over presented FPGA solution.

VI. CONCLUSION

In this paper we presented a hardware solution to the emerging problem of data security in wireless sensor networks. To support maximum security we designed a processor that supports both symmetric and asymmetric cryptography as well as the hash generation. The implemented features provide very high level of security at the cost of additional chip area and power consumption compared to the traditional implementation. However, for many high-demanding applications this kind of trade-off is inevitable. We've shown that with presented approach that applies some unique architectural solutions, we were able to design a processor that provides supreme security features maintaining the power consumption at acceptable level.

REFERENCES

[1] S. Peter, M. Zessack, F. Vater, G. Panic, H. Frankenfeldt and M. Methfessel, "An Encryption-Enabled Network Protocol Accelerator, " 6th International Conference on Wired/Wireless Internet Communications (WWIC 2008), May 28-30, 2008 - Tampere, Finland

[2] R. R. Brooks, B. Pillai, M. Pirretti, and M. C. Weigle, "Multicast encryption infrastructure for security in sensor networks, " International Journal of Distributed Sensor Networks, vol. 5, no.2, pp. 139–157, 2009.

[3] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-Bit CPUs," in Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems (CHES '04), vol. 3156, pp. 119–132, 2004.

[4] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," Wireless Networks, vol. 8, no. 5, pp. 521–534, 2002.

[5] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04), pp. 162–175, Baltimore, Md, USA, November 2004.

[6] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," in Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN 2007 '07), pp. 479–488, April 2007.

[7] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," Communications of the ACM, vol. 47, no. 6, pp. 53–57, 2004.

[8] IPMS430x: http://www.ipms.fraunhofer.de/

[9] F. Vater and P. Langendörfer, An Area Efficient Realisation of AES for Wireless Devices, it - Information Technology 2007, 188-193

[10] S.Peter, "Evaluation of Design Alternatives for flexible Elliptic Curve Hardware Accelerators, Diploma Thesis," BTU Cottbus, 2006

[11] G. Panić, T. Basmer, K. Tittelbach-Helmrich, L. Lopazianski, "Low Power Sensor Node Architecture," 17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2010, pp. 914-917

[12] J. Portilla, J. Andrés Otero, E. de la Torre, T. Riesgo, O. Stecklina, S. Peter, P. Langendörfer, "Adaptable Security in Wireless Sensor Networks by Using Reconfigurable ECC Hardware Coprocessors," International Journal of Distributed Sensor Networks, Vol. 2010