Journal of
Low Power Electronics
Vol. 2, 18–26, 2006

Implementation Strategy and Results of an Energy-Aware System-on-Chip for 5 GHz WLAN Applications

Labros Bisdounis,[1,*] Spyridon Blionas,[1] Enrico Macii,[2] Spiridon Nikolaidis,[3] and Roberto Zafalon[4]

[1] *Emerging Technologies and Markets Department, INTRACOM S.A., GR-19002 Peania, Athens, Greece*
[2] *Dipartimento di Automatica e Informatica, Politecnico di Torino, I-10129 Torino, Italy*
[3] *Department of Physics, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece*
[4] *Advanced System Technology, STMicroelectronics, I-20041 Agrate Brianza, Milan, Italy*

In this paper we present the implementation strategy and results of an energy-aware system-on-chip (SoC) that covers the baseband processing as well as the medium access control and data link control functionalities of a 5 GHz wireless system. It is compliant with the HIPERLAN/2 standard, but it also implements critical functionality of the IEEE 802.11a standard. Two embedded processor cores, dedicated hardware, on-chip memory elements, as well as advanced bus architectures and peripheral interfaces were carefully combined and optimized for the targeted application, resulting in a proper trade-off of silicon area, flexibility and power consumption. A system-level low-power design methodology has been used, due to the fact that power consumption is the most critical parameter in electronic portable system design. The 17.5 million-transistor solution was implemented in a 0.18 micron CMOS process and performs baseband processing at data rates up to 54 Mbit/s, with average power consumption of about 550 mW.

Keywords: Wireless Communications, Low-Power Electronic Systems, Embedded Systems and Processors, Hardware-Software Co-Design, System-on-Chip, VLSI, Memory Partitioning, Bus Encoding.

1. INTRODUCTION

In wireless data communications there have been many standardization efforts in order to meet the increased needs of users and applications. In the 5 GHz band, there were the IEEE 802.1a[1] and the HIPERLAN/2,[2,3] both specified to provide data rates up to 54 Mbps for wireless LAN applications in indoor and outdoor environments. Both aforementioned standards operate in the same frequency band, and utilize orthogonal frequency division multiplexing (OFDM) for multicarrier transmission.[4]

The purpose of this paper is to present the realization of a low-power SoC that implements the baseband processing as well as the medium access control and data link control functionalities of a 5 GHz wireless LAN system. Wireless communication systems require optimization of different factors including real-time performance, area, power, flexibility, and time-to-market. In order to optimize the combination of the above factors, instruction-set processors, custom hardware blocks as well as low-power memory and bus interface synthesis and mapping techniques are followed, offering a good balance between flexibility and implementation efficiency. The evolving scenario has serious consequences for any SoC development to be used in wireless systems. The protocol processor (running the upper protocol's layers) is included to the implemented SoC on the contrary to previous ASIC implementations[5,6] in which an external protocol processor should be used. Additional advantages of the implemented architecture are the adopted low-power design methodology, and the flexibility inserted by using an additional embedded processor core for controlling the baseband modem and implementing the lower-MAC processing of the HIPERLAN/2 standard, and a custom processor (MAC hardware accelerator) for implementing the lower MAC processing of the IEEE 802.11a standard.[7] During the development of the SoC,

*Author to whom correspondence should be addressed.
Email: lmpi@intracom.gr

 1546-1998/2006/2/018/009/$17.00+.25 doi:10.1166/jolpe.2006.003

power optimization techniques were applied in the whole design range of the system, starting from the embedded software and the hardware–software mapping until the memory and bus interface synthesis.[7]

The system-level design, the architectural choices of the implemented SoC, as well as the validation methodology have been presented in detail in Refs. [7–9]. In this paper we present an overview of the used implementation methodology, while our emphasis is on the presentation of the implementation results.

The paper is organized as follows: Section 2 presents the high-level modeling (system-level design) and the architecture of the SoC. Then, in Section 3 the low-power design methodological steps are described. The development of the embedded software implementing the processes of the protocol's upper layers, the lower MAC protocol processes and the specific hardware drivers, is presented in Section 4. Finally, details about the SoC implementation and validation are given in Section 5, and we conclude in Section 6.

2. SYSTEM-LEVEL DESIGN AND SoC ARCHITECTURE

The major problem in the design of a complicated system, such as a wireless LAN SoC, is the verification of the design and the early detection of design faults. For this reason a UML-based flow for the system design was followed in order to produce an executable system specification (virtual prototype) for early verification. This virtual prototype is based on a UML model, which uses the UML-RT profile.[10] The structure of the UML-based system model is presented in Figure 1. The illustrated model regards the access point (AP) device of a WLAN system, while the model of the mobile terminal (MT) device does not include the scheduler entity and it is quite similar to the access point model.

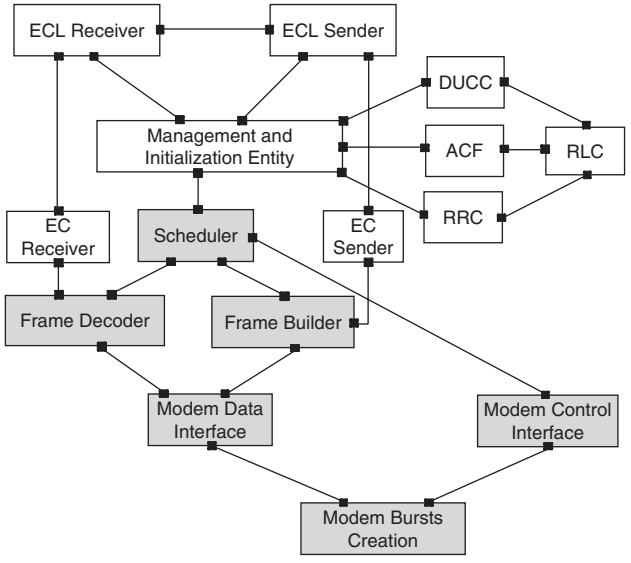

Fig. 1. Object structure of the HIPERLAN/2 high-level model.

Within the structure of the model there is a group of three objects that are mainly related to the interface of the modem (physical layer of the protocol stack) with the rest system. The first object models the behavior of the data manipulation parts of the modem's interface, the second object models the behavior of the control parts of the interface, while the third object models the behavior of the modem (creation of broadcast, downlink and uplink data bursts). The remaining objects regard the upper layers of the HIPERLAN/2 protocol stack[2] (Ethernet convergence layer, initialization and management entities, control, and data transport functions) and a custom implementation of the lower-MAC layer (scheduler, frame decoder, and frame builder). The last three objects are responsible for the interface of the protocol upper layers with the modem. The scheduler is the part of the interface that shares the resources of the physical layer among the various DLC connections, producing the format of each protocol frame. Then, the format of the frame is translated into modem commands, stored inside the modem control interface object. The frame builder is responsible for the creation of the downlink bursts of the frame. Its input is the format of the downlink burst produced by the scheduler, and its operation is to collect transmission packets from the DLC queues and format them accordingly, transferring the resulting parts of the frame to the modem data interface object. The operation of the decoder object is the equivalent for the uplink parts of the frame.

After the modeling of our system at the high level of abstraction, the production of an executable specification and the early system verification, we considered the alternative architectural options. In fact, two alternative architectures have been studied for the target SoC. Both architectures had a powerful instruction-set processor mainly in order the upper layers of the protocol to run properly. The two alternatives concern the implementation way of the lower-MAC controller and the baseband modem's controller. The first choice was to realize both controllers in software running on an additional instruction-set processor and the second was to design a dedicated hardware unit. The first solution has been selected mainly due to the fact that it is more flexible to modifications. The decision for a dual-processor architecture was based on technical criteria related to the specific system's requirements and parameters as well as on criteria related to the reusability and flexibility of the architecture in order to cover both WLAN standards, but also upcoming ones.

There are many alternative solutions for choosing an embedded microprocessor. For our architecture ARM, ARC, and MIPS cores were investigated. Choosing the right core is a tricky procedure that should take into account the target performance, the offered support to the design team, and the cost. Our choice for the protocol

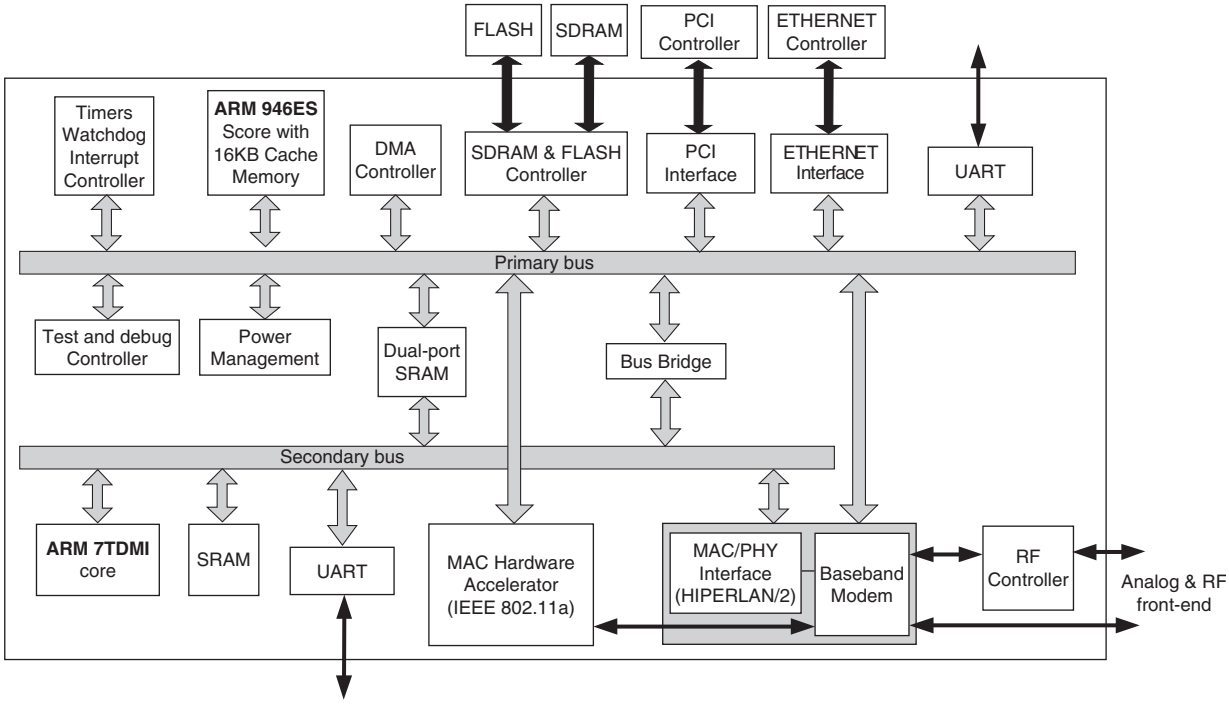

Fig. 2. Architectural template of the SoC.

processor was the ARM946ES core,[11] while for the control tasks we choose the ARM7TDMI core.[12] These processors exhibit small size, meet our needs in terms of performance and power consumption and reduce the system memory requirements and cost. Also, they benefit from a wide range of application software, operating systems, and development tools support. The support of the Advanced Microcontroller Bus Architecture (AMBA)[13] by the ARM cores was an additional reason for their adoption to our SoC architecture.

Thus, after the processing and balancing of the pros and cons of the architecture alternatives that derive in terms of technical and marketing issues, the final system architecture shown in Figure 2 was defined. The SoC is based on a memory-mapped architecture with a primary (AMBA AHB bus) and a secondary bus (interfaced through a bus bridge). The main macrocells of the chip are: protocol processor (ARM946ES core with 16 KB Cache memory along with ARM-related blocks), PCI and Ethernet interfaces for the mobile terminal and access point devices respectively, SDRAM and Flash controller, internal SRAMs (single and dual port), DMA controller, test and debug controller, power management unit (accounting for the special circuitry for clock gating and supply shutdown), two UART I/O serial interfaces, dedicated ARM7TDMI core (for controlling the baseband modem and support critical lower-MAC processes), MAC hardware accelerator (supporting critical lower-MAC functionality of the IEEE 802.11a standard such as encryption/decryption, fragmentation, timing control, protocol medium access),[1] dedicated block for the interface of the MAC and PHY processing elements (for the case of the HIPERLAN/2 standard), and baseband modem implementing the digital part of the physical layer of both HIPERLAN/2 and IEEE 802.11a standards.[1,3]

3. LOW-POWER DESIGN METHODOLOGY

The applied low-power design methodology includes optimization techniques that affect both the embedded software and the architecture of the system. The techniques that affect the embedded software concern data transfer and storage optimizations of the system's specification, as well as instruction-level power modeling and optimization. The techniques that lead to an optimized architecture in terms of power consumption include a power-aware hardware–software mapping of the system's functionality, as well as low-power memory and bus interface synthesis.

In order to support the hardware–software mapping of the system, a profiling and energy estimation methodology and tool were developed.[14] The tool is based on a commercial instruction-set simulator, augmented with functional, timing and power models for peripherals and memory systems. It takes, as input, the executable specification (source code) of the target application, as well as power and timing models for various system components and it generates detailed profiling information, with aggregate performance and energy estimation for all functions in the source code. The profiling data are then used to identify the most energy and performance critical kernels in the application, in order to enable efficient mapping of the functional specification of the chip onto the target architectural template (Fig. 2).

We have evaluated the impact of the mapping approach on the power consumption of the baseband modem-related circuitry, where after the profiling we migrated the HIPERLAN/2 lower-MAC functions from hardware to software, and we compared the result with that produced by using an open-source partitioning tool in which performance and hardware resources are taken into account without any power parameter. The achieved power saving was 19%. Power estimations were taken by using the PowerChecker tool.[15]

The second stage of the applied low-power design methodology regards the preprocessing and pruning of the executable specification in order to extract useful information and produce a pruned version of the code, which will be used in the next optimization stage that regards data transfer and storage energy optimization.[16] At the preprocessing stage, the information concerning the algorithmic behavior was extracted, as well as the different tasks and their communication way, the used data types and the manner in which data are manipulated. During pruning some parts of the code were removed as they are consider unworthy of being optimized. In the second stage, a transformation-based methodology was applied to the pruned code guided by power and performance cost functions. The output is an optimized code in terms of memory accesses, which contributed in the reduction of the total power consumption of the system. In order to evaluate the impact of the applied software optimizations we performed an experiment, in which we estimated the power consumed by the execution of the software specification on the ARM946ES processor. The achieved power saving in comparison with the original executable specification was 18%.

In addition, a methodology and an instrumentation setup for measuring the instruction-level power consumption of an embedded processor were developed.[17,18] They are based on the measurement of the instantaneous current drawn by the processor in each clock cycle. The instrumentation setup for the accurate measurement of the processor current includes a current sensing circuit, a digital storage oscilloscope and a PC with data processing software. After continuous monitoring and measurement of the instantaneous current of the processor, we derived instruction-level power models. Based on these models a framework was developed for the estimation of the energy consumed by the software running on the ARM processors. The framework calculates the base and inter-instruction energy costs of a given program and takes into account all other factors influencing the total power consumed by the software. Estimations and analysis that were performed using the developed framework were used for the optimization of the lower-MAC processes running on the ARM7TDMI processor.

Based on existing memory partitioning techniques,[19] an automatic optimization methodology for on-chip memories was applied to the main memories of the SoC. According to the used methodology, after the mapping of the memory addresses range onto an on-chip SRAM and the analysis of its dynamic access profile, a multi-banked memory architecture is synthesized, optimally fitted to such profile.[19] The profile (obtained by instruction-level simulation) gives for each address in the range, the number of reads and writes to the memory during the execution of the target application. Assume for instance the profile of Figure 3, where a small subset of the addresses is accessed very frequently. A power-optimal partitioned memory organization is shown in the right part of Figure 3 that consists of three memory cuts and a memory selection block. The larger cuts contain the top and bottom part of the range, while 'hot' addresses are stored into a small memory. The average power in accessing the memory is decreased, because a large fraction of accesses is concentrated on a small memory, and the memory banks that are not accessed are disabled through chip select (CS). According to our system architecture (Fig. 2), there are in principle two internal memory structures on which memory partitioning was applied: a single-port 16 KB SRAM and a dual-port 120 KB SRAM. Table I gives the memory components of each memory after their partition made according to the analysis of the access profile. The power savings were about 22% for the case of the single-port SRAM, and 62% for the case of the dual-port SRAM. Power estimations were taken by using the PowerChecker tool.[15]

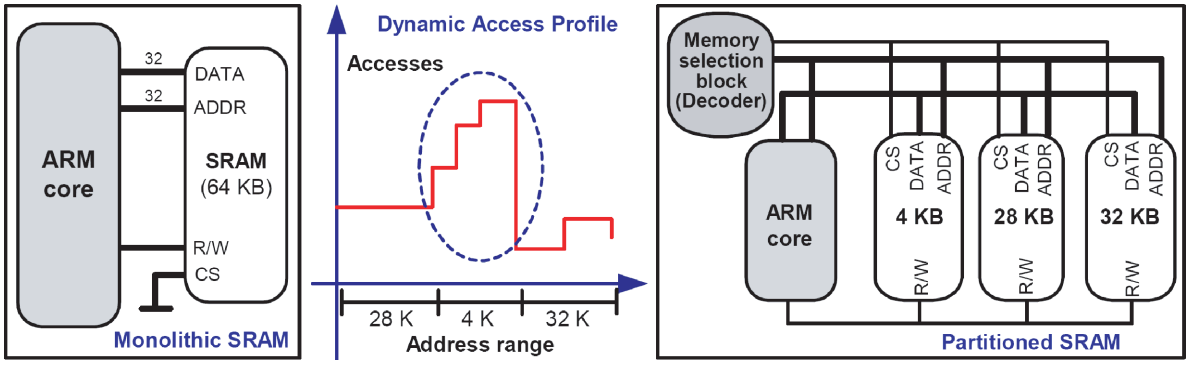

Fig. 3. Memory partitioning for power optimization.

Table I. On-chip memories partitioning.

Memory components	Components after partition
Single-port SRAM: 16 KB	Block 1: 3.4 KB
	Block 2: 12.6 KB
Dual-port SRAM: 120 KB	Block 1: 1 KB
	Block 2: 119 KB

Another method that was used to reduce the power consumption in the SoC was to apply existing data encoding techniques on the information transmitted on the buses. These techniques consist of modifying the way the binary words are represented. In order to apply bus-encoding techniques on the secondary bus of the SoC (see Fig. 2), an exploration tool has been developed, in which several of the existing encoding methods (bus invert,[20] gray,[21] zone[22] adaptive,[23] etc.) have been implemented in software. Data and address traces that have been obtained by profiling of the embedded application were used as input to the exploration tool. The output of the exploration tool is the savings in terms of power-consuming transitions number. Given a set of input bus traces, regarding addresses and data, the first step we have performed was to identify the most convenient encoding scheme using our exploration tool. After the exploration regarding the address secondary bus we found that significant power savings (33%) are achieved by applying gray coding,[21] while regarding the data bus we found that power savings up to 31% are achieved by applying bus-invert[20] coding. The derivation of the above savings took into account the energy consumed by the required encoders and decoders.

4. EMBEDDED SOFTWARE DEVELOPMENT

The software parts of the system include high-level protocol oriented processes, low-level protocol processes and system specific hardware drivers. The different parts of the system software can be categorized into two types: software processes and driver parts (Fig. 4). The development of the software processes has been performed at a high-level of abstraction, using UML as a modeling language,[10] while advanced code generation techniques were utilized for the production of C++ executable code. More specifically, a custom version of the inter-process communication mechanism was used, to replace the standard signal-send functionality offered by the UML library. This mechanism uses directly the specialized services offered by the operating system for sending signals, such as real-time mutexes. This custom way for inter-process communication was widely used in the integration of the processes inside the HIPERLAN/2 protocol stack, resulting in a significant improvement of the integrated code's performance.

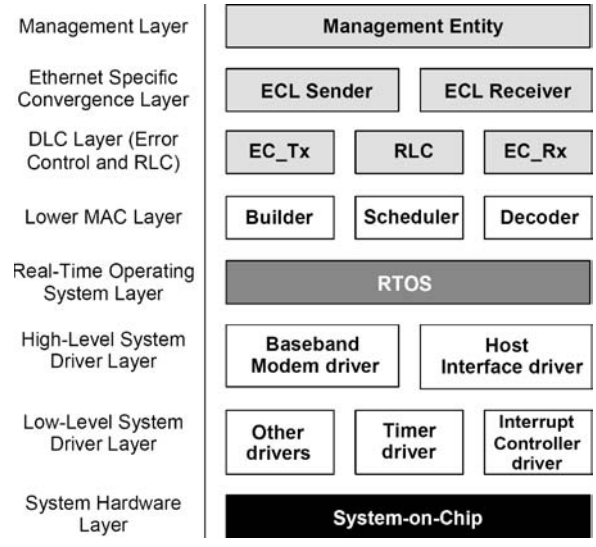

Fig. 4. Software architecture.

Table II. Mapping of the software processes.

ARM946ES	ARM7TDMI
ECL Sender	Scheduler
ECL Receiver	Frame builder
Management entity	Frame decoder
EC Transmit	Modem driver
EC Receive	Interrupt controller driver
RLC	Timer driver
Host interface driver	UART driver
Interrupt controller driver	
Timer driver	
DMA driver	

The porting procedure of the system's software to the implemented SoC consisted of two discrete phases. The first phase deals with the allocation of the various software processes and hardware drivers to the processor cores of the SoC, while the second phase deals with the creation of the proper execution environment for the system functionality. Based on the results of the mapping procedure (Section 2), as well as on exhaustive simulation using the UML-based high-level model of the system, a software-mapping scheme was derived and presented in Table II.

5. SoC IMPLEMENTATION AND VALIDATION

The chip is built on a dual-bus architecture (Fig. 2), each bus dealing with different layers of the protocol stack. This allows each layer to have all the necessary resources for data transfer and control without having to compete for a single-bus with other layers. Furthermore, as each bus has a processor core attached, the layers become independent in what regards processing and control resources. The two buses communicate through a dual-port RAM, where large data-blocks are placed by one bus for the other to read, and through a bridge that allows the upper bus to push (or pull) small mounts of control/data information to (or from) the lower bus. Another device that connects to both buses is the baseband modem. Since we want to transfer data between layers, the modem has ports to both buses. This

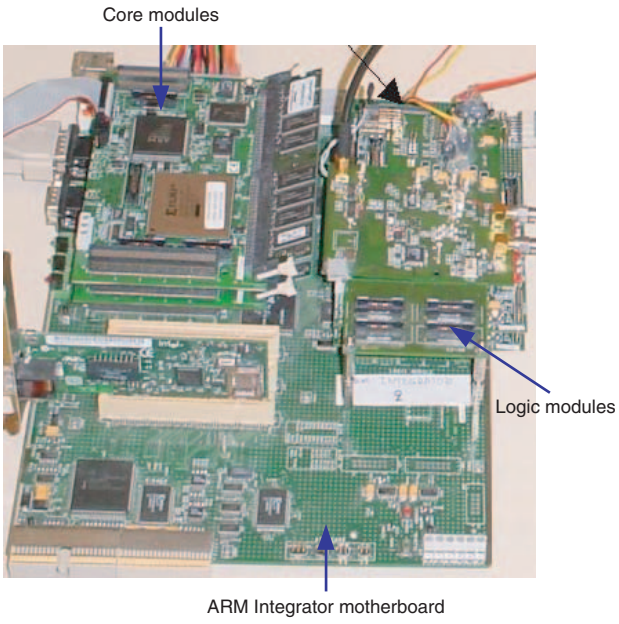

Fig. 5. Prototyping platform.

relieves the secondary bus from simply acting as a mediator between the modem and the upper protocol layers. The primary bus is the "protocol" bus because resources on this bus implement all the complicated classification and data transfer operations that are inherent to the protocol. The secondary bus is the "modem-control" bus. It deals with the lower layers of the protocol, acting mainly as the control and local data transfer resource for the baseband modem.

The followed low-level co-simulation procedure consisted of two main phases. The first phase exploited the advantage offered by a pure VHDL-based design and simulation environment by introducing a framework that converts the embedded core program into a set of force files for the VHDL-based simulation environment.[8,9] The second phase (FPGA-based design evaluation)[7–9] was based on the ARM Integrator platform (Fig. 5) that consists of a main motherboard implementing the system architecture, two core modules containing the required ARM cores with their peripherals, and two logic modules (hosting two XILINX Virtex E 2000 FPGAs that implement the rest logic). The average utilization of the two FPGAs was 87%. During the two validation phases, the system's behavior was validated by using several real-usage test scenarios.[9]

The physical design of the chip was performed using the Magma flow[24] and contained steps such as: pads selection, IO-padring design, package selection and bonding, floorplanning, place and route, formal verification, parasitics extraction, pre/post-layout static timing analysis, post-layout simulation, physical verification, DRC, and mask preparation. Implementation data for the chip are given in Table III. Figure 6a shows the placement of the chip's macrocells, while Figure 6b illustrates the fabricated die photograph. The power consumption of the chip was estimated by using the PowerChecker tool.[15] The estima-

Table III. Chip implementation data.

Process technology	0.18 μm CMOS
Supply voltage	1.8 V (core), 3.3 V (I/O pads)
Operating frequency	80 MHz (some modem's blocks in 40 MHz)
Average power consumption	554 mW
Equivalent gates count	4,400,000
Transistors count	>17,500,000
Pins count	456 (200 of them are for testing/ debugging purposes)
Packaging	BGA 35 mm × 35 mm
Chip area	9.408 mm × 9.408 mm ≈ 88.5 sq. mm
Core area	8.576 mm × 8.576 mm ≈ 73.5 sq. mm
Area occupied by logic	43 sq. mm
Area occupied by memories	30.5 sq. mm
Total length of interconnections	47 m (six metal layers)

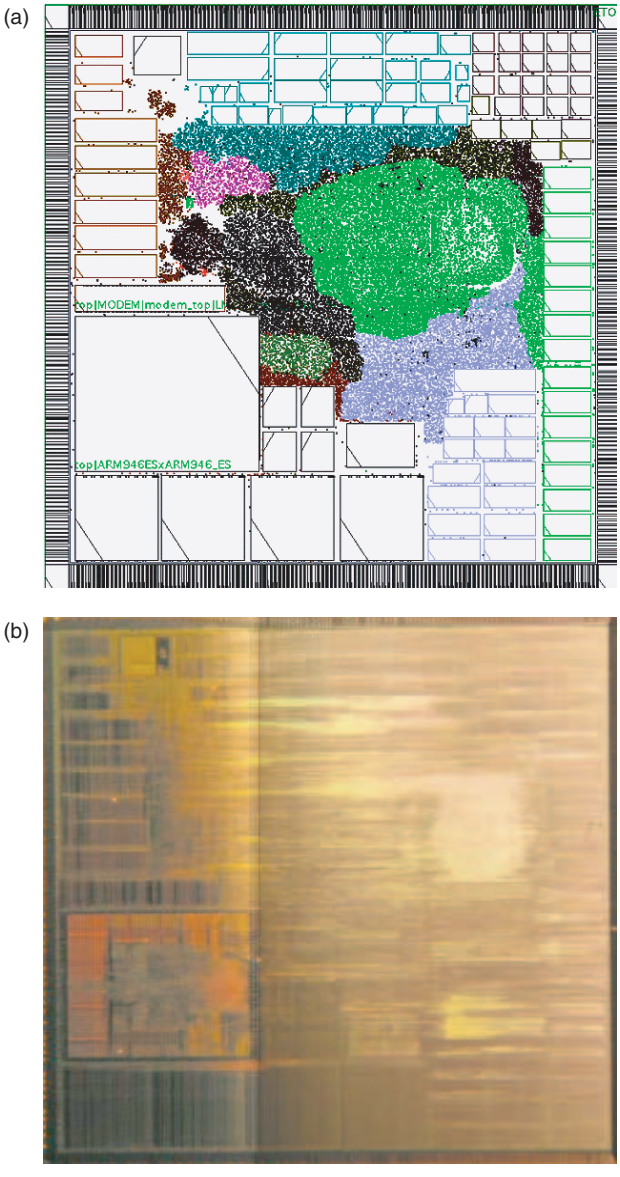

Fig. 6. (a) Placement of the chip's macrocells. (b) Photograph of the die.

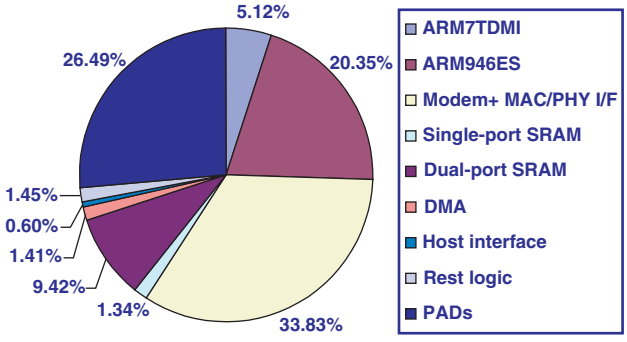

Fig. 7. Contribution of the system parts to the total power consumption.

tion was based on a testbench that includes an access point (AP) instance and a mobile terminal (MT) instance of the chip that exchanging data. The total estimated power for the AP instance was 460 mW, while for the MT instance was 480 mW. For the embedded processors, we used their nominal power consumption (mW/MHz) for the target technology.[11, 12] In Figure 7, we present the contribution of the main macrocells to the total chip power consumption.

Power measurements were taken after the placement of the chip in two reference boards (operating as AP and MT, respectively). The average power consumption of the chip hosted by the AP board was 545 mW, while the average power consumption of the chip hasted by the MT board was 563 mW. In comparison with the power estimations given above, there was a deviation of about 18%.

In order to demonstrate and validate the SoC operation, we developed a system of reference boards. The system contained three boards: the main board hosting the SoC and implementing the digital part of the protocol as well as the analog to digital and digital to analog conversions, the IF board and the RF board. The main board (Fig. 8) contains the following major items: Ethernet controller, SDRAM memory, Flash memories, A/D and D/A conversion and clock generation circuitry, power supply circuitry, various connectors for system's interconnection and test purposes, and of course the implemented SoC. The IF board is based on a monolithic CMOS fully differential IF

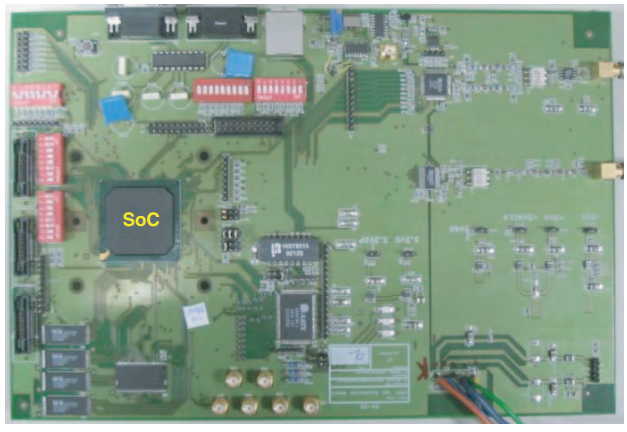

Fig. 8. Reference board hosting the implemented SoC.

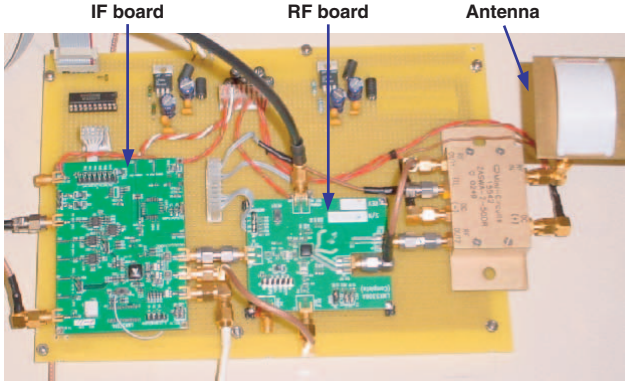

Fig. 9. IF and RF boards.

transceiver converting an OFDM signal between 880 MHz and 20 MHz, while the RF board is based on a monolithic SiGe RF front-end designed to transmit and receive OFDM signals in the 5 GHz band. Both boards are shown in Figure 9.

The target area of the final demonstration setup (Fig. 10) of the wireless system includes two reference board systems (one operating as AP and a second operating as MT) exchanging Ethernet data through 5 GHz RF connection. In a wireless system, the most important procedures are the association (between MT and AP) and the connection setup. During the association procedure, the MT by the time is switched on, associates with the AP. Information that is specific for the particular MT is transferred. The MT after its association is not ready to start sending and receiving data. The connection setup procedure assigns certain DLC connections to the particular MT for data exchanging purposes. In order to observe the execution of these two procedures by our system, a console application has been developed. This application runs to the monitoring PC (Fig. 10), the serial ports of which are connected to the serial ports of the reference board of each device. The console application receives messages from the protocol processor of each device through the serial connections. This application has the ability to illustrate system and status messages, as well as the number of exchanged Ethernet packets between the two devices

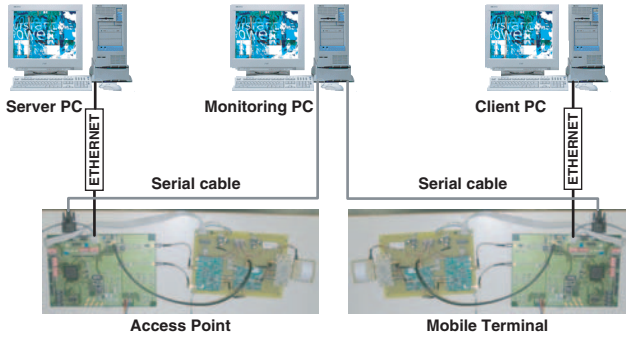

Fig. 10. Demonstration setup of the wireless system.

of the demonstration setup. Apart from the procedures of the wireless system, network applications such as ping and file transfer (FTP) were tested successfully. The first one demonstrated the fact that the one PC (client or server) of the demonstration setup (Fig. 10) is reachable by the other, while the second validated the correct operation of the wireless system since large amounts of data were transferred and the stability of the system was not affected.

6. CONCLUSION

In this paper, the implementation strategy and the results of an energy-aware system-on-chip for 5 GHz WLAN applications, has been presented. The SoC implements the baseband processing as well as the medium access control and data link control functionalities of a 5 GHz wireless system. It is compliant with the HIPERLAN/2 standard, but it also covers critical functionality of the IEEE 802.11a standard. The embedded software development, the hardware implementation and the application of power optimization techniques have been described. Finally, implementation data as well as power measurements and the followed validation strategy were presented. The implemented SoC meets the functional and timing requirements of the WLAN system, and offers a power-efficient and flexible solution.

References

1. IEEE, IEEE Std. 802.11a – Supplement to IEEE standard for information technology, WLAN Medium Access Control and Physical Layer in the 5 GHz band **(1999)**.
2. ETSI BRAN, HIPERLAN Type 2 – System overview, TR 101 683 **(2000)**.
3. ETSI BRAN, HIPERLAN Type 2 – Physical (PHY) layer, TS 101 475 **(2000)**.
4. R. Van Nee and R. Prasad, OFDM for mobile multimedia communications, Artech House, Boston **(1999)**.
5. J. Kneip, M. Weiss, W. Drescher, V. Aue, J. Strobel, T. Oberthur, M. Bole, G. Fettweis, Single chip programmable baseband ASSP for 5 GHz wireless LAN applications. *IEICE Transactions on Electronics* **(2002)**, Vol. 85, pp. 359–367.
6. W. Eberle, V. Derudder, G. Vanwijnsberghe, M. Vergara, L. Deneire, L. Van der Perre, M. G. E. Engels, I. Bolsens, and H. De Man, 80-Mbps QPSK and 72-Mbps 64-QAM flexible and scalable digital OFDM transceiver ASICs for wireless local area networks in the 5-GHz band. *IEEE Journal of Solid-State Circuits* **(2001)**, Vol. 36, pp. 1829–1838.
7. L. Bisdounis, C. Dre, S. Blionas, D. Metafas, A. Tatsaki, F. Ieromnimon, E. Macii, Ph. Rouzet, R. Zafalon, and L. Benini, Low-power system-on-chip architecture for wireless LANs. *IEE Proceedings on Computers and Digital Techniques* **(2004)**, Vol. 151, pp. 2–15.
8. S. Drosos, L. Bisdounis, D. Metafas, S. Blionas, A. Tatsaki, and G. Papadopoulos, Hardware–software design and validation framework for wireless LAN modems. *IEE Proceedings on Computers and Digital Techniques* **(2004)**, Vol. 151, pp. 173–182.
9. S. Drosos, L. Bisdounis, D. Metafas, S. Blionas, and A. Tatsaki, A multi-level validation methodology for wireless network applications. *Integrated Circuit and System Design: Power and Timing Modeling, Optimization and Simulation*, edited by E. Macii, V. Paliouras, and O. Koufopavlou, Lecture Notes in Computer Science Series, Springer-Verlag, Berlin **(2004)**, pp. 332–341.
10. G. Martin, UML for embedded systems specification and design: Motivation and overview. *Proceedings of the IEEE Design Automation and Testing in Europe Conference* **(2002)**, pp. 773–775.
11. ARM Ltd., ARM946-ES Thumb family, Technical reference manual **(2002)**.
12. ARM Ltd., ARM7TDMI, Technical reference manual **(2001)**.
13. ARM Ltd., AMBA specification, Technical reference manual **(2000)**.
14. F. Menichelli, M. Olivieri, L. Benini, M. Donno, and L. Bisdounis, A simulation-based power-aware architecture exploration of a multiprocessor system-on-chip design. *Proceedings of the IEEE Design, Automation and Test in Europe Conference* **(2004)**, Vol. 3, pp. 312–317.
15. BullDAST s.r.l., PowerChecker user manual, Version 4.0 **(2004)**.
16. A. Milidonis, G. Dimitroulakos, M. D. Galanis, A. P. Kakarountas, G. Theodoridis, C. Goutis, and F. Catthoor, A framework for data partitioning for C++ data-intensive applications. *Design Automation for Embedded Systems Journal* **(2004)**, Vol. 9, pp. 101–121.
17. S. Nikolaidis, N. Kavvadias, P. Neofotistos, K. Kosmatopoulos, T. Laopoulos, and L. Bisdounis, Instrumentation set-up for instruction-level power modeling. *Integrated Circuit Design: Power and Timing Modeling, Optimization and Simulation*, edited by B. Hochet, A. J. Acosta, and M. J. Bellido, Lecture Notes in Computer Science Series No. 2451, Springer-Verlag, Berlin **(2002)**, pp. 71–80.
18. S. Nikolaidis, N. Kavvadias, T. Laopoulos, L. Bisdounis, and S. Blionas, Instruction-level energy modeling for pipelined processors. *Integrated Circuit Design: Power and Timing Modeling, Optimization and Simulation*, edited by J. J. Chico and E. Macii, Lecture Notes in Computer Science Series No. 2799, Springer-Verlag, Berlin **(2003)**, pp. 279–288.
19. L. Benini, L. Macchiarulo, A. Macii, and M. Poncino, Layout-driven memory synthesis for embedded systems-on-chip. *IEEE Transactions on VLSI Systems* **(2002)**, Vol. 10, pp. 96–105.
20. M. R. Stan and W. P. Burleson, Bus-invert coding for low-power I/O. *IEEE Transactions on VLSI Systems* **(1995)**, Vol. 3, pp. 49–58.
21. C. L. Su, C. Y. Tsui, and A. M. Despain, Saving power in the control path of embedded processors. *IEEE Design and Test of Computers* **(1994)**, Vol. 11, pp. 24–30.
22. E. Musoll, T. Lang, and J. Cortadella, Working-zone encoding for reducing the energy in microprocessor address buses. *IEEE Transactions on VLSI Systems* **(1998)**, Vol. 6, pp. 568–572.
23. L. Benini, A. Macii, E. Macii, M. Poncino, and R. Scarsi, Architectures and synthesis algorithms for power-efficient bus interfaces. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **(2000)**, Vol. 19, pp. 969–980.
24. Magma Design Automation, Gain-based synthesis: Speeding RTL to silicon, White paper **(2002)**.

Labros Bisdounis

Labros Bisdounis received the Diploma and the Ph.D. degrees in Electrical Engineering both from the Department of Electrical and Computer Engineering, University of Patras, Greece, in 1993 and 1999, respectively. During his collaboration with the above-mentioned department he participated as a researcher in several EU projects related to VLSI design. Since 2000, he is with the Emerging Technologies and Markets Department (former Development Programmes Department) of INTRACOM S.A., Athens, Greece,

working on the design and development of VLSI circuits and embedded systems for wireless network applications, as well as a project manager in EU and national research projects. His main research interest is on various aspects of circuits and systems design such as: low-power and high-speed CMOS digital circuits and systems design, system-on-chip design for wireless telecom applications, circuit timing analysis, and circuit power dissipation modeling. Dr. Bisdounis has published more than 25 papers in international journals and conferences on these areas, and received more than 150 references. Also, he is a contributing author in a book on designing CMOS circuits with low power consumption published by Kluwer.

Spyridon Blionas

Spyridon Blionas is with INTRACOM S.A. working as a program manager in the Emerging Technologies and Markets Department. Previously, he was the leader of the Microelectronics Group of the same department, which carries out among other areas, research in the fields of system-on-chip and VLSI design for wireless communication and biotechnology applications. He holds a Ph.D. degree from the University of Patras on VLSI for digital signal processing (1990), a M.Sc. degree from the University of Athens in Telecommunication Electronics (1986), and a Diploma degree in Physics from the University of Athens (specialization in Electronics, 1983). The areas of his interest are: VLSI and system-on-chip design for wireless communication and biotechnology applications.

Enrico Macii

Enrico Macii is a Full Professor of Computer Engineering at Politecnico di Torino since 2001. Prior to that, he was an Associate Professor (from 1998 to 2001) and an Assistant Professor (from 1993 to 1998) at the same institution. From 1991 to 1995 he was also an Adjunct Faculty at the University of Colorado at Boulder. He holds a Dr. Eng. degree in Electrical Engineering from Politecnico di Torino in 1990, a Dr. Sc. degree in Computer Science from Università di Torino in 1991 and a Ph.D. degree in Computer Engineering from Politecnico di Torino in 1995. Enrico Macii has authored over 300 scientific publications, including 40 articles in IEEE and/or ACM Transactions, 40 articles in other international journals, 5 book chapters and over 200 papers in the proceedings of IEEE/ACM conferences. In 2004, he was the Editor of the "Ultra Low-Power Electronics and Design" book, published by Kluwer. He received the Best Paper Award for an article presented at the 1996 IEEE EURODAC conference. His research interests are in the design automation of digital circuits and systems, with particular emphasis on low-power design aspects. Enrico Macii is the Editor-in-Chief of the IEEE Transactions on CAD/ICAS for the term 2006–2007. Prior to that, he was an Associate Editor for the same journal (1997–2005) and an Associate Editor for the ACM Transactions on Design Automation of Electronic Systems (2000–2005). He was the Guest Editor of a Special Issue on Dynamic Power Management of IEEE Design and Test (in 2000), the Guest Co-Editor of a Special Issue on Low-Power Electronics and Design of the IEEE Transactions on VLSI Systems (in 2001), and the Guest Co-Editor of a Special Issue on Low-Power Design of Systems on Chip of the IEE Proceedings (in 2002). Enrico Macii was the Technical Program Co-Chair (in 1999) of the IEEE Alessandro Volta Memorial Workshop on Low Power Design, the Technical Program Co-Chair (in 2000) and the General Chair (in 2001) of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), the General Chair (in 2003) and the Technical Program Chair (in 2004) of the IEEE PATMOS Workshop. He serves on the Technical Program Committee of several ACM/IEEE conferences and workshops, including DAC: Design Automation Conference, DATE: Design Automation and Test in Europe, GLS-VLSI: Great-Lakes Symposium on VLSI, ISLPED: International Symposium on Low-Power Electronics and Design and PATMOS: Power, Area and Timing Modeling, Optimization and Simulation. He is also a member of the Executive Committee of ISLPED, DATE and PATMOS. Enrico Macii is a Senior Member of the IEEE and a Member of the Board of Governors of the IEEE Circuits and Systems Society (term ending in 2007).

Spiridon Nikolaidis

Spiridon Nikolaidis received the Diploma and Ph.D. degrees in Electrical Engineering from Patras University, Greece, in 1988 and 1994, respectively. Currently he is an assistant professor in the Department of Physics of the Aristotle University of Thessaloniki, Greece. His research interests include: CMOS circuits delay and power consumption modeling, power consumption modeling of processors, and design of application-specific, and reconfigurable processors. Dr. Nikolaidis has published more than 100 papers in international journals and conferences on these areas.

Roberto Zafalon

Roberto Zafalon is the head of the Competence Center for Low Power System Design at the Advanced System Technology group in STMicroelectronics, Agrate Brianza (Milano), Italy. Until 2000 he was in charge of the Advanced Power and Timing team in Central R&D—Design Platform group, contributing to corporate CAD projects for innovative design solutions and methodologies in the fields of timing and power-conscious design, high-level power modeling, optimization and signal-integrity analysis, for achieving the full industrial exploitation of DSM technologies for complex SoC designs. He has published over 50 articles in international journals, conferences, and books and he holds three European and three USA patents in the field of low power design and architectures. He held full tutorials on research topics related to Low power tools and methodologies at several distinguished IEEE international conferences, including International Symposium on Circuits and Systems (ISCAS 2000), Design Automation and Test in Europe (DATE 2001, 2004), Design Automation Conference (DAC 2001). Dr. Zafalon is a member of the technical committees of several international conferences, including IEEE DATE, DAC, PATMOS, and ISLPED.