Verification of an Embedded Sensor Node System-on-Chip

Goran Panić

Abstract - In this paper the verification methodology for an embedded low power sensor node system-on-chip design has been presented. A mixed-signal, power-gated, processor-based sensor node microcontroller has been implemented and verified. The chip implements a number of peripherals, several analog components and a flash memory for program storage. The paper describes applied verification methodology including simulation steps, power analysis and chip measurements.

Keywords – sensor node, low power, system-on-chip, system verification.

I. Introduction

Nowadays, advanced system-on-chips (SoCs) implement not only digital components, SRAMs and I/Os, but also analog components, sensors, passives and non-volatile memories integrated on chip [1]. As most modern SoCs are developed for use in battery-powered devices, the power is considered to be one of the most important design constraints. The power reduction in SoC is accomplished by implementation of advanced low power techniques, which introduce new challenges to design and verification methodology [2]. Low power SoCs support complex power management schemes that allow alteration of supply voltage or complete shut-off of inactive parts of the chip. Usually, a low power system is designed to support several power modes controlled by a power control unit. In a specific power mode, selected chip components are powered down or they are set to low voltage or retention state. Consequently, the verification of such systems is a challenging task, since the verification methodology must prove correct chip functionality with respect to all implemented power modes [3].

Additional complexity to SoC verification is introduced by the integration of analog components. The functionality of an analog circuit cannot be efficiently verified in a digital simulation. Even though, the behavior of analog circuits can be described in a mixed-signal language such as Verilog-AMS [4], which is supported by most verification tools, it is still required to perform full SPICE level simulations of an analog circuit in order to create a golden simulation model to be used as a reference to the behavioral model. A behavioral Verilog-AMS model might be applied in a digital-only simulation for specific tests, but at the cost of increased simulation time. However, designers often choose to test only the interface to an analog block by using a simplified verification model of the analog-to-digital interface.

The verification of a processor-based SoC requires thorough testing of processor's instruction set as well as the functional testing of peripherals and system bus. If a program memory, such as an embedded Flash, is integrated on chip, the verification methodology must prove correct functionality of memory erase, write and read tasks initiated from memory controller or a dedicated port. All system components, memory access and communication between processor and peripherals must be functionally verified.

Finally, a verification methodology must be created in a way that it provides all required inputs for efficient chip testing after the fabrication. During post-production measurements, the stimuli from a functional simulation are applied to the chip and the chip response is being recorded. The measurements results are then compared to the results of functional simulation.

This paper describes the verification methodology applied to the design of an embedded sensor node microcontroller. The designed sensor node SoC integrates a 16-bit RISC processor, multiple I/O peripherals, crypto cores, embedded Flash, SRAMs and analog components. It also implements advanced low power techniques such as power gating and frequency islands. The chip was fabricated and tested.

The rest of the paper is organized as follows: The Section II discusses basic tasks of design verification. The Section III gives details on the target design and Section IV describes applied verification methodology. Section V concludes the paper.

II. Design Verification

Design verification applies set of actions in order to prove physical, electrical and functional correctness of design. The design is verified for power, timing, physical correctness and functional behavior. Functional verification is a fundamental step in design verification, and it is tightly coupled with timing verification. Functional verification is typically done in a functional simulation. Most verification tools support direct RTL (register transfer level) simulations based on common hardware description languages (VHDL or Verilog) as well as more complex methodologies such as universal verification methodology (UVM) defined for simulations based on SystemVerilog, a hardware description and verification language that allows complex, assertion-based simulation scenarios [5]. The

Goran Panić is with IHP GmbH, Im Technologiepark 25, 15236 Frankfurt Oder, Germany, E-mail: panic@ihp-microelectronics.com

UVM provides a SystemVerilog base class library helping users to create well-structured testbenches with clear separation between code structures related to the verification environment and those related to the creation of stimuli [6]. It also allows portability of verification data, reusability of verification tests and enables good interoperability between multiple verification engineers and hardware designers [7].

Per definition, a SoC integrates a number of components such as processor or advanced peripherals on a single chip. Many integrated SoC components are pre-qualified IP-Cores (intellectual property cores) already proven in an in-depth stand-alone verification process. Therefore, it is common to divide the SoC verification problem to the verification of IPs and top-level verification. The IP-level verification requires full-coverage verification of an IP, based on a random simulation technique that includes coverage monitors and scoreboard checkers integrated in the testbench environment. The IP-testbench can be reused in the top-level simulation if desired. The top-level verification requires intensive verification of certain system-level aspects such as communication between components, power management control and bus performance. The top-level simulation usually includes a number of direct tests for peripherals as well.

The back-annotation includes functional verification of design after it has been synthesized or layouted. The post-synthesis verification proves design functionality with respect to timing, which is defined by signal delays in synthesized cell gates. The delays in wires are approximated by so called wire-load models that give statistical approximation of wire delays. In the post-layout verification the wire delays are calculated by RC extraction performed on place- and-routed design. The timing simulation is performed for multiple corners with respect to different power modes that are associated to specific supply voltages and clock frequencies. The quality of clock tree insertion is also verified in the post-layout simulation.

Power verification includes analysis of leakage and dynamic power, analysis of electromigration and signal integrity, checking of voltage drop, and analysis of rush currents and turn-on time. Early power analysis is performed early in the design flow and full analysis is done in the back-end flow. Physical verification searches for violations in design geometry and connectivity, and it is performed during the sign-off checks.

Finally, fabricated SoCs are tested and measured on wafer. The wafer tests include functional tests as well as structural tests that identify physical failures in the chip such as scan chain tests. The chip measurements also prove the compliance to electrical specifications for the chip.

III. Sensor Node Design

In focus of this paper is verification of a complex sensor node system on chip designed for sensor applications with strong security demands [8]. To better understand the complexity of the design to be verified, the architecture of the target system, as well as, the steps related to system design and implementation are briefly described.

A. System Architecture

A sensor node microcontroller, which architecture is shown in Fig. 1, has been designed to cope with the security problems in wireless sensor networks (WSN). The security challenges in WSN have been addressed by hardware implementation of several advanced crypto algorithms. The system integrates hardware accelerators for AES (advanced encryption standard), ECC (elliptic curve cryptography), and SHA-1 (secure hash algorithm). The core of the system is a 16-bit RISC processor compatible to Texas Instruments MSP430X architecture. Additionally, the chip integrates a baseband processor core supporting direct-sequence spread spectrum, a spread spectrum modulation technique used to reduce overall signal interference. The system implements 16 kB of RAM and 64 kB of non-volatile sector-erasable Flash enhanced with error correction capability (EDAC). The designed SoC integrates a number of peripherals, e.g. I/O digital ports, timers, serial ports and controllers for Flash and analog-to-digital converter (ADC). The clock distribution is provided by an integrated DCO and two external sources, one for slow and one for fast clock input. The integrated clock controller allows individual clock setup for each peripheral. The ADC is interfaced to an integrated preamplifier circuit that provides the connection to an external biomedical sensor (BMS). The power management control is maintained by the power controller unit. The system implements five power-gated islands allowing power shut-down of selected crypto cores, the baseband core and UART blocks. The system also supports global clock gating of all peripherals. The debug capability is provided through a dedicated I2C debug port.

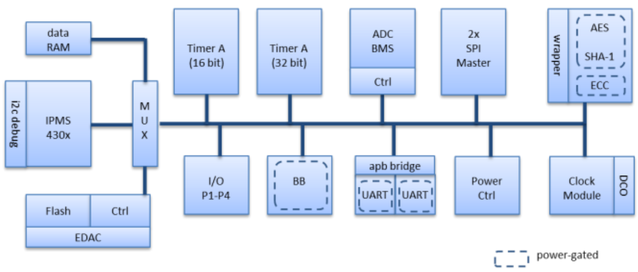

Fig. 1. Sensor node microcontroller

B. System Design and Implementation

The first step in the design of a sensor node microcontroller was system planning that included the exploration of architectures, algorithms and protocols related to the system application. The planning phase resulted in the decision on hardware/software partitioning and the definition of system architecture and power saving strategy for design.

The RTL design process included HDL (hardware description language) design and verification of system components and system itself. Some system components have been taken from existing IP libraries and some have been designed from scratch. The power control logic for selected power saving scheme was designed and verified in RTL as well.

The implementation steps followed the completion of RTL design for the system. The implementation flow started with the creation of power intent description file for the design. The power intent file contains a set of TCL commands, which describe power domains and supply voltages in the design. It also contains the information on low-power specific cells and libraries to be used in the implementation flow. The design power intent was described in CPF format (common power format) supported by the Cadence power-aware implementation tools. The CPF was also used in power-aware RTL simulations and back-annotations to verify correct behavior of the implemented power controller and to test the state switching between different system power modes.

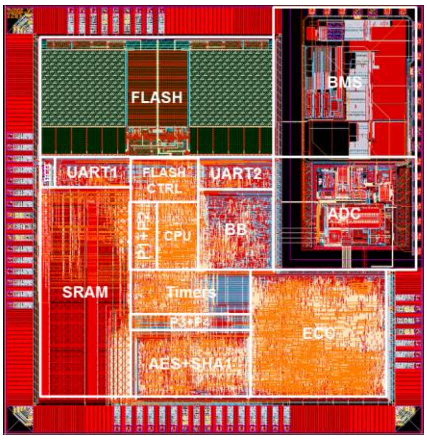

Fig. 2. Sensor node chip layout

The implementation steps included synthesis of the RTL design to physical gates and the backend implementation steps (floorplanning, clock tree insertion, place and route). The creation of power islands was performed during the floorplanning phase. The resulting layout of the system is shown in Fig. 2. The chip was implemented in IHP 0.25 um BiCMOS technology, and it runs with maximum frequency of 11.4 MHz. The estimated and measured power consumption of the chip was around 10 mW at 1 MHz.

IV. Verification Methodology

The verification of the designed SoC included functional and power verification during design process and the testing of fabricated chip. The functional verification included simulations of digital and analog parts and system level simulations. Before the system integration had started, each IP peripheral was thoroughly tested in a dedicated simulation environment. Only UARTs were tested by a UVM-enabled verification process using SystemVerilog verification models. All other IP-cores had their dedicated Verilog/VHDL simulation environment. The peripheral tests included exhaustive tests of IP functionality performed on the RTL-level. The analog components were tested in pure analog and mixed-signal simulations, including schematic simulations and SPICE simulations of the extracted layout netlists. The analog cores were also characterized for power and checked for design rule violations. The system-level tests were created to prove the overall system functionality with respect to timing and power constraints.

A. System Level Tests

The top-level simulation suite included a set of more than 30 test programs that execute from Flash. The test programs implement program routines designed to verify basic peripheral functionality and the functionality of whole system. In a simulation, the testbench loads the image of a test program to the Flash functional model and starts the execution. The results of test operations are then written to external ports and evaluated in the testbench. Additionally, the traces of signal changes during the simulation were captured in an EVCD file (extended value change dump), a standard industry-enabled file format for capturing simulation data, which was then used as an input for post-production tests. Since the Flash programmability was provided by the I2C debug port, a number of I2C-based tests were created to test the basic Flash operations, erase, write, read, and sector-based operations. For that purpose, a simulation model of I2C debug device was created and implemented in the test environment. The I2C-based simulations were used to simulate the instruction set of the processor and to create the program loading sequences for post-production tests. A simplified illustration of the applied verification environment is shown in Fig. 3.

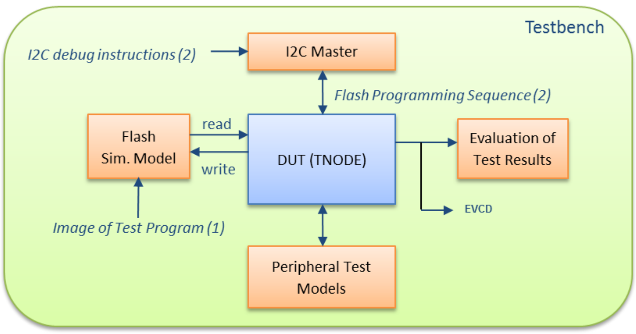

Fig. 3. Testbench environment: (1) tests executed from Flash; (2) tests executed via I2C debug port.

The behavior of power control logic was tested using dedicated test scenarios that forced controller to switch between different power modes of the chip. Those simulations were performed using power-aware capability of the Cadence simulation tool (Innovus) able to interpret design power intent information specified in a CPF file. The CPF-based simulations were designed to check for correct functionality of the power controller unit and to report eventual power state violations. The signal activity from the power-related simulations was captured in a full-scope EVCD file that is used when performing dynamic power simulations of the routed design. The results of power estimations for different case scenarios were compared in post-production to the power measurement results of fabricated chips.

B. Post-Production Tests

The post-production tests included functional wafer testing of fabricated microcontroller chips and power measurements. The tests were performed with an industrial tester device (Agilent Verigy 9300) capable for both wafer and package testing. The tester is constructed to operate with input stimuli extracted from EVCD files generated in simulations. A standard test procedure applies input stimuli to the device under test and samples the response on the chip output ports. The recorded output values are then compared to the expected values extracted from the same EVCD file. The measured sensor-node chip could not be stimulated externally, since it was designed to read the program data from an integrated Flash memory. Therefore, the test stimuli data had to be written to the Flash memory and executed from it. For that purpose a set of initial programming sequences that program the Flash via I2C debug port was created. The I2C programming sequence was erasing the Flash, writing the test program to it and reading the code back to check if it was correctly written. Following this approach, a complete set of test programs was sequentially stored and executed from Flash in a fully automatized test procedure.

The basic functionality of ADC was tested separately by applying predefined DC values to ADC inputs and reading back the converted data from the ADC internal registers. The extracted values were compared to the expected values obtained from analog simulations.

The chip power consumption was measured for representative test cases and compared to the power estimation results. Finally, the functioning chips were selected for packaging and further utilization.

V. CONCLUSION

The verification of a complex SoC is a challenging task requiring detailed planning of verification methodology with respect to the design process and post-production chip measurements. This paper described a verification approach in the design of a complex sensor node microcontroller SoC. The paper discussed main aspects of SoC verification and gave the details of the applied verification methodology. A special attention was given to the verification challenges introduced by implementation of low power techniques and integration of analog components in the system. The practical approach to the verification of a Flash-based system is described in detail.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Commission's Horizon 2020 under grant agreement from project myAirCoach – No. 643607.

REFERENCES

[1] Arden, W., Brillouët, M., Cogez, P., Graef, M., Huizing, B., & Mahnkopf, R., "*More-than-Moore white paper. Version 2"*, ITRS, 2010.

[2] Flynn, D., Aitken, R., Gibbons, A., & Shi, K., "*Low power methodology manual: for system-on-chip design"*, Springer Science & Business Media, 2007.

[3] Srikanth, J., Janick, B., Yoshio, I., & Flynn, D., "*Verification Methodology Manual for Low Power*," Synopsys, 2009.

[4] Kundert, K., & Zinke, O., "*The designer's guide to Verilog-AMS*," Springer Science & Business Media, 2006.

[5] Mintz, M., & Ekendahl, R., "*Hardware Verification with System Verilog: An Object-Oriented Framework (Vol. 230)*," Springer Science & Business Media, 2007.

[6] Accellera, "*UVM 1.0 Reference Manual*", 2011

[7] Yun, Y. N., Kim, J. B., Kim, N. D., & Min, B., "*Beyond UVM for practical SoC verification*", In SoC Design Conference (ISOCC), 2011 International (pp. 158-162). IEEE, 2011.

[8] Panić, G., Stecklina, O., & Stamenković, Z., "*An embedded sensor node microcontroller with crypto-processors*," Sensors, 16(5), 607, 2016