Adaptive Multi-Sensor Interface System-On-Chip

Jinwen Xi, Chao Yang, Andrew Mason and Peixin Zhong
Department of Electrical and Computer Engineering,
Michigan State University, East Lansing, U.S.A.
{xijinwen, yangchao, mason, pzhong}@egr.msu.edu

Abstract—This paper presents a system-on-chip using 0.18μm CMOS technology that integrates an 8-channel reconfigurable sensor interface with an 8-bit Σ-Δ A/D converter and a 16-bit sensor signal processor. The configurable sensor interface employs Wheatstone bridge and switched-capacitor topologies to efficiently perform signal conditioning for a wide range of resistive and capacitive sensors. The customized processor executes software for sensor interface configuration and calibration, data conversion control, sensor data calibration and compensation, and communication with external systems. On the 4x4mm chip with a 1.8V supply, the sensor readout circuit requires only 300μA and the processor power consumption is 209μW/MHz. The adaptive multi-sensor interface provides a platform solution with the flexibility crucial for implementing low-cost microsystems of the future.

I. INTRODUCTION

In ultra-miniature and low power multi-sensor systems, the level of integration and functional flexibility are essential considerations for platform design. The sensor interface circuitry (signal conditioning and processing) is a key component in the whole system. Low-cost smart sensor systems increasingly contain multiple sensors of different types, which require heterogeneous interfaces for pre-amplification and pre-filtering. Integrating a programmable processor and an adaptive sensor interface into one system-on-chip (SoC) combines analog and digital processing capability and permits rapid customization to different applications using the same integrated circuit. Single-chip integration of these components enhances performance while simultaneously reducing size, cost and power consumption.

This paper presents a sensor application SoC that integrates an 8-channel configurable sensor interface, an 8-bit Σ-Δ A/D converter (ADC) and a 16-bit sensor signal processor (SSP). The sensor interface employs Wheatstone bridge and switched-capacitor topologies to perform signal readout for resistive and capacitive sensors, providing a unified interface for heterogeneous mix of sensors. The customized SSP executes software to support interface configuration and calibration, ADC control, and sensor data processing. This multi-sensor interface can intelligently adapt to the readout and signal processing needs of many different sensor applications, providing flexibility that is crucial for the implementation of low-cost microsystems.

Recently there has been a lot of focus on intelligent sensory systems and universal sensor interfaces. In previous work [1,2] we have realized a generic multi-channel interface circuit that utilizes different circuit modules to readout different types of sensors. An SoC implementation of a fingerprint sensor with 32-bit RISC microprocessor and embedded memory has been presented [3]. A monolithic CMOS microplate-based gas sensor SoC has been proposed [4]. Compared with these efforts, the chip described here 1) provides a more highly configurable sensor interface for heterogeneous sensor types, and 2) utilizes a transaction-level design flow in the architectural design of the system, especially the digital sub-systems, to permit better system-level design and verification.

II. SYSTEM ARCHITECTURE AND DESIGN METHODOLOGY

Figure 1 depicts the system architecture of the proposed SoC, which is composed of three main functional blocks: Universal Sensor Interface (USI), A/D converter (ADC) and Sensor Signal Processor (SSP). The USI provides adaptive pre-amplification and pre-filtering of sensor-array signals by programming its control words via the SSP. Using the programmable sensor interface makes it more flexible when dealing with heterogeneous sensors in multi-sensor systems. The pre-conditioned sensor signal in the analog domain will be converted to digital format using an ADC featuring sigma-delta architecture. Both USI and ADC are controlled by SSP with 8-bit GPIOs mapped onto its memory addresses. The SSP is a 16-bit microprocessor featuring conventional Von-Neumann architecture with a simple 16-bit instruction set. The 16-bit address and data buses provide on-chip interconnection between the SSP and other functional blocks including on-chip RAM/ROM, multiplier, clock management unit, UART/SPI, and 8-bit GPIOs. One of the most important roles of the GPIOs is to provide controllability for the USI and ADC by setting their...
corresponding control bits. The motivation of employing SSP and configurable USI in this architecture is to provide flexible configuration of the multi-sensor system for tens of different types of sensors. The ADC is time-multiplexed for multiple sensor inputs. On-chip SRAM provides buffering for data calculations as well as program code storage, and UART/SPI offers the communication interface between the SoC with external devices or host PC for data post-processing.

A shared medium bus provides the interconnection between components. The programming of each functional block is mapped onto the central microprocessor’s 64KB address range. Some high performance SoC architectures with multi-processors, such as OMAP [5] and Nexperia [6], use dedicated bus-bridge for the translation between multibuses. However, the sensor applications targeted by this SoC do not require high data throughput. Instead, simplicity and power efficiency are more valued in this design. Thus the proposed architecture provides easy control of the configurable sensor interface with low power overhead.

The base system can be augmented with additional accelerators. For example, in this implementation, computation-intensive sensor applications can take advantage of an embedded 16-bit hardware multiplier that can perform a 16-bit MAC (multiplication and accumulation) operation in single clock cycle.

The design flow employed in the proposed SoC integrates standard Synthesis and Placement & Route (SPR) flow in the digital blocks with full custom design flow in the analog mixed-signal blocks. SPR flow automates most of the design process for the digital blocks using a vendor-specific standard-cell library, and it gives high fidelity of the success of the first-silicon. For analog parts, each transistor needs to be sized properly according to design specification, and there are no mature automated methodologies that are silicon-proven, so full custom design is applied. Using different design flows generates a challenge in verifying that the design process for the digital blocks using a vendor-specific standard-cell library, and it gives high fidelity of the success of the first-silicon. For analog parts, each transistor needs to be sized properly according to design specification, and there are no mature automated methodologies that are silicon-proven, so full custom design is applied. Using different design flows generates a challenge in verifying that the whole system integrates both digital and analog blocks with good coverage and fast simulation speed. Due to the design tool resource limit, full chip-level verification before signing-off is not performed in this design. Alternatively, digital and analog sub-systems are verified at the layout-level separately with their specific stimuli vectors before the final SoC integration is performed.

A. Sensor Signal Processor (SSP)

The target of the SSP is to provide maximum control to other functional blocks by running software code. Its 16-bit address bus provides the SSP’s 64KB address map, onto which all the digital peripherals are mapped as shown in Figure 2(a). The lowest 512 bytes of the 64KB address map are occupied by digital peripherals, in which the GPIO, UART/SPI, DCM are 8-bit peripherals, while the hardware multiplier is treated as 16-bit peripheral. The 16KB on-chip SRAM is partitioned into two blocks: Data SRAM (DSRAM: 0x0200–0x21FF) provides buffering for computation, and Program SRAM (PSRAM: 0xE000–0xFFFF) provides on-chip program code storage. After power-up, the on-chip boot loader fetches the program code from off-chip Flash/ROM to PSRAM, from which the code is executed. Using the boot loader and on-chip PSRAM reduces the signal transition on the external memory bus, and in turn reduces the dynamic power consumption.

The architecture of the SSP is presented in Figure 2(b), which describes the interconnection among the main functional blocks: ALU (Arithmetic Logic Unit), GPR (General Purpose Registers), IDU (Instruction Decoding Unit) and CU (Control Unit). Internal MUXes and DeMUXes manipulated by the control signals that are decoded by the IDU provide control over the data path. To reduce dynamic power consumption, clock gating is employed on the register file.

In SSP execution, each instruction is fetched from the code memory, which may be mapped on-chip or off-chip. It is then processed by IDU, which decodes the current instruction to the fields of op-code, addressing mode, access type (read/write) and the source/target registers. The IDU also calculates the absolute address by the addressing mode and decoded address, which are used to drive the address bus. The CU selects register files and set the corresponding signals for operation based on the decoded op-code. The data is fetched from the memory addressed by IDU, the GPR, or the ALU itself under the control of CU. Then the ALU will perform the specified operation before writing the results to the memory or register files.

B. System Level Design Methodology

It is widely believed that the earlier the software is developed along with the hardware, the more design time saving will be achieved by overlapping hardware and software design periods. To facilitate this hardware-software co-design methodology, SystemC [8] design flow was utilized for the SSP design and validation. SystemC is a system description language with a set of library routines and macros that extend C++, making it possible to model hardware, concurrency, and communication. The simulation is much faster than a typical VHDL/Verilog simulation. During execution, the modules can communicate in a simulated real-time environment, using signals of any data.
type (provided in C++, SystemC, or user-defined). The simulation in SystemC provides visibility into the system at an early stage that is useful in debugging and determining areas for improvements, whether the modifications are in hardware and/or software.

Shown in Figure 3, the SSP’s SystemC model was developed to allow for co-design of hardware and software for the digital part of the SoC. The SSP BF (Bus Function Model) comprises five functional modules: processor, on-chip/off-chip memory and other digital peripherals. The processor ISA is modeled as simplified cycle-based, while the processor bus transactions, which are of interest in this early model since they control other functional blocks, are modeled by cycle-accurate bus_write and bus_read functions. The SSP SystemC model provides an early execution platform for the software code, which is developed using ANSI C. To reduce the software development overhead by using the available GNU tool flows, the ISA of the SSP is fully compatible with the Texas Instruments MSP430, a widely used commercial processor. With this SSP SystemC platform, the application code can be compiled and executed to verify the correctness of the software before an FPGA prototype or real silicon is ready. Moreover, the SystemC model provides an executable reference model for the RTL model that will be synthesized to the netlist, which will be used to generate the final GDS-II layout.

Based on the SSP SystemC BFM reference model, the functionality-equivalent RTL model was designed and verified. The Xilinx Spartan-3 FPGA was used in the SSP design flow to provide a hardware emulation platform to verify the digital system’s timing and functionality. During FPGA emulation, the Block RAM (BRAM) in Spartan FPGA is used to map the on-chip memory blocks. Once the application code was compiled and linked, the output binary was converted into the BRAM map file where Block RAM was initialized in the specified format for the FPGA post-layout simulation and bit-stream generation. The generated bit-stream with both hardware and software designs was downloaded to the FPGA platform shown in Figure 4(a) for execution.

The final SSP RTL model was implemented using IBM 0.18μm standard cell design flow. Figure 4(b) shows the SSP layout with the size of 1.5mm × 1.8mm, where on-chip memory blocks take ~50% of the total silicon area. The SSP achieves the clock speed of 15.2MHz with active energy consumption of 209μW/MHz at 1.8V during gate-level simulation. In sensor applications, the SSP will operate at under 10MHz to lower power consumption. The resistive readout and capacitive readout shares hardware as much as possible. To achieve this, the capacitive and resistive information are converted to the uniform output and then processed with the shared hardware. For details of the sensor interface design are presented in [7].

D. Analog to Digital Converter

The overall design goal of the SoC is to achieve low power consumption and small system size with a moderate system speed. To meet these constraints, a Σ-Δ architecture that features low power, low hardware complexity and good robustness was chosen for the ADC. The ADC consists of five major function blocks shown in Figure 6: integrator, comparator, 1-bit sub-DAC, clock generator and digital counter. The switch-capacitor technique was employed to implement the ADC. To solve the charge injection issue in switch-capacitor circuits, a multiphase non-overlapping clock was used. The multiphase clocks are generated from an external main clock through an inverter-based delay chain.

The ADC design achieves the maximum sampling rate of 4KS/s with a maximum 2.5MHz oversampling rate. The input dynamic range was simulated to be 1.2V, and the maximum power consumption was determined to be 2.025mW with 1.8V power supply.
III. SENSOR SOC CMOS IMPLEMENTATION

The final multi-sensor SoC has been implemented using IBM 0.18μm mixed-signal CMOS technology. The 4.0mm × 4.0mm chip is shown in Figure 7(b) with functional blocks covered by top layer metal labeled in Figure 7(a). A second USI block (bottom left) not connected to the ADC was added to facilitate analog testing. A dedicated 1-to-2 MUX controlled by a pin is used to select which of the USI blocks the SSP’s control signals are passed to.

The test platform is shown in Figure 8. An external FPGA board is connected to the chip to provide the testing vectors. BRAM in the FPGA is configured to emulate the SoC’s external ROM. An RS-232 serial interface is also included in the test platform to communicate data with host PC. 8 LEDs connected to one of the SSP’s GPIO indicate the SoC’s current work status. The fabricated chip is currently under test.

IV. CONCLUSION

This paper presented a multi-sensor SoC integrating programmable processor, universal sensor interface and ADC as well as other on-chip resources. The USI utilizes a configurable architecture for resistive and capacitive sensors, and offers 0.33~15 and 0.035~31.75 gain ranges for resistive and capacitive sensors, respectively. By sharing hardware resource, the USI achieves a low power consumption of 300μA with a 1.8V supply. The on-chip ADC features a single-stage Σ-Δ architecture and 8-bit resolution with a power consumption of 2.02mW at a 2.5MHz oversampling rate. The USI and ADC, along with other functional blocks, are controlled by the SSP, which executes software to orchestrate the whole system’s operation. With simplified RISC architectural design, the SSP achieves 209μW/MHz power consumption at 1.8V supply. The SystemC-based hardware/software co-design flow employed in the SSP design enables early software development, shortening the overall system design time. The SoC has been fabricated in IBM 0.18μm CMOS technology and is currently under final functional test.

REFERENCES

[8] www.systemc.org