A Low-Energy Chip-Set for Wireless Intercom

M. Josie Ammer, Michael Sheets, Tufan Karalar, Mika Kuulusa, and Jan Rabaey
University of California, Berkeley
{mjammer, msheets, tufan, mkuulusa, jan}@eecs.berkeley.edu

ABSTRACT

A low power wireless intercom system is designed and implemented. Two fully-operational ASICs, integrating custom and commercial IP, implement the entire digital portion of the protocol stack. Combined, the chips consume 13 mW on average when three nodes are connected to the network. A high-level design methodology was used to define the protocol stack and communication algorithms, select architectures, and minimize energy.

Categories and Subject Descriptors

B.4.1 [**Input/Output and Data Communications**]: Data Communications Devices

General Terms

Measurement, Design.

Keywords

Design Methodology, Wireless Communication, Low Power.

1. INTRODUCTION

Great progress has been made in wireless communication over the past decade, enabling the proliferation of applications such as wireless Ethernet and cellular telephony. However, a wealth of applications can be enabled only through reduction of power consumption. These applications include voice communication in some specialized environments, environmental control in office buildings, and smart home applications such as security, identification, and personalization. The need for lower power consumption is acknowledged by the existence of several initiatives such as Bluetooth and ZigBee in the commercial sector and PicoRadio [2] in the academic sector.

While the traditional approach in low power design is to focus on optimizations at the circuit level, it is essential to begin at the system level to meet aggressive low power design goals. This enables the exploration of a larger design space to find a minimal power solution. New design methodologies that support high-level design exploration, integration of existing IP blocks, and refinement to implementation are required to realize these next generation ultra-low power wireless devices. This work covers the design of a low power wireless system for a voice intercom application including using a high-level design methodology to define the digital protocol stack and communication algorithms.

2. BACKGROUND

The Intercom protocol allows ad-hoc peer-to-peer communication of 64kbps uplink/downlink channels between 20 sensor/communicator nodes. This protocol supports voice-quality, intercom-style communication between groups of people who are physically close, but where ambient noise or other restrictions would prohibit a normal conversation.

The basic protocol uses TDMA to allow multiple nodes to access a single wireless channel. Any node can be configured to act as the basestation, which controls access to the slots in the TDMA frame using a session database. When a remote node requests communication with another remote node, the basestation assigns it a TDMA slot. Once a slot is assigned, the remote nodes can communicate in peer-to-peer fashion without further basestation intervention. Nodes save power by listening only to the basestation control and active communication slots.

The Intercom protocol stack consists of application, session, transport, data link (DLL), and physical (PHY) layers of the OSI reference model [5]. The application layer provides a user-interface consisting of buttons and an audio interface with optional compression. The session layer maintains a database of active sessions. The transport layer provides error recovery and automatic retransmission of lost control messages. The basestation DLL sends synchronization bits that the remote DLLs use to identify the TDMA frame. The DLL also pairs communication sessions with time slots, automatically listens during only the required slots, and transmits during the slot assigned by the basestation. Transmitted bits from the DLL are sent to the PHY transmitter, where they are modulated and sent over the air. Received data from the air is synchronized, demodulated and sent to the higher protocol layers.

As in most wireless devices, the functions at higher protocol layers are dominated by control flow operations, such as table lookups, while the lower levels are dominated by data path operations, such as filtering. Separate design flows are used for data path- and control-dominated components, allowing designers to work in the most appropriate environment using high-level modeling tools for design exploration and successive refinement.

A system-level analysis indicates a tradeoff between stricter digital requirements and reduced power consumption for the analog components in a wireless communication device [4]. The approach we take is to increase complexity of the digital portion so that a simplified low-power analog front-end can be eventually incorporated. This project considers only the digital sections of the physical layer and is divided into two custom ICs: Baseband Processor (BBP) and Wireless Protocol Processor (WPP), as shown in Figure 1.

DAC 2003, June 2-6, 2003, Anaheim, California, USA.

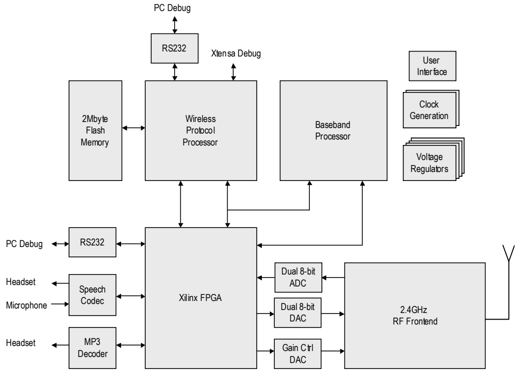


Figure 1: System block diagram.

3. BASEBAND PROCESSOR

The physical layer is made compatible with a commercially available RF front-end (performing carrier up/down conversion), ADC, and DAC. Although the commercial components have high power consumption resulting from their tight design specs, the PHY accommodates significantly relaxed specs for eventual integration with a custom, low-power front end (e.g. by only requiring a free-running clock with 50 ppm accuracy). The chip integrates all other PHY receiver and transmitter functions, such as carrier detect, synchronization, and detection.

The air-interface is DSSS with a length 31 spreading code at 25 Mcps and QPSK modulation resulting in a raw data rate of 1.6 Mbps. The primary receiver specifications are a +/-100KHz maximum carrier frequency offset (+/-50 ppm, 2GHz reference), 5 dB minimum input SNR at the ADC, and a 50ppm ADC sample clock. The BBP supports a typical indoor frequency-selective wireless channel with mobile units traveling at foot speeds.

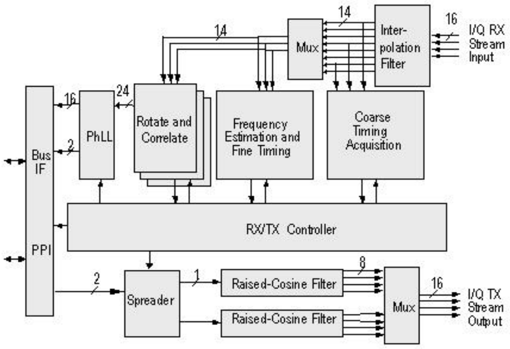


Figure 2: Baseband Processor (BBP) block diagram.

High-level system exploration in Simulink enables algorithm refinement and power optimization of the physical layer. A block diagram is shown in Figure 2. The RX/TX Controller interfaces with the WPP and controls the data flow from one datapath block to another by adaptively activating the 5 gated clock domains.

During receive, the baseband signal is 4x over-sampled by dual off-chip 8-bit ADCs at 100 Msps. These 100 MHz streams are each split into four parallel 25 MHz streams so that the BBP can operate off the slower 25 MHz chip clock reducing power by allowing a lower operating voltage. Parallel filter techniques interpolate the streams to increase the receiver timing resolution to 8 samples per chip. Performing on-chip interpolation of the signal is lower power than running the ADC at twice the rate.

The use of two successive stages halves the power consumption of the timing recovery function. First, the coarse timing block performs carrier detect, and estimates timing to within 3/8 chip. Then, the fine timing block estimates timing to within 1/8 chip and the carrier frequency offset to within 2.5 Hz.

The rotate and correlate block corrects the frequency offset, correlates the incoming signal with the spreading code, and performs early/late detection to track the optimal timing instant. A phasor locked loop (PhLL) corrects the phase error of the correlated symbols using feedback and the QPSK symbols are demodulated. Where possible, coefficients are restricted to factors of two, so that shift-and-add operations can be used instead of the more power hungry multiplication operations.

In the transmit mode, data bits are mapped into QPSK symbols, spread by a dual-channel spreader, raised-cosine filtered (25-taps, alpha = 0.30), and passed to dual off-chip DACs.

The BBP has several features that facilitate testing, including a full scan chain. The transmitter output pins are converted into a 64-bit test bus during receiver testing. Internal receiver signals such as the RX/TX controller state, code matched filter outputs, frequency estimate, and soft symbols, are output to this bus to aid in testing and debug.

4. BBP DESIGN METHODOLOGY

The design flow of the datapath-dominated BBP allows high-level design exploration using MATLAB/Simulink dataflow diagrams. Directly mapping these dataflow algorithms into hardware yields maximum parallelism, allowing the minimum clock rate and supply voltage to be used, resulting in reduced energy per operation. High-level power estimation and successive refinement are achieved with parameterized modules programmed in Synopsys Module Compiler. An in-house back-end design flow, called SSHAFT [1], allows a direct path from Simulink and Module Compiler to heavily parallelized, direct-mapped ASIC implementations.

Since the entire design is encapsulated in Simulink, it can be simulated along with models of analog front-end. Therefore, the effect of analog nonidealities and fixed-point computation can be evaluated at a system level. Extensive system level simulations were done to ensure proper operation over the range of channel and circuit nonidealities. For instance, Figure 6 shows the locking behavior of the PhLL.

The SSHAFT flow enables early exploration for architectural tradeoffs between power consumption, speed, and die area. Fixed-point Simulink library models correspond to parameterized arithmetic units designed in Module Compiler. Specific modules can be quickly compiled and simulated for accurate gate-level power estimation.

The SSHAFT flow also automates the physical design process. Simulink 'enable signals' are converted into gated clocks during automatic clock tree generation. This reduces power by eliminating the switching activity when a block is not in use. While the BBP is datapath dominated, some control is still required to steer the data and manage the gated clock domains. Controllers are described as state machines in Simulink/Stateflow and they are automatically translated to VHDL and merged into the design.

The BBP uses a hierarchical floorplan consisting of two levels of hierarchy. Each block at the lower level (outlined in Figure 5a) is placed and routed separately and then connected at the top level. Trial runs done on a preliminary netlist show that hierarchical place and route result in 18% smaller area than flat place and route. Switch-level simulations of the extracted layout are conducted in PathMill to ensure that the design met timing over all corners.

5. WIRELESS PROTOCOL PROCESSOR

The WPP is an energy-efficient realization of the DLL, transport, session, and application protocol layers. The architecture is a system-on-chip design consisting of multiple cores connected by a system bus, as shown in Figure 3. The main components on the chip are a Sonics SiliconBackplane system bus that connects a Tensilica T1030 Xtensa RISC microprocessor with 64/64kbyte instruction/data memories, a Protocol Processing Engine (PPE), and various interface units. All cores on the chip use a standard interface to permit communication using a generated SiliconBackplane system bus from Sonics. Adoption of a standard protocol (OCP) simplifies the design process by clearly defining the interface between blocks and enables easy core reuse, which can reduce time to market for future projects [3].

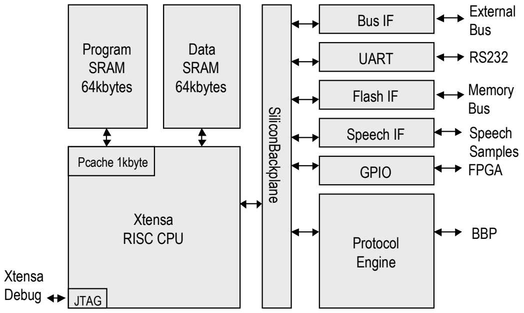


Figure 3: Wireless Protocol Processor (WPP) block diagram.

The Xtensa microprocessor performs system initialization and allows flexible implementation of the application, session, and transport protocol layers. Analysis of the processor requirements shows that a modest 12.5 MHz system clock is sufficient, which reduces power by lowering the switching frequency and allows the use of a 1.0V core supply voltage. The Xtensa supports design-time customizations, which allow the selections of data path components and memory hierarchy to be tailored to match requirements, which prevents wasted power due to over-design.

Custom logic implements the DLL, audio compression, and several off-chip interfaces. A serial port interface is used to output an activity log and to download software upgrades. A test access port allows an external debugger control the Xtensa by setting breakpoints and single-stepping the software code. A special test mode allows detection of manufacturing faults by converting 19 I/O pins into a port that controls access to the on-chip scan-chains and memory Built-In-Self-Test (BIST). A FPGA interface is included to ease pattern generation during system testing.

6. WPP DESIGN METHODOLOGY

The design flow of the control-dominated WPP is based on system-level exploration between hardware and software implementation trade-offs. The protocol stack includes 45 Co-design Finite State Machines (CFSMs) because of their suitability for modeling control-dominated systems without requiring assumptions on the underlying implementation. The CFSMs are captured and simulated within the Cadence VCC tool.

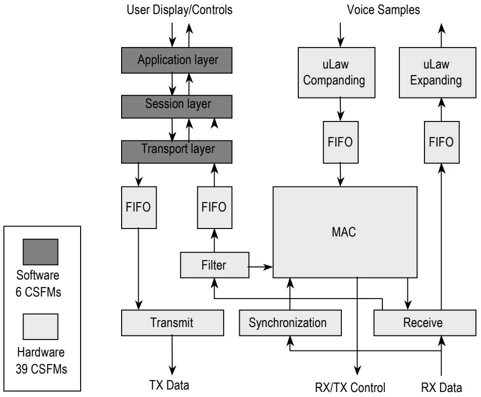
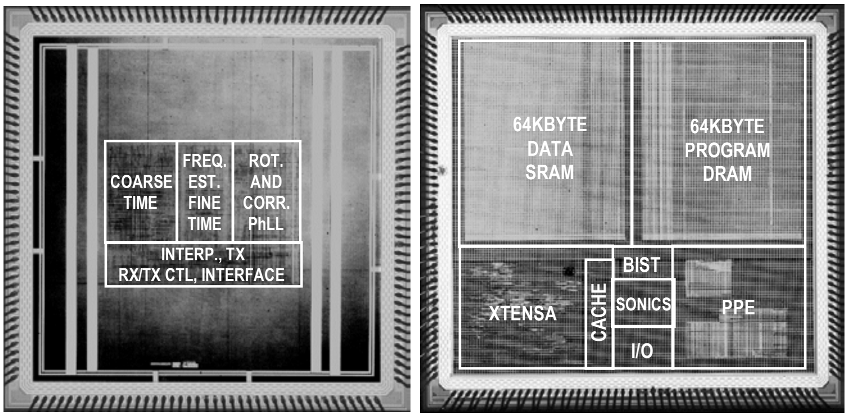


Figure 4: Protocol stack functions and hardware/software partitioning.

The VCC model is iteratively refined as part of the implementation process. The initial simulation allows the designer to focus on the correct operation of the algorithm sequences by abstracting time. Once the correct operation is verified, alternative functional mappings onto hardware architectures are evaluated to identify an implementation that minimizes power consumption while meeting the timing and flexibility requirements. The conceptual protocol stack and final partitioning is shown in Figure 4. For most functions, hardware realizations are favored for their energy-efficiency but high-level protocol layers are mapped into software to allow changes to the user-interface and communication channel allocation algorithm. Once a suitable architectural mapping is chosen, software is generated directly from the VCC description and custom hardware is implemented by hand-written Verilog code combined with generated code for the Xtensa and SiliconBackplane.

Three levels of simulation ensure the correct implementation of the WPP before physical implementation. First, each core on the chip is simulated independently within a Verilog simulator to verify the functionality and check compliance to the standard interface. Second, a complete node is co-simulated using an instruction set simulator (ISS) for the software and Verilog simulator for the custom hardware. Once the co-simulation operates correctly, the RTL code for the Xtensa is substituted for the ISS to verify the processor interface. The co-simulation step is preferred for early simulations due to its enhanced software debugging features and reduced simulation run-times. Third, a system test consisting of a base station and two remotes is used to check correct operation of the TDMA protocol.

The resulting Verilog code is implemented using an industry-standard timing-driven digital design flow. Power consumption is reduced using a high V_t, low-leakage standard cell library during synthesis. After floorplanning, placement, and routing, the extracted layout, including parasitics, is verified through static timing analysis and switch-level simulation.



	BBP	WPP
Process	Triple-well, 0.18u digital CMOS, with 6 metal layers	
# Transistors	600 K	1.3 M
Area	Core: 2.2mm^2 Die: 14.5mm^2	Die: 17.6mm^2
Package	208-pin PGA	208-pin PGA
Core power supply	1 V	1 V
I/O voltage	1.8 V	1.8 V
Clock Frequency	25 MHz	12.5 MHz
Avg. power/node (3 node network)	3 mW (15% duty cycle)	10 mW

Figure 5: (a) BBP die micrograph, (b) WPP die micrograph, (c) BBP and WPP statistics

7. TESTING AND RESULTS

The BBP and WWP die micrographs are shown in Figure 5a,b. The fabrication process, area, clock frequency and power consumption are summarized in Figure 5c.

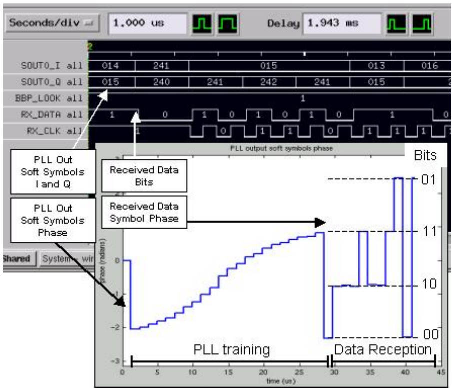


Figure 6: Simulation and matching chip test results of the BBP PhLL output.

The BBP and WPP chips are integrated on a system board that, in combination with a 2.4GHz RD0310 radio board, form the test system. The Xilinx on the test board acts as a pattern generator during test. For the BBP, the chip outputs were compared with the expected outputs from the Simulink simulation (as shown in Figure 6 for the PhLL outputs). For the WPP, a loopback mode is implemented in the Xilinx to emulate the BBP bit stream. The inter-chip interfaces are verified by comparing the logic analyzer traces with the expected values from the RTL simulations.

The BBP chip consumes 14 mW on average when receiving a short packet consisting of 40 synchronization symbols and 20 data symbols. (Longer packets have lower average power consumption.) During idle mode (TX and RX off), the chip consumes less than 1 mW. When three nodes are connected to the network, the BBP has a duty cycle of 15%, so the expected power consumption is 3 mW. The WPP chip consumes 10 mW on average when three nodes are connected to the network. The actual power consumption varies depending on whether the node is a remote or a basestation and the number of slots a remote is monitoring.

8. SUMMARY

A low power wireless system for a voice intercom application was designed. High-level design methodologies were used to define the control-intensive protocol stack and data-path-intensive physical layer algorithms and successfully identify low-power architectures. Several commercial IP blocks were integrated, and two custom ASICs were fabricated and tested. The two chips implement the entire digital portion of the protocol stack and are fully-operational. Combined, the chips consume an average of 13 mW during normal operation when three nodes are connected to the network.

9. ACKNOWLEDGMENTS

This work is funded by DARPA, SIA GSRC, the BWRC sponsoring companies and the National Technology Agency of Finland. IP was provided by Sonics, Inc. and Tensilica. The chips were fabricated by ST Microelectronics.

10. REFERENCES

[1] W. R. Davis, et.al., "A Design Environment for High Throughput, Low Power Dedicated Signal Processing Systems," IEEE JSSC, vol. 37, no. 3, pp. 420-431, March 2002.

[2] J. M. Rabaey, et.al., “PicoRadio supports Ad-Hoc Ultra-Low Power Wireless Networking,” Computer, vol. 33, no. 7, pp. 42-48, July 2000.

[3] D. Wingard. “On-chip Networks: the Secret to High-Performance Communication SoCs, Flexible Platforms. *Integrated Communications Design*, August 2000.

[4] D.G.-W. Yee, “A design methodology for highly-integrated low-power receivers for wireless communications,” Ph.D. dissertation, UC Berkeley, Department of EECS, Berkeley, CA 2001.

[5] “Information Technology – Open Systems Interconnection – Basic Reference Model,” International Organization for Standardization (ISO). Standard number ISO/IEC 7498-1:1994, 1994.