Extended Abstract: Tradeoffs and Design Choices for a Software Defined Acoustic Modem: A Case Study

Andrew Gray, Payman Arabshahi, Sumit Roy,
Newell Jensen, Leonard Tracy, Nathan Parrish, Chung Hsieh
University of Washington, Dept. of Electrical Engineering, Box 352500, Seattle, WA 98195-2500
(626) 757-6903, aagray@washington.edu

1. INTRODUCTION

Challenges in developing high performance underwater acoustic modems can be summarized as:

1. *Lack of maturity in underwater acoustic communications technology at all layers of the communication stack.* This stems in part from the complex and dynamic nature of the underwater channel.
2. *Spatially and temporally dynamic nature of the underwater acoustic channel.* Theoretically "high performance" communications may not be achievable using the largely channel-specific (channel-static) processing techniques similar to those which have evolved for radio frequency and optical communications.
3. *Resources.* Without the expectation that the processing being implemented will reach a point of high maturity and meet a high-priority near-term operational need, it can be difficult to justify such an application specific investment.

Underwater acoustic communications technology is still relatively immature, with significant improvements possible at all layers of the network stack. Different acoustic channels will require significantly different modem functionality – ranging from frequency selection, modulation and error control coding type, to MAC layer and network routing algorithms – to achieve optimal or near-optimal performance. Important work has been initiated and demonstrated in this area and continues to this date, while much remains to be done. Given these assessments, we have focused on the following set of goals as drivers for the design of the underwater acoustic modem that is presented herein; these are:

- Low-cost adaptability and upgradability to:
  1. Facilitate the evolution of the state-of-the art in mathematical processing through field experimentation, testing, and validation of new algorithms.
  2. Facilitate aligning operational capability with an evolving state-of-the art in underwater acoustic communication.
- A migration path to a device that can change functionality "on-the-fly" as new channels are encountered operationally.
- A migration path to an application specific device that can be produced to serve the needs of specific operations more efficiently (volume, mass, power) than a generic approach, once algorithms have reached a desired level of maturity.

2. MODEM DESIGN

The first step in the design of our software defined modem was to create the appropriate trade space to analyze various design criteria, derived from our goals stated above. The trade studies were used to make key decisions in the development of a point-design that is currently being prototyped, and is presented below.

WUWNet'09, November 3, 2009, Berkeley, CA, USA. Copyright 2009 ACM 978-1-60558-821-6

2.1 Trade Studies

The figures-of-merit used in making key decisions about the processor-type for acoustic modem architecture are presented in the rows of Table 1, while the columns represent the processor types described below. For this effort the most critical figure-of-merit is the "cost to experiment." There is a need to test and validate new algorithms in the field rapidly and cost effectively that is not currently being met by existing solutions. It should be noted that most modem architecture choices will contain multiple processor types; the table indicates processors utilized for "user defined" functionality (signal processing algorithms). For example, any architecture will contain analog-to-digital converters (ADCs), which are ASICs, but the user defined processing may be accomplished with any one of the four types of processors or a hybrid architecture containing one or more processor type. The design team eliminated such hybrid architecture choices since in most cases they are not aligned with the goal of low-cost-to-experiment; multiple processor types often requires multiple engineers to make even minor changes to the implementation.

Table 1. Processor type evaluation using figures-of-merit

	Processor Type			
Figures of Merit	**GPM**	**DSP**	**FPGA**	**ASIC**
"Cost-to-experiment" (software)	Low	High	High	High
Open source	High	High	Hi-Med	Low
Migration paths/synergies	High	Med	Med	Low
Non-recurring engineering costs	Low	Medium	High	High
Power	High	Low-Med	Low	Low
Mass/Volume	High	Low	Low	Low
Risk (schedule/cost/performance)	Low	Medium	Medium	High

The first critical architecture design decision is the type(s) of processor(s) to be used:

- General-purpose microprocessors (GPM) that use an operating system
- Digital processors (DSP) that use high level languages but do not have a full operating system.
- Field programmable gate arrays (FPGA) which have no operating system and cannot be programmed easily or at low cost (configuration or reconfiguration requires detailed and specific coding approaches that are more specialized than coding in say, C or C++).
- Application specific integrated circuits (ASIC), whose functions are determined via a one-time configuration of transistors that occur at manufacturing.

These processor building blocks each have their own advantages and disadvantages and can be compared in a number of ways; however it is difficult to compare them generically in a quantitative way. However, in general the GPM with an operating system such as Linux or Microsoft Windows™ can have functions changed very rapidly and inexpensively relative to the other three choices, via software languages such as Matlab, C, C++, or C#.

Based on the analysis and figures-of-merit outlined above the architecture chosen was one based on a general-purpose processor with an operating system, with the following characteristics:

- Commercial-off-the-shelf architecture augmented with some level of custom design (adopts significant standardization, especially in operating systems, drivers, and interfaces: user, electrical, physical). This results in low non-recurring engineering costs and development risk, low-cost-to experiment, and open source environment (Linux).
- Composed of flexible programmable and/or reconfigurable components. This results in many potential migration paths.
- The design is simple to modify and use. We follow a disciplined approach to utilizing standardization when possible, and a very high level of open source adoption and documentation created for the design, implementation, and test. This is aligned with the low cost-to-experiment goal.
- The design's power, mass, and volume are significantly higher than other choices. However, migration paths exist to application specific devices that will feature improvements in all of these areas, in the future.

2.2 Prototype Design Overview

The critical components of the acoustic modem prototype are presented in Table 2, and a block diagram is presented in Figure 1.

Table 2. Critical components of the acoustic modem prototype

Processing Platform	**Operating systems**
- ITX (mini) with FPGA buffer/filter front-end	- Linux is baseline
Microprocessors	- Investigating RTOS
- x86 CPU (500 MHz)	**Interfaces**
- MSP430 microcontroller	- RS-232
Power subsystem	- USB
- Standby/shutdown/recover. op modes	- PCI
- Dual asymmetric power supply	- COTS ADC/DAC (14 bits)
Package:	**Expected prototype specs:**
- TBD (semi-custom Pelican case)	- Mass: 0.5 Kg
Acoustic front-end (transmit/receive):	- Volume: ~ 20 x 20 x 6 cm
- COTS discrete components	- Bandwidth (end-to-end): 50 KHz

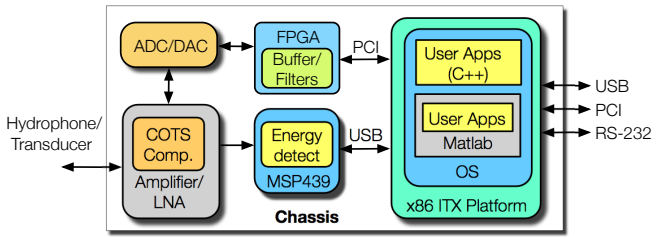

Figure 1. COTS-based modem prototype design

2.2.1 Transmit/Receive Front-end

The analog-front end of the software modem consists of a linear amplifier and a projector (for the transmitter, TR) and a hydrophone and pre-amp (for the receiver, RX). The architecture is designed to be as independent as possible from the digital-back-end starting with the analog-to-digital (ADC) and digital-to-analog (DAC) converters. The sampling rates for the ADCs/DACs are well above any usable acoustic frequency, thus changing the signal bandwidth and center frequency to match the analog hardware is simply a matter of adjusting the parameters in the firmware (FPGA buffer and filters) and software implementation.

For our initial Phase I in-water tests the prototype implements the transmission front-end utilizing an Instrument Inc. L2 amplifier which has a linear operation range from 1-200 kHz and maximum power output of 600 VA, and an ITC-2044 transducer. The receiver front-end is implemented with an ITC-6050C broadband omni-directional hydrophone which includes a built-in pre-amplifier and is sensitive to approximately 75 kHz. The ITC-2044 transducer has an approximately flat frequency response between 8 kHz and 14 kHz thereby defining our system bandwidth to be approximately 6 kHz centered at 11 kHz. These components form a sample configuration intended for research and testing, and were chosen primarily based upon their availability from previous work. Depending on the application, other front-end TR/RX designs will be better suited. Energy efficienc, for example, can be drastically improved by using an amplifier well matched to the projector.

2.2.2 ITX Platforms

The ITX platforms have been developed for the embedded systems market and are available in three standard dimensions: 1) Mini-ITX – 170mm x 170mm, 2) Nano-ITX – 120mm x 120mm, and 3) Pico-ITX – 100mm x 72mm. The ITX motherboards host x86 based CPUs ranging from 300 MHz to 2.0 GHz dual core. Since they are x86 based, common operating systems such as Linux or Microsoft Windows™ are readily available. This enables researchers to port algorithms working on their desktops in Matlab™ or C/C++/C# immediately to the field environment. The prototype design presented in this paper utilizes the mini-ITX; however there is a migration path to smaller mass/volume profile and lower power by using the nano-ITX or pico-ITX platforms.

2.2.3 Power System Design

To minimize power a system has been designed to keep the microprocessor on the ITX and the FPGA used for digital TR/RX buffering and filtering idling when not being used for transmitting or receiving. A microwatt Texas Instrument MSP430 microcontroller remains awake implementing a simple energy detector, along with the ADC. When the MSP430 determines (using the energy detector) an event of interest is occurring it wakes up the FPGA and microprocessor unit to process the received signal. The analysis and optimized design (e.g. minimizing cost functions incorporating probability of false-alarm) for networked operations are ongoing.

3. VERIFICATION AND EXPERIMENTS

Various receiver designs can be readily verified at the University of Washington through simulation using a comprehensive underwater channel simulation tool, e.g. the Sonar Simulation Toolset (SST), developed at the University of Washington's Applied Physics Laboratory (UW-APL). SST enables users to build an artificial ocean that sounds like a real ocean. SST's simulated signals include reverberation, target echoes, discrete sound sources, and background noise with specified spectra. Externally generated or measured signals can be added to the output signal or used as transmissions.

The next step following simulation will be field testing of various receiver designs (physical layer and above). The first phase of the experiment plan for our software modem includes a day-long experiment with two UW-APL ships in Lake Washington, an environment characterized by a muddy bottom (resulting in short channel delay spreads at long distances). The second phase will be additional day-long tests with three UW-APL ships in a shallow ocean environment (Puget Sound), representing an extremely harsh underwater channel.