Survey of Hardware Platforms for an Energy Efficient Implementation of Matching Pursuits Algorithm for Shallow Water Networks

Bridget Benson, Ali Irturk, Junguk Cho, and Ryan Kastner
Department of Computer Science and Engineering
University of California San Diego
San Diego, CA 92093-0404 USA
email: {b1benson, airturk, jucho, kastner} @ cs.ucsd.edu

ABSTRACT

Coral reefs worldwide are in serious decline. Underwater wireless sensor networks may be the answer to providing the persistent monitoring presence needed to obtain the data necessary to better understand how to protect these ecosystems for the future. Many advances have been made in underwater acoustic communication devices for underwater wireless sensor networks, but a major challenge that still remains is obtaining an energy efficient modem design. To begin to address this challenge, we implement the Matching Pursuits algorithm for channel estimation, an energy consuming portion of an existing underwater acoustic modem designed for shallow water networks, on a variety of hardware platforms. We determine that a dedicated field programmable gate array (FPGA) intellectual property core provides the most energy efficient hardware platform for Matching Pursuits which motivates future work to port the entire modem design to an FPGA for an energy efficient modem design.

1. INTRODUCTION

Coral reefs worldwide are in serious decline, owing primarily to over-harvesting, pollution, disease, and climate change. Even the Great Barrier Reef, widely regarded as one of the most `pristine' coral reefs in the world, shows systemwide decline [1]. We need persistent long-term monitoring of key physical, biological, and chemical variables to better understand the dynamics of coral reefs to learn how to protect these ecosystems for the future.

Underwater wireless sensor networks (UWSN) may be the answer to providing a persistent monitoring presence in coral reef ecosystems as they could provide a means to gather long term data without the need for manual retrieval of instruments or the use of potentially damaging undersea cables. However, the undersea environment proposes many challenges to wireless communications including severely limited range-dependent bandwidth and attenuation, extensive time-varying multi-path propagation (especially in shallow water), long propagation delays, and limited battery power [2]. These challenges demand the need for communication devices designed specifically for the underwater application.

Many of these challenges have been addressed as commercial and research underwater modems do exist. However, commercial modems, such as [3-5] were designed for sparse, long-range (10s of kilometers), systems making them too power hungry and expensive for small, dense sensor-nets [6]. The research designs in [7-8] and the low-power, low-rate mode of [9] are based on noncoherant frequency shift keying (FSK) which is not necessarily the most energy efficient signaling scheme. It has been shown that direct sequence spread spectrum waveforms yield significantly lower error rates than FSK due to frequency diversity [10-11] allowing for lower energy consumption. The UCSB AquaModem [11-12] designed specifically for use in the coral reef UWSN application, uses direct sequence spread spectrum signaling, but was implemented on a TMS320C6713 Digital Signal Processor (DSP) which may not be the most energy efficient hardware platform.

Therefore, low energy consumption still remains a significant design challenge for underwater communication devices. Low energy consumption is an extremely important design parameter for underwater acoustic modems for UWSNs as it is difficult and expensive to retrieve underwater devices for battery replacement. To begin to address this challenge, we implement the Matching Pursuits algorithm for channel estimation, on a variety of hardware platforms to select a hardware platform that provides the most energy efficient implementation. We focus on the AquaModem design because it was successfully field tested from July 24th — July 28th, 2007 at the Moorea Coral Reef Long Term Ecological Research Site [12]- our application of interest.

The paper is organized as follows: In section II we present the Matching Pursuits algorithm for channel estimation and describe the design specifications for its use in the AquaModem. In section III, we survey three different hardware platforms used to implement this algorithm: the TMS320C6713 DSP, the Spartan xc3s4000 field programmable gate array (FPGA) chip running MicroBlaze, and the Spartan xc3s4000, running a custom FPGA intellectual property (IP) core. In section IV we compare the energy consumption of each of these implementations and discuss which hardware platform provides the most energy efficient implementation. We conclude with a discussion on future work in section V.

2. MATCHING PURSUITS ALGORITHM AND DESIGN SPECIFICATIONS

The AquaModem uses Matching Pursuits for joint channel estimation and detection. Matching pursuits is a computationally intensive and energy consuming portion of the AquaModem design, therefore reducing the energy consumption of MP will reduce the energy consumption of the entire modem design. In this section we present the MP algorithm for channel estimation and the design specifications necessary for its implementation for the AquaModem.

The MP algorithm for channel estimation is shown in Figure 1. MP takes in four matrices as input: the receive signal vector E C^{2}Ns x 1, the signal matrix S, E R^{2}Ns Ns defined in [11], the Hermetian matrix *A* = *SS*, E RNs x Ns , and vector *a*, E RNs x 1. The vector *a* is simply one divided by the diagonal elements of *A* and is used to eliminate the need for division operations. The *S, A*, and *a* are static matrices as the values are known *a priori* and therefore can be pre-computed once and stored in memory. In steps (1-5), MP computes matched filter outputs, *V* E CNs x , and initializes the channel coefficients, *F*E CNsx , and temporary channel coefficients, *G* E CNs 1, to zero. In steps (7-15) MP loops over the hypothesized number of paths, iteratively canceling the strongest detected signal component to estimate the next channel coefficient. Specifically, MP updates the matched filter outputs by canceling the strongest detected signal component (8) and computes decision variables, Q E RNs x 1, and temporary channel coefficients, MP then searches for the next strongest channel coefficient by finding the index, *4*, of the maximum decision variable, Q, that is not equal to any index that has already been found (13). MP saves the temporary channel coefficient value at that index, G_4, as the next strongest channel coefficient(14). This coefficient is then used in the next iteration of the for loop in (7) for successive cancelation. When the algorithm is complete, it returns the estimated channel coefficients (16).

The algorithm applies to any direct sequence CDMA (DS-CDMA) signal, but we describe the design specifications of the DS-CDMA signal of the AquaModem as we are specifically interested in these signals.

[Insert Figure 1]

The AquaModem uses M-ary direct sequence spread spectrum signaling based on eight composite Walsh and m-sequence waveforms [11]. These waveforms are instantaneously wideband providing robustness to frequency-selective multipath. Each waveform is comprised of 8 symbols each symbol comprised of 7 chips (as shown in Figure 2). This 56 chip waveform must have a time duration greater than 10 milliseconds, the duration of the multipath spread in shallow water. Therefore, the chip duration is given as 0.2 ms, making the waveform duration 11.2 ms. An 11.2 ms time guard band for channel clearing is added to eliminate the need for equalization. Nyquist sampling requires the sampling interval to be half the chip duration, giving 112 samples per symbol for the transmitted waveform and an additional 112 symbols for the time guard band. Thus the receive vector consists of 112 + 112 = 224 samples and the time between received signals is 11.2 + 11.2 = 22.4 ms. These design specifications are summarized in Table 1.

These design parameters govern the size of the input receive vector, r, and signal matrices S, A, and a

for MP algorithm shown in Figure 1. The receive vector is of size 224 x 1, the signal matrix, *S,* is of size 224 x 112, the Hermetian matrix, *A* matrix is of size 112 x 112, and the vector *a* is of size 112 x 1. Each element in these matrices must be 8 bits wide for accurate channel estimation [13] making the total size of the static *S*, *A*, *a* matrices —38KB combined. *G* E CN*s* x (10-11).

[Insert Table 1]

[Insert Figure 2]

3. HARDWARE PLATFORMS

In this section we present three different hardware platforms used to implement the MP algorithm described in section 2. These three platforms include the TMS320C6713 DSP, the Spartan xc3s4000 FPGA chip running MicroBlaze, and the Spartan xc3s4000 running a custom FPGA IP core. These platforms were selected to encompass the platform options of a DSP, microcontroller, and custom FPGA IP core for embedded systems.

3.1 DSP

The AquaModem was implemented in a TMS320C6713 DSP evaluation board. The C6713 is a recent version of the TMS320 family of DSPs provided by Texas Instruments and claims to be the highest performance digital signal processor. It provides clock rates of up to 225 MHz, a cache-based memory architecture that provides up to 256KB L2 RAM, and a suite of peripherals (such as the TLV320AIC23 codec) used to interface with the analog components of the modem. The board is great for new designs as it can be easily re-programmed in C in an integrated development environment such as Code Composer Studio. The integrated peripherals also ease its integration with the modem's analog peripherals. However, DSPs come with a price of high power consumption and therefore is not the best choice for an energy efficient modem design.

3.2 MicroB/aze

MicroBlaze is a 32-bit soft processor core from Xilinx optimized for use in Xilinx FPGAs and is another viable option as a hardware platform for Matching Pursuits. MicroBlaze combines the design flexibility of FPGAs with a standard C programming environment, allowing the designer to produce highly customized designs at a relatively fast rate. MicroBlaze offers complete flexibility to select any combination of peripherals, memory and interface features to give good system performance. It can be implemented on any Spartan 3 or high performance Virtex 5 FPGA device without any royalties.

We implemented the Matching Pursuit algorithm on a MicroBlaze implemented in a Spartan xc3s4000-5fg676 FPGA because this device offers relatively low power consumption (has 0.283W quiescent power) and sufficient on chip block RAM (BRAM) to store the large input *S, A*, and *a* matrices. Implementing the Matching Pursuits algorithm on MicroBlaze involved using Xilinx SDK to write the MP algorithm in C and store the static matrices *S, A*, and *a* in on-chip Block RAM.

3.3 Custom IP Core

An FPGA IP core is a block of logic or data that is used in making an FPGA customized for a specific application. FPGA IP cores allow embedded systems designers to create application specific designs that contain only the resources necessary for their particular application. FPGA IP cores can be written in a hardware description language (such as Verilog or VHDL) or can be implemented using higher level design tools such as Xilinx' System Generator.

We use Xilinx' high level design tool, System Generator, to implement and test an FPGA IP Core for the MP algorithm. This design is based on the MP core implementation found in [13] and is shown in Figure 3. The design uses duplicate hardware to process the real and imaginary data at the same time and makes use of two dedicated DSP48 multipliers, 3 adders, 8 multiplexers, 2 accumulators, and 3 storage registers in addition to BRAM to store the *S, A*, and *a* static input matrices and registers to store the matched filter outputs, *V*, temporary channel coefficients, *G*, channel coefficients, *F*, and decision variables *Q*. The control of the block first computes the 112 real and 112 imaginary matched filter outputs, F, (steps 1-5) and then performs the successive interference cancellation in steps 8-12. To determine the next path to cancel, the 112 real Q, 112 real G and 112 imaginary G values are fed into the q-gen block to find the index q, of the next strongest channel coefficient (step 13). The outputs of the q-gen block are then fed back into the Filter and Cancel block to perform the successive interference cancellation again (the loop in steps 7 15). Once the loop in 7-15 has finished, the F registers contain the estimated channel coefficients. The Filter and Cancel (FC) block can be duplicated up to 112 times to allow one FC block to process each of the 112 columns at the same time.

[Insert Figure 3]

4. RESULTS

In this section we present the results of the MP implementations on the TMS320C6713, the Spartan xc3s4000 chip running MicroBlaze, and the Spartan xc3s4000 running our custom FPGA IP core. All implementations use an estimated number of paths of NT6 (based on the estimated number of paths used in the AquaModem). The total computational time, computational power, and energy consumption of each implementation are shown in Table 2.

[Insert Table 2]

The total computational time for the DSP was estimated by measuring the time to compute one coefficient (about 78 us) and multiplying this number by the size of Nf (6 coefficients). The total computational time for the MicroBlaze implementation was calculated using an embedded timer. The power for the DSP design was estimated using TI's SPRA889A2 Excel spreadsheet Power estimator and the power for the MicroBlaze and FPGA IP Core designs were estimated using the Xilinx Spartan 3 XPower Estimator. The MicroBlaze design was synthesized, placed and routed with Xilinx SDK 9.1 and the FPGA IP Core design was synthesized, placed, and routed with Xilinx ISE 9.1. The number of clock cycles for the FPGA IP Core design was measured in the System Generator environment and the computational time was calculated by multiplying this number with the maximum clock frequency reported by Xilinx ISE 9.1.

As can be seen from Table 2, the FPGA IP core implementations provides a significant reduction in energy consumption because of either reduced computation time (for the more parallel implementation) or reduced power consumption (for the serial implementation). The Microblaze design provides the worst solution in terms of energy consumption because of its high computation time (due to its lower clock rate (over the DSP) and inefficiencies.

5.CONCLUSION AND FUTURE WORK

The results clearly indicate that the IP Core implementation offers the most energy efficient design, but comes with a cost of a much higher design time, as simple C code cannot be used for such an implementation. The fact that the energy consumption for the custom IP Core is less than that of the DSP motivates porting the entire AquaModem design to an FPGA for an energy efficient underwater modem implementation. Furthermore, the benefit of using FPGAs will further improve in the future as FPGA devices develop the capability to turn off unused area of the device in a specific design allowing for even lower power consumption.

5. ACKNOWLEDGEMENTS

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship and National Science Foundation Grant #0816419. Our thanks to Daniel Doonan, the UCSB AquaModem's hardware engineer, for providing us with measurements for the DSP implementation.

6. REFERENCE

[1] D.R. Bellwood *et al.*, Confronting the coral reef crisis, *Nature* 429 (2004), pp. 827-833.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks Journal*, pp.257279, march 2005.

[3] Benthos, Inc. Fast and reliable access to undersea data. http://www.benthos.com/pdf/Modems/ModemBrochure.pdf.

[4] LinkQuest, Inc. Underwater acoustic modems. http://www.link-quest.com/html/uwm_hr.pdf

[5] DSPCOMM, Underwater wireless modem. http://www.dspcomm.com.

[6] J. Heidemann, Y. Li, A. Syed, J Wills and W. Ye, "Research Challenges and Applications for Underwater Sensor Networking", Proceedings of the IEEE Wireless Communications and Networking Conference. April 2006.

[7] J. Wills, W. Ye, and J. Heidemann, "Low-power acoustic modem for dense underwater sensor networks," in Proc. of WUWNet,Sept. 2006.

[8] B. Benson, G. Chang, D. Manov, B. Graham, and R. Kastner, "Design of a low-cost acoustic modem for moored oceanographic applications," in Proc. of WUWNet, Sept. 2006.

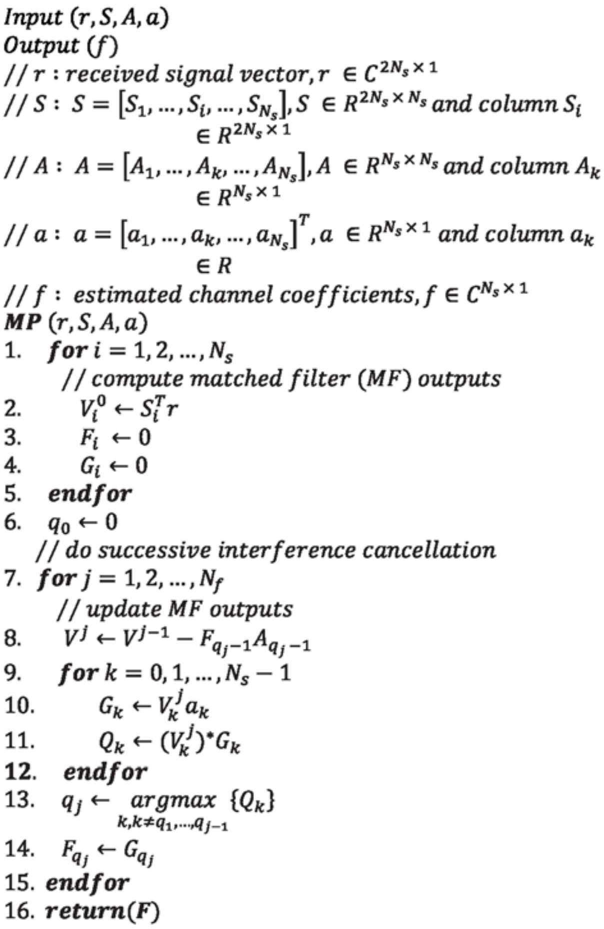
[9] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The WHOI Micro-Modem: An acoustic communications and navigation system for multiple platforms," in Proceeding of OCEANS, 2005.

[10] L. Freitag, M. Stojanovic, S. Singh, and M. Johnson, "Analysis of channel effects on direct-sequence and frequency-hopped spread-spectrum acoustic communication," *IEEE Journal of Oceanic Engineering*, vol. 26, pp. 586-593, 2001.

[11] Ronald A. Iltis, Hua Lee, Ryan Kastner, Daniel Doonan, Tricia Fu, Rachael Moore and Maurice Chin, "An Underwater Acoustic Telemetry Modem for Eco-Sensing" MTS/IEEE Oceans, September 2005

[12] Tricia Fu, Daniel Doonan, Chris Utley, Bridget Benson, Ryan Kastner, Ronald A. Iltis, and Hua Lee. Work In Progress Poster: "AquaModem Field Tests in Moorea" *International Workshop on Underwater Networks (WUWNet)*, September 2007

[13] Y. Meng, A.P. Brown, R.A. T.Sherwood, H.Lee, and R.Kastner, "MP Core: Algorithm and Design Techniques for Efficient Channel Estimation in Wireless Applications." Design Automation Conference (DAC) 2005



```
$Input\ (r, S, A, a)$
$Output\ (f)$
$//\ r : received\ signal\ vector, r \in C^{2N_s \times 1}$
$//\ S : S = [S_1, \dots, S_i, \dots, S_{N_s}], S \in R^{2N_s \times N_s}\ and\ column\ S_i \in R^{2N_s \times 1}$
$//\ A : A = [A_1, \dots, A_k, \dots, A_{N_s}], A \in R^{N_s \times N_s}\ and\ column\ A_k \in R^{N_s \times 1}$
$//\ a : a = [a_1, \dots, a_k, \dots, a_{N_s}]^T, a \in R^{N_s \times 1}\ and\ column\ a_k \in R$
$//\ f : estimated\ channel\ coefficients, f \in C^{N_s \times 1}$
$\boldsymbol{MP}\ (r, S, A, a)$
$\boldsymbol{for}\ i = 1, 2, \dots, N_s$
    $//\ compute\ matched\ filter\ (MF)\ outputs$
    $V_i^0 \leftarrow S_i^T r$
    $F_i \leftarrow 0$
    $G_i \leftarrow 0$
$\boldsymbol{endfor}$
$q_0 \leftarrow 0$
$//\ do\ successive\ interference\ cancellation$
$\boldsymbol{for}\ j = 1, 2, \dots, N_f$
    $//\ update\ MF\ outputs$
    $V^j \leftarrow V^{j-1} - F_{q_j-1} A_{q_j-1}$
    $\boldsymbol{for}\ k = 0, 1, \dots, N_s - 1$
        $G_k \leftarrow V_k^j a_k$
        $Q_k \leftarrow (V_k^j)^* G_k$
    $\boldsymbol{endfor}$
    $q_j \leftarrow \underset{k, k \neq q_1, \dots, q_{j-1}}{argmax} \{Q_k\}$
    $F_{q_j} \leftarrow G_{q_j}$
$\boldsymbol{endfor}$
$\boldsymbol{return}(\boldsymbol{F})$
```

Figure 1: Matching Pursuits Algorithm for Channel Estimation

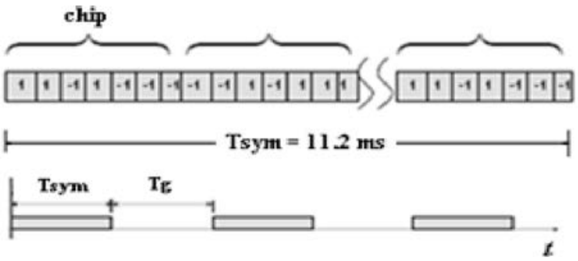


Figure 2: Wa/sh/m-sequence signa/s for the AquaModem

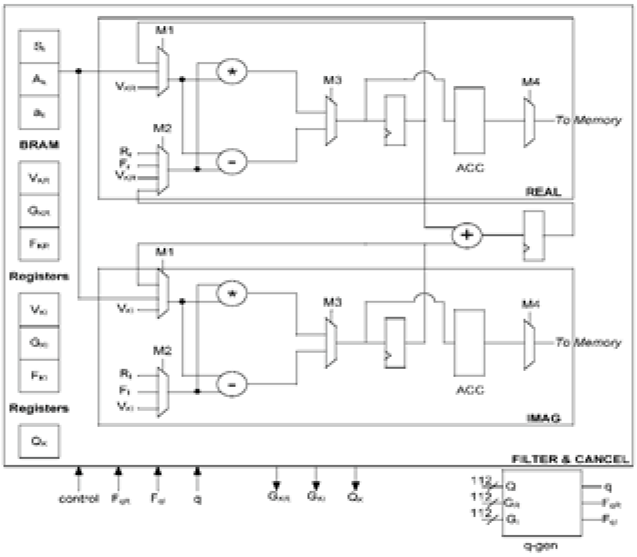


Figure 3: Custom MP Core for Channel Estimation

Table 1: AqauModem Design Parameters

Walsh Symbol length: N_w	8 symbols
m-sequence length: L_{pn}	7 chips
Chip Duration: T_c	0.2 msec
Sampling Interval: $T_s = T_c/2$	0.1 msec
Sybmol duration: $T_{sym} = L_{pn}*N_w*T_c$	11.2 msec
Time guard interval: $T_g = T_{sym}$	11.2 msec
Samples/symbol: $N_s = T_{sym}/T_s$	112 samples
Samples/time guard: $N_t = T_g/T_s$	112 samples
Total receive vector samples: $R_v = N_s + N_t$	224 samples

Table 2: Hardware Platform Implementation Results

Platform	Time (us)	Power (W)	Energy (mJ)
DSP	468	1.07	0.50
MicroBlaze	6341.84	0.38	2.40
IP Core (1 FC block)	643.10	0.30	0.19
IP Core (56 FC blocks)	23.08	0.72	0.016