Field-Programmable Gate Array in Miniature Ion Mobility Spectrometer Sensor System

S. M. Loo[1], Jon Cole[1], Robert Youngberg[1], Jake Baker[1], Molly Gribb[2]
smloo@boisestate.edu

[1]Department of Electrical and Computer Engineering, [2]Department of Civil Engineering
Boise State University, Boise, ID 83725

Abstract

This paper presents the use of field-programmable gate array (FPGA) in a miniature Ion Mobility Spectrometer (IMS) sensor system. This IMS sensor is for detection of subsurface gaseous volatile organic compounds (VOCs). This paper describes how an FPGA is used to process the data collected from the sensor and control its overall operation. A Xilinx Microblaze soft-core processor is used to ease *the development of the signal-conditioning algorithm.*

Keywords: FPGA, Processor core, IMS sensor, Microblaze, hardware/software codesign

I. Introduction

Ion Mobility Spectrometer (IMS) based sensor system has been used in the laboratory to analyze gaseous chemicals, and, more recently, in the field to detect chemical warfare agents, explosives, and narcotics. The IMS sensor uses the ion mobility spectrometry to separate and quantify ions based on the drift of ions at ambient pressure under the influence of an electric field against a counter flowing neutral drift gas [1].

Most existing methods for detecting and measuring VOCs in the vadose zone are often hampered by issues associated with east of use, accuracy, and cost. As a result, there remains a need for inexpensive, minimally invasive, real-time instrumentation and sensor systems that can be used for characterizing or long-term monitoring of contaminated sites. A new sensor system to meet this need is under development at Boise State University, in collaboration with Washington State University.

An FPGA has been used as the control and signal-conditioning unit in this miniature IMS sensor. A Xilinx Microblaze soft-core processor has been embedded in the FPGA to ease the development, which allows different signal conditioning algorithms to be used [2].

This paper is organized as follows. In section II, an overview of the IMS sensor system is provided. The use of FPGA in the IMS sensor system is described in section III. A description of our current prototype is described in section IV. Future work is listed in section V. The conclusion is then provided.

II. IMS Sensor System Overview

This miniature IMS sensor is designed and fabricated for use in unsaturated soils. The sensor system is comprised of many subsystems that reside either down-hole in the probe housing or up-hole on the surface. Figure 1 shows a block diagram of the system and Figure 2 shows the sensor system in its probe housing.

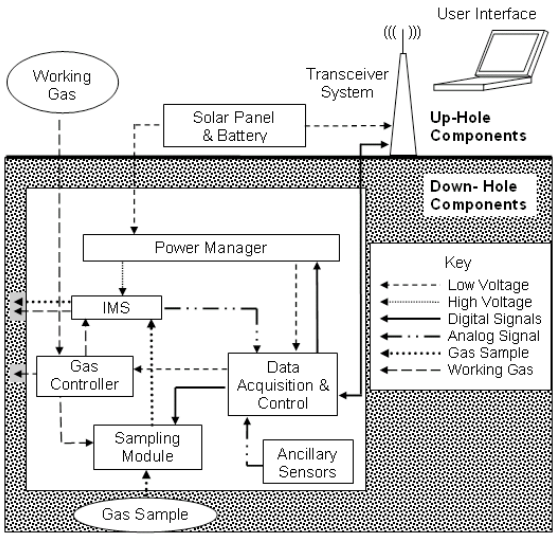

Figure 1. IMS sensor system block diagram.

When deployed, the IMS sensor probe will have "up-hole" system components consisting of a power source, data collection, and wireless transmission system (satellite or cellular network). The wireless data delivery will allow the data to be combined together from multiple IMS probes to determine the scenario at the monitored site. This IMS system will allow a cost effective method for long-term monitoring of sites and eliminates expensive equipment and labor [1].

The overall sensor system is composed of an IMS sensor, gas controller and sampling module, data acquisition and control module, communication module, and remote user interface (see Figure 2). This paper describes the use of an FPGA in the data acquisition, control, and communication modules.

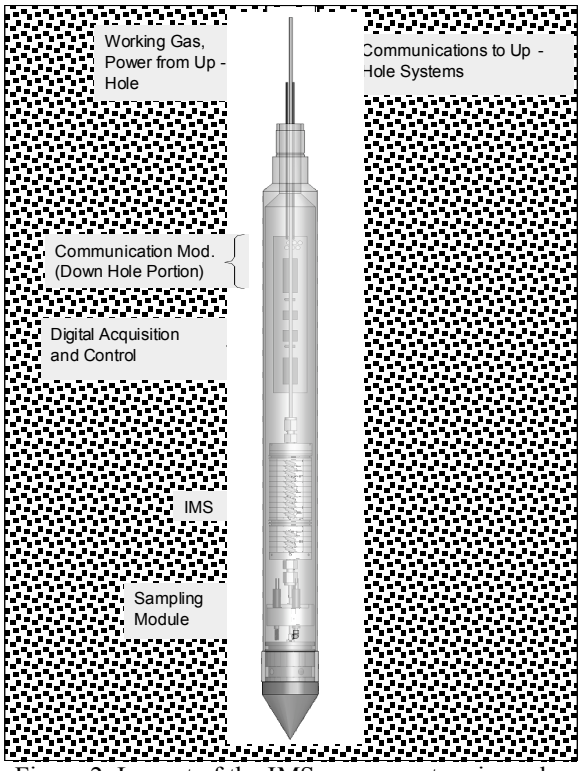

Figure 2. Layout of the IMS sensor system in probe housing.

III. FPGA & IMS

Due to the large amount of digital signal processing that is required to analyze and collect the data from the IMS; a field-programmable gate array (FPGA) was chosen over a more conventional microcontroller to handle the system control and data collection and data processing for this project. An FPGA is a device that can easily be configured and reconfigured any number of times to perform user-specific tasks. This reconfigurability makes an FPGA ideal for the IMS system during the prototyping phase, where changes can occur often. This reconfigurability also allows for easy updates or modifications to the device when it is in the field. The FPGA chosen for this design is the Xilinx Spartan-3 XC3S200 [3]. One very important factor in choosing the Spartan FPGA family is cost. It is envisioned that this sensor system will be deployed in large numbers to monitor a contaminated area. The initial cost of the sensor system will play a big factor on the number of units that can be deployed. Utilizing this family of FPGAs also allows for the design to be easily ported over to larger FPGAs within the same family and with the same device footprint.

FPGAs can also take advantage of the benefits of both hardware and software, by implementing a soft-core processor. A soft-core processor is a microprocessor that is created within an FPGA. The soft core processor used for this design is the Xilinx Microblaze soft core processor. Microblaze is a 32-bit Harvard RISC architecture IP core distributed with the Xilinx Embedded Development Kit (EDK) [2]. Microblaze is capable of running at speeds over 100 MHz and supports the C programming language. Microblaze has several interface methods to transfer data between itself and any hardware cores. This design makes use of a high-speed dedicated interface, called Fast Simplex Link bus (FSL)[4].

Figure 3 shows another block diagram of the IMS sensor system, but concentrates on the digital control and processing portion of the system. The FPGA is central to the entire IMS system. The FPGA serves three main roles:

- Communication with up-hole device
- Overall system control and timing
- Data collection and processing

Each of these important roles will be discussed in more detail in the following sections.

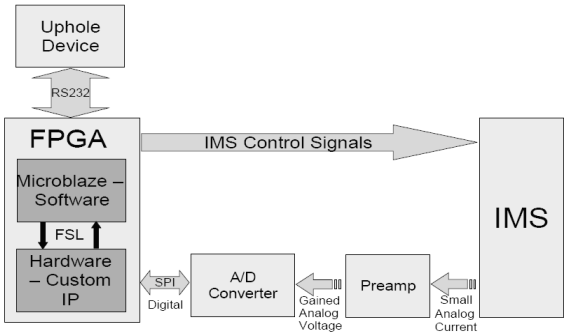

Figure 3. IMS electrical sub-system

A. Communication with the up-hole device

For this system, the FPGA acts as a slave to the up-hole device; the up-hole device then acts as the master. As a slave, the FPGA will not transmit any data to the up-hole device unless requested to do so by the up-hole device. The FPGA will stay in a standby mode until woken up by the up-hole device. During standby mode, the FPGA will consume minimal amounts of power (the Spartan-3L family is being considered). Once awoken out of standby mode, the FPGA will accept commands from the up-hole device. These commands may include starting a new scan, checking environment sensors, sending data, or altering the IMS timing and control parameters.

The communication protocol is implemented in Microblaze, where code can be written in C, a software programming language. Microblaze was chosen to implement the protocol because handling of commands is relatively simple in a software environment, where parsing character arrays is easy. This is a benefit of developing in a

hardware/software co-design environment, where the designer can choose which portions of the project are better suited for hardware or software.

Communication with the up-hole device is achieved via a communication standard called Recommended Standard 232 (RS232). This standard was chosen because of its wide use in industry and ease of implementation. RS232 is full duplex, communication in both directions, and transmits digital data serially at baud rates ranging from 110 to 115200 bits/sec.

B. Overall system timing and control

The FPGA is in charge of the overall timing and control of the IMS system. The timing of the IMS system is very complex and must be very accurate in order to achieve a high resolving power for the IMS. The resolving power loosely refers to the ability of the IMS to differentiate between different compounds that are detected. In order to meet the strict timing requirements, a state machine was developed as a custom Intellectual Property (IP) hardware core attached to Microblaze. The state machine is designed so that changes can be made easily. The hardware core is tied directly to Microblaze via FSL. FSL is a very fast, 32 bit, unidirectional interface that has direct access to the registers in Microblaze. Because of the non-arbitrated, direct access, FSL can send data in two clock cycles. Since FSL is unidirectional two FSL busses are required for bidirectional communication, one for each direction [4].

C. Data collection and processing

In order for the IMS to be useful, the IMS signal needs to be collected and then processed. Collection and processing of the IMS signal consists of four stages:

- Preamplifier - Converts the nanoampere analog currents generated by the IMS sensor into larger analog voltages.
- Analog-to-digital conversion - Converts the analog voltage that is output from the preamplifier into digital voltages.
- Data storage - Collects digital representation of the analog voltages and stores them in memory.
- Post processing – removes noise from the signal and identifies compounds and their concentrations

C.1. Preamplifier

The preamplifier stage is achieved by the use of an inverting op-amp configuration with a large gain factor. This op-amp configuration can be seen in Figure 4.

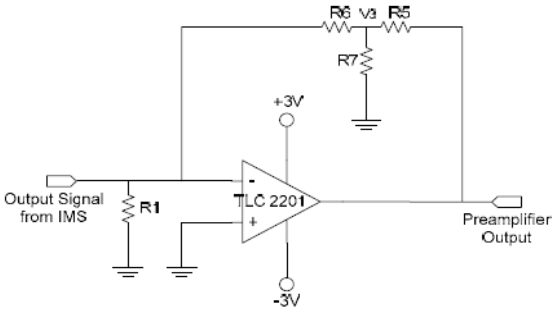

Figure 4. Preamp Circuit Diagram [5]

C.2. Analog-to-digital conversion

There are two proposed methods for the analog-to-digital (A/D) conversion process. The first method involves the use of a Delta Sigma Modulator (DSM). The DSM employs an A/D converter that uses averaging and sampling to reduce the inherent noise in the system. In a sense, the DSM acts as both a low pass filter and A/D converter at the same time. This approach can be useful at eliminating noise; however, it is difficult to ensure that the signal isn't eliminated along with the noise. The second method of A/D conversion is more conservative. This method is implemented with a traditional A/D converter. This approach would allow for any noise filtering to occur during post processing and ensure that the signal remains intact. For this project, both methods have been implemented.

C.2.1 Delta Sigma Modulator

The DSM (Figure 5) is a combination of simple integrator and a clocked comparator. The integrator constantly accumulates the difference between the input signal and the feedback signal, and the obtained signal is compared with a reference voltage to quantize the signal. The comparator is used for quantizing and the quantized signal is digitally filtered using a D-flip flop. Some of the output is fed back to the input using a digital to analog converter, which is a simple resistor in this case, for averaging the signal [5].

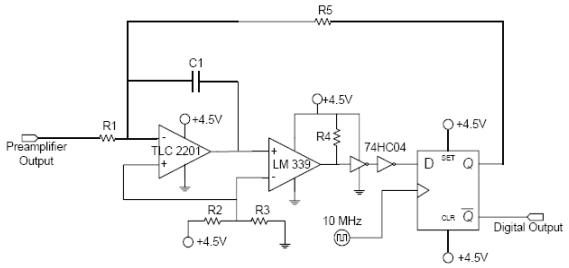

Figure 5. Delta Sigma Modulator [5]

The DSM interfaces with the FPGA via a custom 2-wire protocol. These wires are used for clock and

data. The FPGA generates the clock for the DSM. Since the FPGA has control of the clock, synchronization of data transfer is achieved by the DSM sending a new data on every rising edge of the clock. Data is collected by the FPGA one bit at a time in a moving average algorithm. This moving average algorithm keeps track of a history of data bits and outputs a single value representing a sum of these histories. The number of iterations of histories that get summed for each output is referred to as step size. The moving average algorithm acts as a low pass filter to the system and will remove any high frequency noise. The length of history that is kept will determine the effective bit resolution and sampling rate of the DSM. A longer history will result in a greater amount of filtering, more bit resolution, and a lower sampling rate. Conversely, a shorter history will result in less filtering, less bit resolution, and higher sampling rate. The step size will also have an impact on the sampling rate. The smaller the step size, the higher the sampling rate, at a slight sacrifice in bit resolution. An example of the moving average algorithm is shown in Table 1.

The custom 2-wire protocol is implemented in the FPGA as a custom hardware core. The core allows the user to alter the history length and step size according to the need of the system.

Table 1: Example of Moving average algorithm

Moving Average Algorithm										
Step Size = 9					**= 9**					**History Size**
Iter. No.	**History Content**									**No. of 1's**
	0								**8**	
1	0	1	0	0	1	0	0	1	**1**	4
2	1	0	0	1	0	0	1	**1**	0	4
3	0	0	1	0	0	1	**1**	0	0	3
4	0	1	0	0	1	**1**	0	0	1	4
5	1	0	0	1	**1**	0	0	1	1	5
6	0	0	1	**1**	0	0	1	1	1	5
7	0	1	**1**	0	0	1	1	1	1	6
8	1	**1**	0	0	1	1	1	1	0	6
9	**1**	0	0	1	1	1	1	0	1	6
Output Value = Sum of Iterations										**43**

C.2.2 Traditional Analog-to-digital converter

The traditional A/D converter used in this project is the Burr-Brown® ADS7818 has 12 bits of resolution and 500Ks/sec sampling rate [6]. Although a sampling rate of 500Ks/sec is likely not needed to achieve the necessary bandwidth for the system, this over sampling allows for decimation filtering during post processing. Decimation filtering is discussed in detail in the section *Post processing*.

The traditional A/D converter interfaces with the FPGA via a 3-wire Serial Peripheral Interface (SPI) protocol. SPI consists of a clock signal, a conversion timing signal, and a data signal. The SPI protocol is implemented in the FPGA as part of the custom IP hardware core describe earlier in the section *Overall system timing and control*. Once the hardware core collects the data from the A/D converter, it sends the data directly to Microblaze via FSL.

C.3. Data storage

The data that is passed to Microblaze gets stored in onboard SRAM for post processing and later transmission to the up-hole device. Once the data has been transmitted to the up-hole device, it is removed from SRAM.

C.4. Post Processing

Post processing of the data occurs after the data has been successfully collected from the IMS. Post processing may be performed on the FPGA prior to sending the data up-hole, or once the data has been collected by the up-hole device. Since the FPGA contains multiple embedded multiplies, the FPGA is ideal for any post processing algorithms that need to be performed down-hole.

There are two areas of post processing for this project: noise reduction, compound identification and concentration. Each of these areas is discussed in the following paragraphs.

C.4.1 Noise Reduction

In order to better identify the compound, it is important to remove as much of the noise as possible, while retaining the signal. The first method for removing noise is by taking averages of multiple scans. A series of scans are added together and then divided by the number of scans in the series. This type of noise filtering is simple and very efficient at removing random noise from the signal.

The second method for removing noise from the signal is called decimation filtering. Decimation filtering (Figure 6) is a technique that takes an over sampled digital signal and passes it through a digital low pass filter, and then down samples it. A decimation filter acts very much like an ideal filter and is much easier to implement than an analog filter with similar performance. Decimation filtering is used to eliminate known high frequencies of noise, while retaining the signal. The decimation filter can be created using the Xilinx Logicore Multiply Accumulate Finite Impulse Response v5.1 (MAC FIR) [7]. The MAC FIR is a highly configurable, highly efficient core that is included with the Xilinx ISE tools. The MAC FIR utilizes multiple Multiply Accumulate Engines to efficiently perform the many sum-of-product calculations for the decimation filter. The MAC FIR incorporates polyphase architecture to increase computational performance.

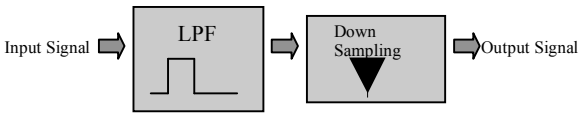

Figure 6. Decimation Filtering

C.4.2 Compound identification and concentration

Once the noise has been removed from the signal, the data is processed in order to determine the type of compounds that are detected, and the concentration of each compound. The types of compound that are detected depend upon the location in time of each peak found in the scan. When the IMS gate is open and the compounds travel toward the Faraday plate, where it gets collected and converted into a current, each compound will take a predetermined amount of time to reach the plate. Therefore, a peak detection algorithm must be implemented in order find each peak. The concentration of a compound is largely dependent upon the area under its respective peak and so this area must also be calculated.

IV. Status of Prototype

As the final field IMS will be deployed up to 50 meters into the ground, it is critical that the electronics are designed on a PCB less than 1.6" wide. This size limitation is necessary in order to fit into the probe housing, which is very small in diameter to reduce the effort required to push the IMS system into the ground. The WSU lab currently uses a pre-amplifier with a very high range of variable gain and fits within a 3"x9"x11" box. This pre-amplifier can currently be replaced with a 1.2"x2" amplifier on a printed circuit board (PCB) and has a fixed gain of 1V/1nA. Attempts may be made to increase this gain. As the signal ranges from .05nA to .8nA, an A/D converter with a 0 to 3.3nA or 0 to 3.3V range will be more than sufficient.

The laboratory IMS setup uses a PC-based National Instruments data acquisition card (DAQ) for the digitization and National Instruments Labview software for the processing of the signal. This PC-based DAQ can be replaced by the Burr-Brown® ADS7818 [6], 0 to 3.3V ADC in a very small SOIC package or the DSM. The chosen ADC will later be placed on a single PCB with the FPGA. The FPGA is in an ft256 package that is 1.7mm x 1.7mm on a 1.6"x4" (approx.) board. This package offers the freedom to use a variety of FPGA devices in the Spartan-3 and Spartan-3L family. This freedom yields great design flexibility for power and device resources including LUTs, BRAM, etc. The ft256 FPGA package will have up to 173 input/outputs (IO) [3]. Even using a 2 layer PCB, at least 80 I/O's will be accessible. Eighty I/O's should be plenty for the field IMS system, which will interface to SRAM, a 3-pin RS232 port, serial EEPROM, and an additional 20 I/O's for the IMS sensors and control interfaces. The first revision of a simplified generic FPGA PCB has been developed at BSU based on the Spartan-3 Starter Kit. However, further reduction in PCB width will be required for the final field IMS revision.

V. Future Work

Future work is required in the following areas:

- Communication with up-hole device – the communication protocol has been developed and implemented in Basic programming language and needs to be ported over to C programming language.
- Overall system timing and control – A state machine has been implemented in the FPGA for system timing and control. This system timing and control includes controlling the IMS gate controller, solenoids, and data collection circuitry. Other auxiliary sensors (humidity, temperature, pressure) have yet to be added to the system.
- Data collection and processing – Data has been successfully collected from the IMS; however, the data collection analog circuitry (pre-amplifier and A/D) needs to be optimized. Averaging of multiple scans has been implemented and tested; however, other noise filtering techniques (decimation filters or other filters) have yet to be implemented. There is still a lot of research left to be done to develop algorithms that can accurately calculate the areas and peaks. The problem gets increasingly difficult when two peaks are very close to each other and the area under each peak is blended together. These algorithms are being developed first in Matlab for proof of concept, and will later be ported over to C programming language to be run on Microblaze.
- Prototype design – A PCB has been designed and tested for the pre-amplifier. Prototype boards have been implemented for the solenoid controller, data collection circuitry, and gate controller. These components may be developed on individual PCBs, and later merged onto a single PCB. The FPGA is currently being prototyped using the Xilinx Spartan 3 starter kit board. A PCB for the FGPA is currently being developed and will later be merged with the other components on the same PCB.

V. Conclusion

An FPGA is ideal for the data acquisition and control system of the IMS system. An FPGA is low cost and small enough to meet the strict size requirements of the IMS system. An FPGA is also reconfigurable, which allows for changes to be made easily throughout the design process. An FPGA

provides the flexibility of a hardware/software co-design environment, which can maximize computational efficiency and speedup the design process. The critical timing of the IMS and computationally intense data processing algorithms can be met easily in hardware; any highly software oriented features, like the communication protocol, can be handled by Microblaze.

Reference

[1] M. Gribb, R. Walters, J. Imonigie, et al: An In-Situ Ion Mobility Spectrometer Sensor System for Detecting Gaseous VOCs in the Vadose Zone-DRAFT, to be presented at Fourth International Conference on Unsaturated Soils (UNSAT '06) Conference, April 2-6, 2006, Carefree, AZ.

[2] Xilinx Embedded Development Kit, http://www.xilinx.com/microblaze.

[3] Spartan-3 FPGA Data Sheet, Xilinx Inc., 2004-2005

[4] Xilinx XAPP529 (v1.3) May 12, 2004

[5] M. Gribb, J. Baker, et.al: Progress Report: Multipurpose Sensors to Detect and Analyze Environmental Contaminants, July 15, 2005

[6] ADS7818 12-Bit High Speed Low Power Sampling Analog-to-Digital Converter Data Sheet, Burr-Brown®, 2005

[7] Xilinx DS245, MAC FIR (v5.1) Product Specification, April 28, 2005