Proceedings of the 2006 WSEAS International Conference on Mathematical Biology and Ecology, Miami, Florida, USA, January 18-20, 2006 (pp214-220)

Rapid Prototyping of an FPGA based sensor system for Biomedical Monitoring

KIMBERLY NEWMAN, NATHAN LARAMIE, and CASEY MEDINA
Department of Engineering
University of Denver
2390 South York Street, Denver, CO, 80208
USA

Abstract: - The growing need for low-cost, pervasive health monitoring and the ability to develop computing systems in a rapid fashion have opened new areas of research and development at the intersection of engineering and biomedicine. Sensor interfacing adds a new twist to the area of designing small computing systems that enable low-power operation, greater system mobility, and user-friendly interfacing for patient self-monitoring and clinical monitoring. Additionally, the improvements in integrated circuit technology in the form of low-cost, high-density field programmable gate arrays (FPGAs) with reduced package sizes have broadened the selection of microprocessor and digital interfacing solutions. These aspects are combined in a smart sensor system that was initially developed as a temperature sensor and is evolving into a biomedical monitoring system.

Key-Words: - gate arrays, sensors, rapid prototyping, applications

1 Introduction

The advances in microelectronic fabrication have reduced the size of silicon devices to a level that ubiquitous computing is becoming a reality. Nanotechnology, micro electromechanical systems (MEMS), and micro-optical and electromechanical systems (MOEMS) are combining with the advancements in silicon to develop unique sensing capabilities for engineering applications. In the academic realm, the area of Computer Engineering is merging with this technology advancement to create novel system implementations for networking, computing architectures, and information solutions. One area that will benefit from these advancements is the implementation of health-care.

2 Background

With the improvements in sensor technology in the areas of processing, connectivity, reliability, and power consumption, the ability to develop pervasive health-care solutions will soon be commercially realized. These systems require a comprehensive approach from both the engineering and medical communities to ensure product reliability, safety, and accuracy. Many simple solutions have been developed in this area for specific issues related to health monitoring of various conditions [1]. One example is a home-monitoring system that notifies a central source when an emergency has occurred [2].

Additionally, the combination of computing and wearable sensors has vastly increased the amount of data that can be gathered concerning the status of a patient [3]. Several wearable sensor systems are already in the clinical testing phase for commercial implementation. These sensors were implemented in unobtrusive garments such as rings and shirts [1]. From the diagnostic and medical side, the data obtained from wearable smart sensor systems for medical applications can be incorporated real-time with an expert system that is customized to the known and probable conditions associated with a given patient. These types of systems are often referred to as smart sensor systems.

A major advantage of a smart sensor system is the ability of a patient to self-monitor and check in with a trained professional if an abnormal condition is detected [4]. For practicality, these systems require a means of wireless communication with a centrally located monitoring station so that the wearer is not tethered to a fixed location. Possible modes of wireless communication include Bluetooth and the newly developed IEEE 802.15.4. These wireless standards provide a comparatively low-cost, low-power solution at the expense of bandwidth [5].

Inevitably, the designer of a sensor system to be used in the biomedical field must consider the user's ability to interface with the system as well as the environment in which it will be applied. In order to achieve this task, the seven principles of universal design were established in order to address issues such as ease of use, ease of comprehension, and error tolerance and recovery. These guidelines allow for the development of a sensor system able to accommodate the widest range of users [6].

In addition to the guidelines established for design there are requirements for usage. Wearable sensors for medical monitoring applications have two main modes of operation: wellness and disease management, and remote monitoring for independent living [4]. In both cases, it is necessary that the sensor be customizable to the individual needs and abilities of the user. For wearable sensors for medical monitoring applications, it is preferable to provide the user with self-monitoring capabilities to enhance the user's comfort with and confidence in the system's performance with respect to the monitoring of stable conditions and the detection of abnormal conditions. Additionally, remote monitoring systems allow medical personnel to monitor a patient's condition. In this case the user is not burdened with self-monitoring and, if necessary, emergency services can be rendered [4].

Another issue that is critical to the success of the adoption of a smart sensor system is the ability of a minimally trained caregiver to interface with the technology. A recent study has shown that increasing the ratio of patients per nurse can increase the likelihood that a patient will die [9]. Unfortunately, an increasing number of medical institutions are finding this ratio reaching dangerously high levels. With such severe understaffing problems, medical professionals are left with insufficient time to be trained in the operation of complex remote sensor systems. Likewise, some users (i.e. elderly) may be unable or unwilling to understand the basic operation of a complex system [4].

There are many challenges associated with the development of a system that integrates available technologies for medical applications. From the medical perspective, the accuracy of the data, the ability to perform a diagnosis, and the capability of an observer to assist a patient are critical issues. Collaboration with professionals in the medical field is vital to the success and adoption of these systems. From the technology side, a short list of smart sensor design issues includes the method of collection, processing, and storage of the data, the networking implementation of a mobile sensor system, the display of information for monitoring and care, the reliability of the system, and the power requirements of a complex system.

3 System Implementation

The remainder of this discussion focuses on the development of a low-cost prototype system for temperature measurement that includes a fixed and mobile monitoring unit. An overview of the smart sensor system prototype is shown in figure 1. The system is composed of a wearable unit that contains the sensor, signal converter, digital computation device, status display unit and communication interface. A separate monitoring station is also shown that displays the readout of the sensor, controls the sampling rate of the sensor, controls set points for alarm conditions, verifies the operation of the communication interface and provides feedback to the wearable unit if alarm conditions are detected.

Research is progressing in the areas of wireless networking, intelligent display methods for care-giving, system reliability, power management, sensor-to-sensor communication, parallel sensor implementation, and the ability to monitor additional vital statistics.

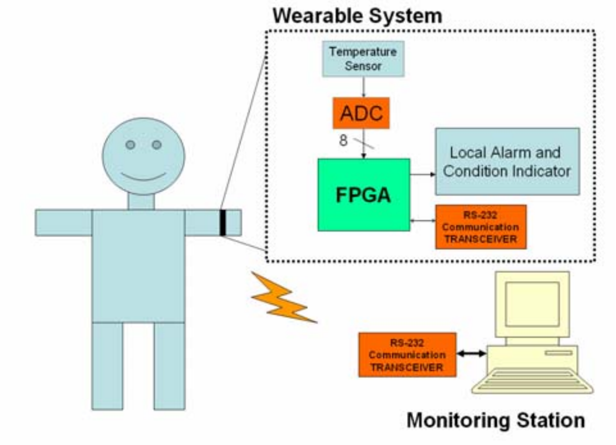

Wearable System
Temperature Sensor
ADC
8
Local Alarm and Condition Indicator
FPGA
RS-232 Communication TRANSCEIVER
RS-232 Communication TRANSCEIVER
Monitoring Station

Fig. 1: Smart Sensor System Overview

For the rapid prototyping of smart sensor systems, it is favorable to select a versatile hardware platform. The utilization of programmable devices such as field programmable gate arrays (FPGAs) and hardware design languages such as Verilog enable the rapid prototyping of the wearable portion of these systems for use in both laboratory and clinical environments.

The advantages of System-on-a-Chip (SoC) development on an FPGA platform have been noted. For instance, the flexibility of FPGAs allows a designer to quickly produce a low-cost, modular system. Moreover, the ability to reconfigure the FPGA allows for numerous advantages: the FPGA may be configured as a Built-in Self-Test (BIST) controller for the entire system with no additional overhead, the FPGA may be configured to adapt to different protocols and hardware on demand, and the FPGA may be upgraded remotely [7]. This is particularly advantageous when one considers the design and development of smart sensors as new standards such as IEEE 1451 emerge [8] as well as instances where the application of the sensor demands different protocols and hardware. The parallel resources offered by FPGAs also facilitate the parallel monitoring of a variety of vital signs from multiple sensors.

From the clinical side where an observer is interfacing with the smart sensor system worn by a mobile user, several design aspects should be taken into consideration. For the development of a friendly Graphical User Interface (GUI) for the remote monitoring of the smart sensor system, a method that saves training and implementation time is highly beneficial. The LabVIEW software from National Instruments was used for this process. Using LabVIEW in such applications is not new, for example, in one study, a group of researchers from Graftek France used LabVIEW to develop a GUI for a system designed to monitor eye motion [10]. LabVIEW accelerates the design cycle with a simple, visual programming environment. This environment enables intuitive testing and verification as well as a straightforward expansion of programs. Similarly, LabVIEW provides a variety of options for creating simple user-friendly interfaces and functions for communicating with external devices.

The combination of the FPGA technology with the LabVIEW software platform provides a simple and powerful method to develop and prototype smart sensor systems. The remainder of the components required to implement the prototype temperature system were selected based on the ease of use and commercial availability. A low-cost temperature sensor was selected in addition to an analog-to-digital converter (ADC) for sampling and interfacing to the digital system.

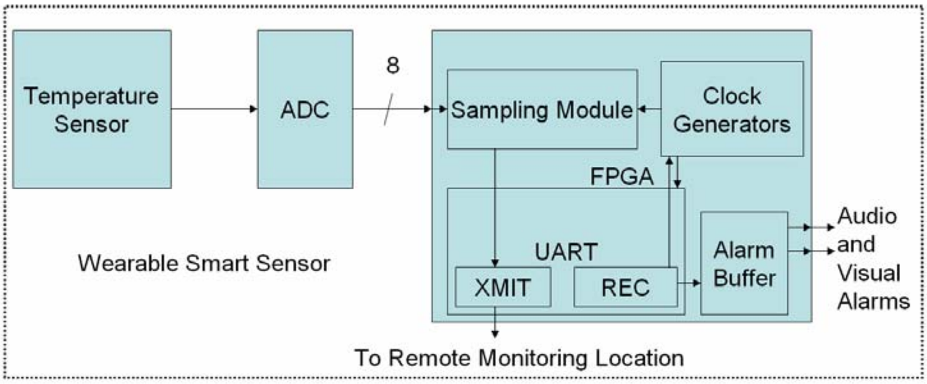


Fig. 2: Wearable Smart Sensor System Overview

A smart sensor prototype that monitors temperature, as illustrated in figure 2, was successfully realized by integrating the FPGA hardware platform with the LabVIEW software platform. The prototype uses the analog output of the sensor to monitor the temperature of the user. This signal is converted into a digital value with an ADC. An FPGA is utilized to sample the data from the ADC at a user-selected rate. The mobile portion of the smart sensor consists of the temperature sensor, the ADC, and the FPGA. The data from the FPGA is transmitted serially to a fixed monitoring station consisting of a PC with the LabVIEW GUI. The LabVIEW GUI is designed to be an intuitive user-interface that provides feedback to the mobile system including the selected sampling rate and alarm status. The LabVIEW GUI allows for remote, real-time monitoring of the smart sensor as well as control of the performance of the smart sensor.

A low-cost, low-power, commercially-available ATP 1000 TEMPLATE Decora Temperature Sensor was used in the prototype. This sensor has a voltage output that corresponds to the temperature in Fahrenheit where 10 mV corresponds to 1 ºF. This voltage to temperature relationship enables a simple conversion of the digitized temperature value based on the selected range of the ADC. The present configuration of the temperature unit in the prototype consumes 1 mW. This sensor is connected to a commercially-available ADC. An ADC0801 from National Semiconductor was utilized for the ADC unit. Low-power variations of this unit consume as little as 39 mW. Additionally, the 8-bit output of this particular ADC unit is a common word size that is easily manipulated in LabVIEW.

The FPGA is programmed with a bit file generated by a number of Verilog files. These files enable the data to be sampled from the ADC at a user-specified rate. The user-specified clock rates are generated with a network of clock divider modules realized in Verilog. These rates are easily modified and are only dependent on the speed of the system clock oscillator and the length of the desired clock period; therefore, these modules require only slight modifications across FPGA platforms. Similarly, the FPGA transmits the data at the user-specified sampling rate via a Universal Asynchronous Receiver Transmitter (UART) module realized in Verilog. The receiver and transmitter portions are controlled by clocks that are generated with the same type of clock divider modules that generate the sample clock. The speed of the transmitter clock (baud rate) is also easily modified and is constrained only by the speed of the system clock oscillator and resources available. The receiver module of the UART receives the user-specified feedback generated remotely by the user from the GUI. The feedback includes the selected sampling rate and the presence of alarm conditions. This feedback is stored to a buffer that controls the selected sample clock and alarm indicators respectively. The FPGA places signals to external pins upon the detection of an alarm condition that enable both sound and light indicators. These features allow the wearer to self-monitor by examining the smart sensor itself.

Qualities like word size are localized to specific segments of code; therefore, the use of Verilog allows for versatility in the external hardware. For example, the Verilog code can be modified to communicate with an ADC with a greater word length by updating selected parameters. Similarly, there is flexibility in the word length of the UART through modification of localized parameters. The modular quality of Verilog facilitates modifications in certain aspects of the system configuration with minimal side effects to the system implementation. The availability of additional configurable I/O also enables the expansion of the system for expanded parallel applications of the system. For example, other communication protocols may be used simply by replacing the UART module with the desired module and modifying parameters as necessary. Additionally, the parallel resources available in FPGAs along with the modular nature of Verilog enable the implementation of parallel applications of the smart sensor by reusing existing modules to increase the number of sensors that may be monitored by the system.

For the monitoring and control station of the smart sensor system, the GUI prototype was designed to accommodate a wide range of potential users. The user has the ability to customize the performance of the system through four parameters: sampling rate, upper and lower thresholds, and data file path. The GUI also gives the user real-time displays including a real-time strip chart, digital temperature readout, and three color-coded visual alarm indicators. Because of the intuitive layout of the GUI, the system allows the wearer or operator to monitor any number of vital signs from the monitoring location. The present prototype enables the user to monitor temperature. Only slight modifications are required to allow the interface to accommodate different sensors and/or multiple sensors in parallel.

The LabVIEW GUI on the monitoring station establishes two-way serial communication with the FPGA hardware. First, an ASCII character with the appropriate hexadecimal representation is written to the FPGA. This character represents the desired sampling rate and the presence of any alarm conditions. The prototype monitors three alarm conditions: upper and lower temperature thresholds exceeded and communication lost. The visual indicators are color coded to allow for quick analysis of any anomalous conditions with minimal training. The interface reads current data from the FPGA through the RS-232 serial connection. This data is interpreted as a character string that must be converted to a numerical value and conditioned using a sensor-specific formula. The conditioned data is then displayed for the operator in both graphical and digital formats. The data is compared with the upper and lower thresholds and the feedback to the mobile system is calculated. The data is also archived in a spreadsheet file designated by the operator. The feedback from the GUI is a single ASCII character that is sent to the mobile system serially. The feedback character is selected based on the sample rate requested by the user on the GUI and the presence of alarm conditions. Loss of communication is detected when there is no reception of data from the mobile system within a specific time frame that is equal to three times the selected sample rate.

Because a single ASCII character is written to the mobile system, it is simple to equate specific characters to a range of particular conditions. This feature gives the designer flexibility in the type of feedback sent to the mobile system. The LabVIEW platform allows the designer to add new features with minimal side effects to existing features, to accommodate different types of data, and to expand the program to accommodate additional sensing capabilities.

4 RESULTS

The FPGA on the mobile system samples data from the ADC at a user-specified sampling rate. The available sampling rates include 1 second per sample and 10 seconds per sample through 60 seconds per sample in increments of 10 seconds. This data is sent to the remote monitoring location via a UART module on the FPGA. The FPGA receives feedback via the UART from the remote monitoring location; feedback includes the user-selected sampling rate and alarm conditions. Figure 3 shows the schematic diagram of the customized FPGA module. The key features are the 8-bit ADC interface (adc_in) pins, the UART transmitter (data_in) and receiver (data_out) pins, the dedicated output pins for sound and light alarm indicators, and the system clock pin (sys_clk). This module consumes 550 slices in the Spartan 2 FPGA from Xilinx, Corporation. The present system clock signal is generated by a 50 MHz oscillator. The present I/O requirements consist of 10 critical pins for input and 3 critical pins for output. The asynchronous active low input (resetn) is an optional input. Similarly, the 8-bit sample value (Sample_dat) and led toggle (ldg) are optional output pins that allow the user to display the present digitized sample.

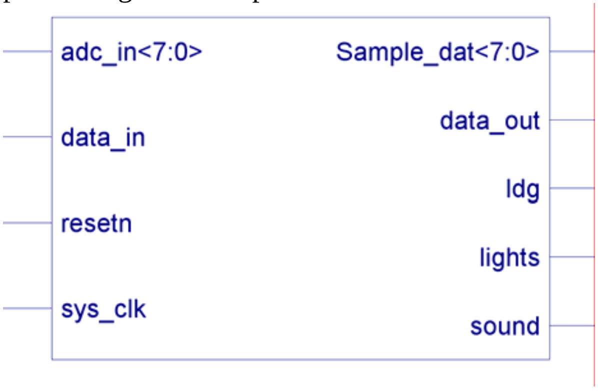
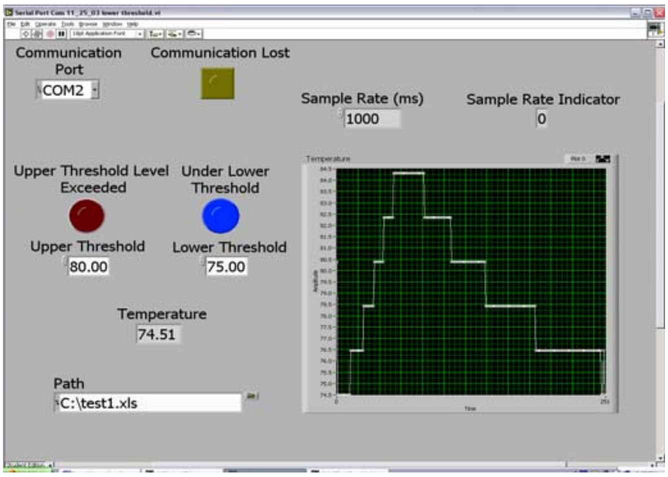


Fig. 3: Block Diagram of Verilog Module for Mobile FPGA implementation of Smart Sensor Interface

The other major component of the system is the GUI shown in Figure 4. This interface establishes two-way communication with the sensor system and provides real-time access to sampled data. Visual alarm indicators are used to provide a quick analysis of a patient's status; alarms include thresholds exceeded and communication lost. In the prototyping environment, the operator is allowed to set threshold conditions and sampling rates. This feature may be hidden from the wearer with minimal modifications if desired. The system allows the wearer to self-monitor with a variety of intuitive real-time displays.



Communication Port: COM2

Sample Rate (ms): 1000

Sample Rate Indicator: 0

Upper Threshold: 80.00

Lower Threshold: 75.00

Temperature: 74.51

Path: C:\test1.xls

Fig. 4: The Graphical User Interface Screen for the Smart Sensor Monitoring Station Developed in LabVIEW

As seen in Figure 4, the operator has only four major parameters to control: sample rate (given in units of milliseconds), alarm conditions (Upper Threshold and Lower Threshold), data storage location (Path) and communication port specification. The remaining display consists of visual alarms and a real-time graphic display of the conditioned sensor data that facilitate self-monitoring. The LabVIEW application transmits updated sample rate and alarm condition settings each time data is received from the sensor.

Table 1 shows the format of the archived data. The data presented in Table 1 is abbreviated and illustrates four of the temperature changes demonstrated on the graphical display in figure 4. The first column represents the CPU time and is given in units of milliseconds. The second column represents the conditioned temperature data received from the FPGA. The third column of the output file represents the sample rate. A Sample Rate Indicator (SRI) that ranges from 0 to 6 is used to indicate the rate. The number corresponds to the sampling frequencies that range from 1Hz to 16.6 mHz respectively. This number represents the selected sample rate for each respective measurement. It allows the sampling rate to be identified from the archived data. This is important as the GUI allows the operator to modify the sample rate at any time; therefore, the sample rate tags allow the user to correctly interpret the data.

Table 1: Abbreviated Sample Data Logged into Spreadsheet File

CPU Time	Temperature(ºF)	SRI
353880448	74.51	0
353880704	74.51	0
353881696	74.51	0
353891680	76.471	0
353892704	76.471	0
353893696	76.471	0
353903680	78.431	0
353904704	78.431	0
353905696	78.431	0
353913696	80.392	0

5 Conclusion

The FPGA platform provides an environment for rapid prototyping of smart sensors for biomedical applications with numerous advantages. Among these advantages is the flexibility to add features and provide system updates and support on demand. This capability is critical when developing sensors for use in biomedical applications due to the array of standards and individual wearer abilities to which the device may need to adapt as well as the need to rapidly produce different types of sensors or expand the capability of current sensors.

LabVIEW provides an expandable means of communication between the sensor and monitoring system. It also presents a rapid prototyping tool for the monitoring of remote sensor systems. The built-in features of LabVIEW reduce the design cycle for the development of a GUI. The creation of user-friendly GUIs for data management is critical for both patients who want to self-monitor and healthcare professionals who require data to be presented in a simple form.

Further research is being conducted in order to increase the resolution of the prototype sensor with alternative hardware and system configurations. Additionally, research is underway to increase the stability of the sensor with modified sampling techniques as well as more comprehensive handshaking techniques and error detection methods. The use of multiple sensors and different types of sensors is under investigation.

Because it is desirable to ultimately produce smart wearable sensors, different and competing modes of wireless communication are under consideration. These modes include Bluetooth and IEEE 802.15.4. Similarly, power consumption and the size of the devices used are concerns for the implementation of a wearable system. These issues will be addressed as the project continues.

References:

[1] P. Bonato. “Wearable Sensors/Systems and Their Impact on Biomedical Engineering,” *IEEE Engineering in Medicine and Biology Magazine*, pp.18-20, (May/June 2003). [2] J. Winters, Y. Wang, J. Winters. “Wearable Sensors and Telerehabilitation,” *IEEE Engineering in Medicine and Biology Magazine*, pp.56-65, (May/June 2003).

[3] T. Martin. “Wearable and Ubiquitous Computing,” *IEEE Pervasive Computing*, Vol. 2 No. 3 pp. 8-12 (July-September, 2003).

[4] I. Korhonen, J. Parkka, M. Van Gils. “Health Monitoring in the Home of the Future,” *IEEE Engineering in Medicine and Biology Magazine*, pp.66-73, (May/June 2003).

[5] R. Ashok, D. Agrawal. “Next-Generaton Wearable Networks,” *IEEE Computer Magazine*, Vol. 36 No. 11 pp. 31-39 (November, 2003).

[6] M. Gancy, D. Ross, T. Starner. “Universal Design: Lessons for Wearable Computing,” *IEEE Pervasive Computing*, Vol. 2 No. 3 pp. 19-23 (July-September, 2003).

[7] M. Abramovici, C. Stroud, M. Emmert. “Using Embedded FPGAs for SoC Yield Improvement,” *Proceedings of the Design Automation Conference (DAC)*, New Orleans, LA pp. 713-724 (June, 2002).

[8] K. Lee. “IEEE 1451: A Standard in Support of Smart Transducer Networking,” *IEEE Instrumentation and Measurement Technology Conference*, Baltimore, MD (May, 2000).

[9] L. Aiken, S. Clarke, D. Sloane, J. Sochalski, J. Silber. “Hospital Nurse Staffing and Patient Mortality, Nurse Burnout, and Job Dissatisfaction,” *The Journal of the American Medical Association*, Vol. 288 No. 16 (October, 2002).

[10] P. Sauvan-Magnet, “IMAQ and LabVIEW Automate the Study of Eye Motion” *Graftek Imaging, Inc.* www.ni.com