FPGA Based System design suitable for Wireless Health Monitoring Employing Intelligent RF module

Khalil Arshak and Essa Jafer
Department of Electronic and Computer Engineering
University of Limerick
Limerick, Ireland
Khalil.arshak@ul.ie, Essa.jafer@ul.ie

Christian Serge Ibala
CAE Logic Drive
XILINX
Dublin, Ireland
Christian.Ibala@Xilinx.com

Abstract— **The aim of this study is to design and develop a wireless sensor system integrating 24 bits analog to digital converter (ADC) and long range RF transceiver unit. The system is reading from two types of sensors, pressure and temperature. The serial communication between the ADC unit and RF transceiver is controlled by Spartan-3 FPGA device. A number of tasks are implemented by the FPGA device like interfacing, buffering and compression. A capacitive interface circuit has been employed for the pressure sensors and it is based on the principle of capacitance-frequency-voltage conversion. The RF transmitter is communicating with the FGPA through handshaking protocol and sending the data through ISM band. The pressure data will be compressed to avoid sending any repeated values aiming further reduction in the power consumption.**

I. INTRODUCTION

FPGAs devices are nowadays one of the most important alternatives to construct high-speed digital systems. This technology was marketed in the middle of the 1980s with a simple but strong argument: Its capability to be in-house erased and reconfigured in few milliseconds would allow the designers to correct errors or introduce last-minute modifications. This feature clearly distinguished FPGAs from other alternatives like standard cells or gate arrays, and guaranteed the success of the new devices [1]. As FPGA-based Systems-on-a-Chip (SoCs) are getting more popular, many of the issues regarding sensor monitoring need to be tackled [2,3].

On the sensor side the most important I/O modules are the analog inputs that accept the voltages generated by the sensors. These are then measured by an integrated Analog-to-digital converter for subsequent processing by the CPU and transmission towards the host. In addition, microcontrollers provide a number of digital I/O pins that can be controlled, or queried, under program control. They can also be associated with additional internal modules to perform dedicated functions. A typical example is a PWM (Pulse-Width Modulation) generator that is often used to control the speed of a motor. This is usually implemented by using a timer module whose value is constantly compared to a register that contains the desired pulse-width. The (binary) result of this comparison then leaves the chip directly through a dedicated pin which is connected to the motor circuitry. Now, while the above approach is fine for many smaller projects, it is not uncommon that an instrument or installation design requires more than just a few PWM or other digital I/O modules. What's more is that the timers are also required for other tasks, e.g. generating the MIDI clock. Unfortunately, simply switching to a bigger microcontroller only helps to a certain extent as the focus is usually on more memory and general purpose pins rather than I/O modules. And adding more microcontrollers complicates the overall system design as they have to communicate with each other.

FPGAs, in contrast, do not have these limitations. The building blocks of these devices are gates, the logic foundation of any digital device. However, rather than wiring up countless TTL chips like in the early days, the designer can enter the schematic with a suitable editor, which is then compiled into a configuration file for the FPGA [4]. This means that before programming the microcontroller one would first design it according to the requirements. While this may seem like re-inventing the wheel, modern design tools hide a lot of the complicated details while the added flexibility outweighs the extra effort. More to the point though, we will see that many components of the microcontroller can be left out of the design.

In this paper, the design of a long-range wireless system prototype used for health multi-sensor monitoring will be presented. The system meets timing and low power constraints, thus a configurable state machine has been developed to make the system adaptive with the variations in the sensors due to environmental factors

II. SYSTEM OVERVIEW

The wireless system is consisting of two main units at both the transmitter and receiver sides as shown in Figure 1. At

the transmitter, the data recorded by two sensors, pressure and temperature, will be first converted to digital form using the multi-channel 24 bits ADC (ads1224) from Texas Instruments [5]. Then an FPGA device will read the data serially and implement different processes like buffering, compression and framing before send it to the FSK transmitter unit. A second FPGA will be used to control the receiver side and it is responsible for processing the data inversely like de-framing and de-compressing. Spartan™-3 device from Xilinx [6] has been used in the design since it meets our requirements and has enough resources. Nordic NRF905 multiband transceiver has been selected for the prototype [7], since it has a very small size with other useful features

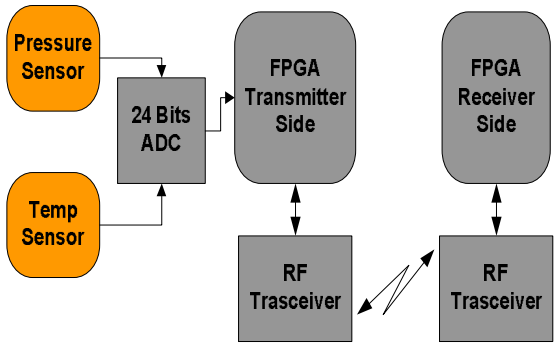

Figure 1. System overview block diagram

The design blocks of the transmitter side FPGA are shown in Figure 2. The ADC interface block has been designed to implement the Serial Peripheral Interface (SPI) protocol. The data will be received through the Din pin, where the two control signals, En and Mux are used to enable the ADC chip and select the required analog channel. Two Digital Clock Manager (DCM) units have been used to provide the ADC with a free fclk and non-free sclk running clocks at different speeds.

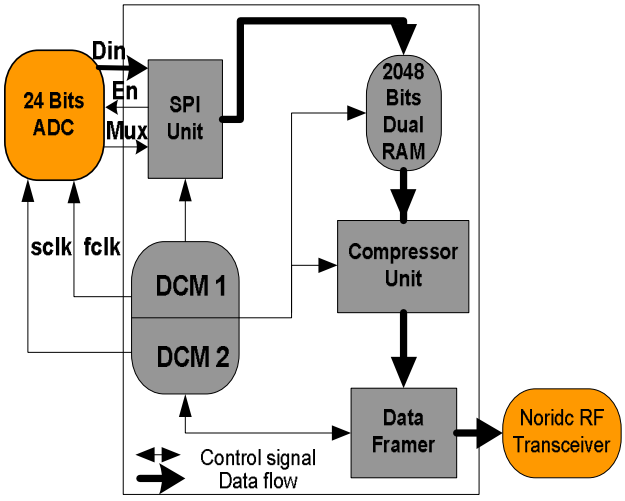

Figure 2. Building blocks of the transmitter FPGA

The data collected by the SPI will be buffered and then compressed. The Compressor unit has been designed especially to suit the nature of the two sensors signals and reduce the amount of the transmitted data to achieve low power consumption. The Framer is responsible of interfacing with the Nordic transceiver through custom designed asynchronous protocol.

III. ASYNCHRONOUS HANDSHAKING PROTOCOL

The data between the FPGA and the transceiver will be transferred in a form of 4 bits at a time from each byte using a fully asynchronous protocol. The Least Significant Bits (0 to 3) is transferred first, followed by the Most Significant Bits (bits 4 to 7). Two pairs of handshake lines, data_ready and data_receive control the flow of data in each direction as Figure 3 presents.

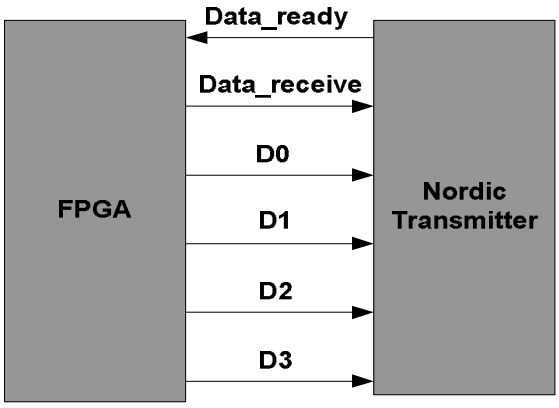

Figure 3. FPGA to Nordic Transmitter connection

The packet transferred between the two devices consists of 32 bits, where the first three bytes represent the data and the last byte carries information regarding the sample nature whether its pressure or temperature. The sequence for data transfer is as follows:

1. FPGA asserts Data_ready high to initiate transfer

2. FPGA set the data lines to output and place 4 bits on the data lines

3. Transceiver asserts Data_receive high and read the 4 bits.

4. FPGA asserts Data_ready low and then transceiver asserts Data_receive low to start another transmission.

The above protocol characterised to be simple in implementation and fast. The reason behind sending only 4 bits at a time is because of the available number of the transceiver input/output (I/O) ports that can be used for this purpose. In order to test the performance of this handshaking protocol, an efficient test-bench has been written to mimic the behaviour of the transceiver and verify the operation of the Framer block. A sample of the output waveforms are shown in Figure 4. It can be seen from the figure that the protocol is working perfectly to transfer arbitrary generated data.

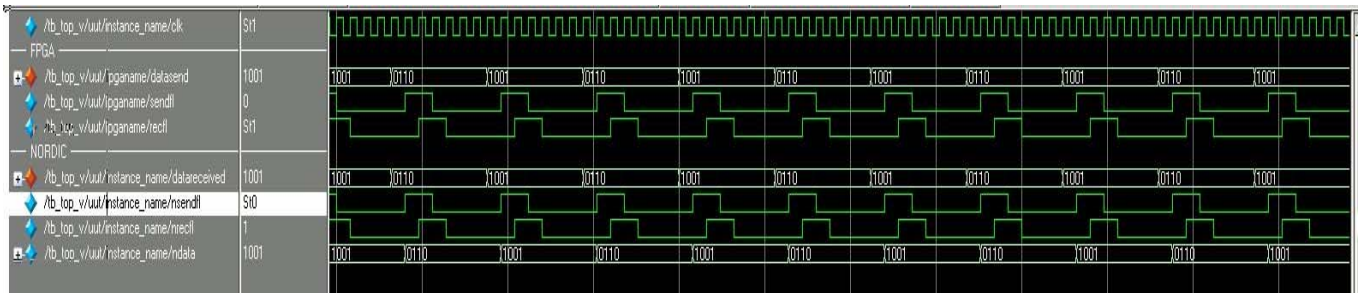

Figure 4. Testbench waveforms of the transmitter handshaking

IV. COMPRESSION

The compressor presents the main core of the FPGA system, where data will be buffered and compressed based on a specific assumptions. As the medical pressure data expected to change more rapid than the temperature, the control state machine will direct the block to read a certain number of pressure and one temperature sample each time if the pressure doesn't change. When a change in the pressure is detected, then temperature will be read as well to see the correlation between the two signals. The finite State Machine (FSM) of the compressor has been extracted in Figure 5 using the code coverage.

The compression will be applied on the repeated samples of a single value by replacing them into a single instance of symbol and a run account. Finally the data will be structured into a packet format as shown in Figure 6. Only one bit will be used to indicate the data type whether its pressure or temperature.

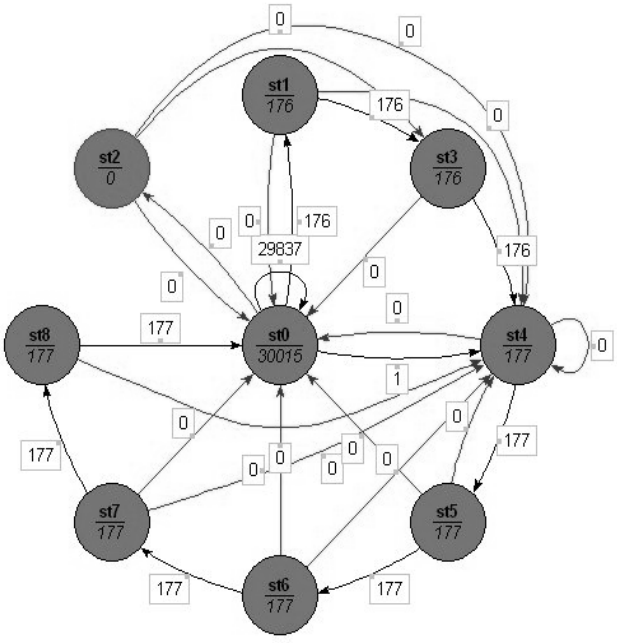

Figure 5. Transmitter FPGA-Compressor unit FSM

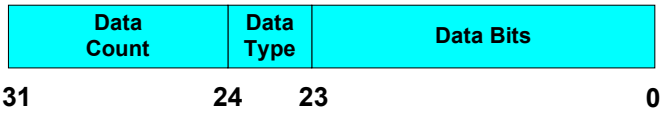

Figure 6. Structure of the sent 32 bits data packet

. It has to be mentioned that the performance of the compressor block was found to be crucial to the overall power consumption of the transmitter side system which is required to be low.

V. SYSTEM HARDWARE DEBUGGING

Chipscope technology provided by XILINX was introduced to test the functionality of the FPGA blocks [8]. The ChipScope program consists of two basic debugging tools. These are Core Generator and Core Inserter. In this project, core generator has been used because of the need for using the VIO (virtual input/output) core that is not available in the inserter flow. The main three cores used for debugging different parts of the digital design are: Integrated Logic Analyzer (ILA), ICON (Integrated Control) core and VIO core. Figure 7 illustrates the interaction of the three mentioned cores with the FPGA design

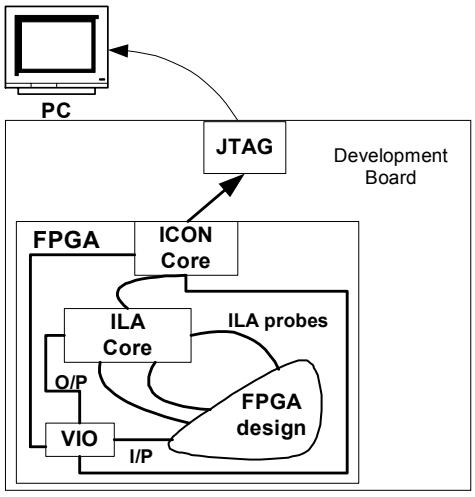

Figure 7. Chipscope interface with FPGA system

Figure 8 presents sample from Chipscope waveform results where a certain value has been shifted through the SPI register and captured by the Integrated Logic Analyzer. In the same way, the handshaking protocol has been verified for both transmission sides as in Figure 9, where the sending time intervals have been marked.

VI. CONCLUCIONS

In this paper, the design of FPGA based multi-channel wireless sensor system has been presented. A novel approach has been developed to integrate two slow varying sensors and RF communication channel on single FPGA chip. The main blocks of the FPGA transmitter side were designed to effectively handle the data flow and reduce sending any unnecessary information. In addition the system has been made adaptive to the sensors variations due to environmental factors.

ACKNOWLEDGMENT

This work was supported by the Enterprise Ireland Commercialization Fund 2003 as part of the MIAPS project, reference no. CFTD/03/425.

REFERENCES

[1] Sergio, L, B., Javier, G., and Eduardo I, B., "Dynamically inserting, operating, and elemnating thermal sensors of FPGA based system", *IEEE Trans. Components and Packaging Technologies*, Vol. 25, 2002, pp. 561-566.

[2] Velusamy, S., Wei Huang, Lach, J., Stan, M., Skadron, K., "Monitoring temperature in FPGA based SoCs", International Conference Proc. on Computer Design, Oct. 2005, pp.634 – 637.

[3] Sagahyroon, A., Al-Khudairi, T., "FPGA-based acquisition of sensor data", IEEE International Conference Proc. on Industrial Technology, Vol.3, Dec. 2004, pp. 1398-1401.

[4] S. Kartadinata, "The gluion advantages of an FPGA-based sensor interface " presented at the 2006 conference on New interfaces for musical expression Paris, France pp. 93-96 2006.

[5] ADS1224 24-bits Capacitance to Digital Converter data sheet, Texas Instruments, http://www.ti.com.

[6] XILINX, Inc., "Spartan 3, FPGA FPGA Family", Complete data sheet, 2006.

[7] Nordic VLSI nRF905 Multiband Radio Transceiver data sheet, nRF905rev1_3, http://www.nvlsi.com.

[8] K. Arshak, E. Jafer, and C. Ibala, "Testing FPGA based digital system using XILINX ChipScope logic analyzer," presented at 29th International Spring Seminar on Electronics Technology (ISSE '06), St. Marienthal, Germany pp. 355-360, 2006.

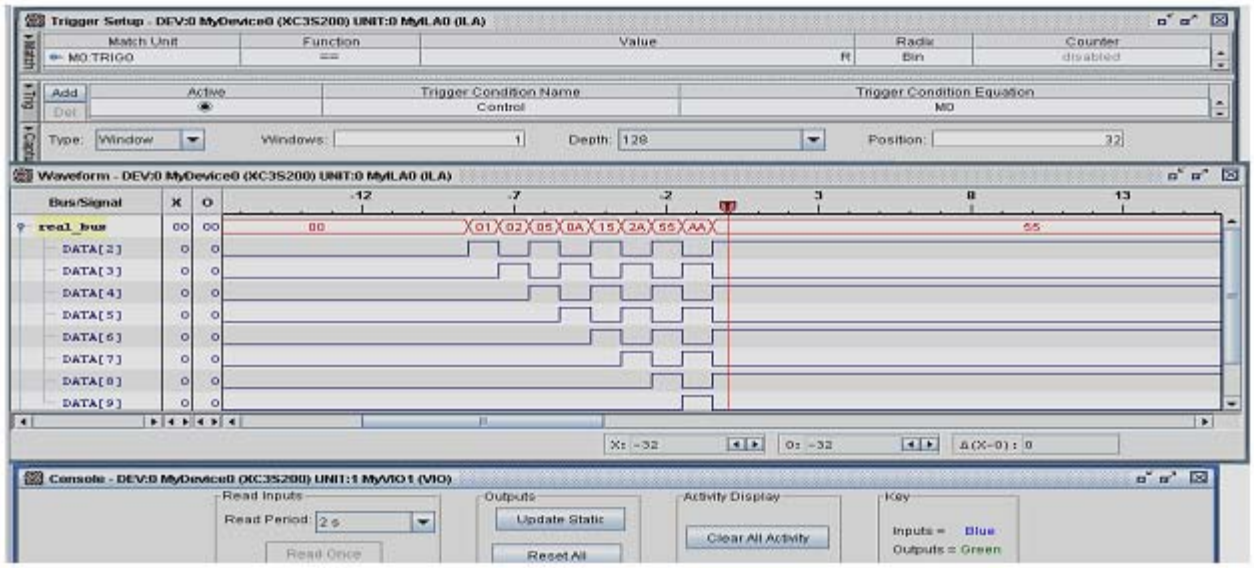

Figure 8. ChipScope trigger and waveform window of the SPI debugging

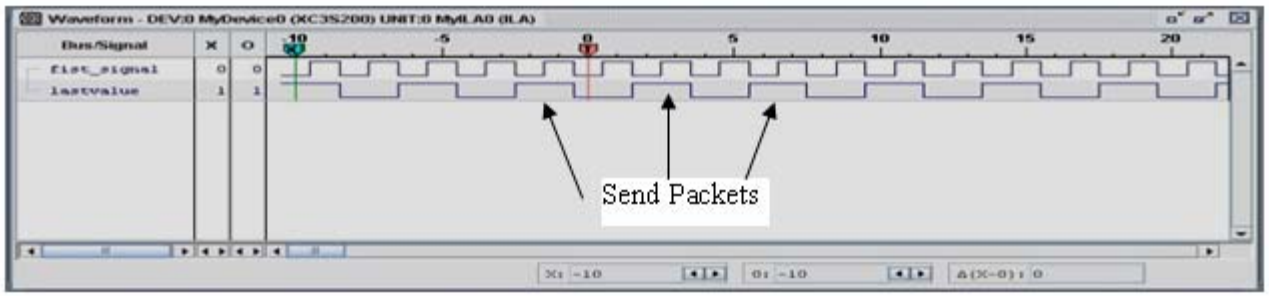

Figure 9. Pressure Sensor response using the wireless prototype