ERNEST JAMRO
Department of Electronics, AGH – University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
Academic Computer Center CYFRONET, AGH-UST
ul. Nawojki 11, Kraków 30-950, Poland
jamro@agh.edu.pl

FPGA Implementation of High Speed Diagnostic Systems

Abstract

This paper presents the hardware solution denoted as Programmable Unit for Diagnostic (PUD) based on Filed Programmable Gate Arrays (FPGAs) adapted for diagnostic systems. The sampling frequency of the input analog signals and digital signals processing speed of the PUD is high beyond comparable DSP based systems.

Keywords*: FPGA, DSP, high speed computation, real-time diagnostic systems.*

Introduction

Electronic implementations of signal acquisition, digital signal processing and finally diagnostic procedures are often disregarded in academic considerations. Nevertheless they often determines the diagnostic procedure and final results. Therefore in this paper the electronic solutions for high speed diagnostic devices will be considered.

Usually two different electronic solutions are considered:

1) General-purpose processors or digital signal processors (DSPs)

2) Field Programmable Gate Arrays (FPGAs)

General purpose processors (e.g. Pentium) or DSPs are very flexible, easy to be programmed and relatively cheap nevertheless the signal sampling frequency and signal processing speed is relatively small. Consequently this solution is only suitable for relatively slow real-time diagnostic systems, for which signal sampling frequency is below 100kHz. The given threshold frequency differs for different algorithm complexity, number of channels, processor

computation power and so on, therefore the 100kHz should be regarded only as a rough number.

For diagnostic systems for which data sampling frequency is above roughly 100kHz the FPGA solution is the best one. FPGAs can be programmed by an end-user in similar way as microprocessors, nevertheless they employ different design cycle which causes that they can be designed only by electronics engineers. For FPGAs, sampling frequency is up to roughly 500 MHz, similarly data processing speed is also very high.

There are many signal-acquisition and digital signal processing devices based on FPGAs e.g. [1, 2], nevertheless they are either not dedicated for diagnostic, require PC connection or do not use hardware-software co-design approach. Therefore the main thought throughout this paper is to present a FPGA solution, its possibilities and design cycle, and finely to describe the developed FPGA-based system.

FPGAs and Digital Signal Processing

A Field Programmable Gate Arrays (FPGAs) e.g. [3] device is roughly a semiconductor device containing programmable logic components such as AND, OR, XOR, NOT gates or more complex functions such as adders, multipliers, memories. This programmable logic elements are connected by programmable interconnects. In fact, in most cases, FPGAs do not contain standard gate logic – they incorporate relatively small memory cells (usually 16×1), denoted as Look-Up Tables (LUTs), which performs any logical function of up to 4 inputs. These logical functions can be grouped to form either Finite State Machines (FSM) suitable to control external components, e.g. Analog Digital Converters (ADC), memory. A (soft) processor e.g. MicroBlaze [4] is an example of a very complex FSM which can also be implemented in FPGAs. Besides these LUTs can be used to form arithmetic modules such as adders, multipliers, etc.

A fundamental signal processing procedure is filtering, therefore an example of implementation of a filter in FPGAs is given hereby. Finite Impulse Response (FIR) filters execute the following equation:

$$y(i) = \sum_{k=0}^{N-1} h(k) \cdot x(i-k) \qquad (1)$$

where: *x(i)*, *y(i)* – input and output samples, *i* – the index of a sample, *h(k)* – FIR filter coefficients, *N*- the number of filter taps.

For Digital Signal Processors (DSPs) every filter tap requires at least a single clock cycle [5]. Nevertheless for number of filter taps *N= 100* and sampling frequency *f= 500 kS/s*, the required DSP clock frequency is well beyond 50 MHz. It should be noted that several independent channels are often processed and the DSP is also occupied by other activates (e.g. input data acquisition). Summing up, most DSPs are not capable of processing high speed input signals. Consequently an alternative solution - FPGAs is more and more often adopted for digital signal processing.

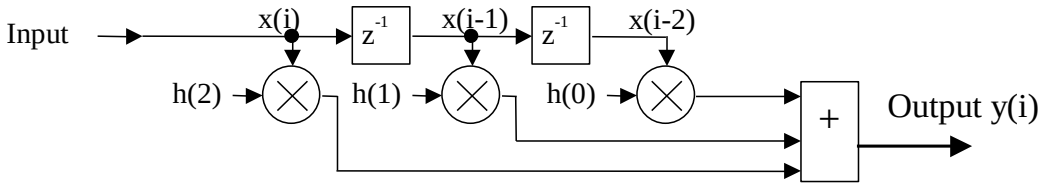

Fig. 1. Block diagram of FIR filter

Block diagram of a FIR filter implemented in FPGAs is given in Fig. 1. Multiplier (*n×n-bit*) occupies less than n^2 4-input LUTs, furthermore for constant coefficient multipliers the number of occupied resources can be significantly reduced [6]. For example, Xilinx XC3S1500 [3] contains roughly 30 000 LUTs, thus more than 100 16×16bit multipliers clocked by roughly 100MHz can be implemented in this FPGA chip. Besides FPGAs incorporate dedicated multipliers, e.g. XC3S1500 contains 32 18×18-bit dedicated multipliers. Summing up, for selected operations, computation power of FPGAs is 10÷1000 (or even more) times larger than for DSPs.

Hereby only a simply example of the FIR filter is given, nevertheless the same hardware architecture can be employed for wavelet transforms and correlation calculations. A slight modification is only required for IIR filters. Similarly, efficient FPGA implementation of neural networks [7] or FFT [8] are available.

FPGAs and Software Solutions

FPGAs and microprocessors complement each other, i.e. usually the most computationally-intensive algorithms are data-driven algorithms (e.g. filtering, FFT), therefore they can be speed-up by FPGAs. On the others hand, program-driven algorithms,

i.e. complex algorithms processed a limited number of times, cannot be easily implemented in FPGAs. Usually these algorithms are not computationally-intensive and therefore are well-suited for microprocessors. Partitioning an algorithm into software part (executed by microprocessors) and hardware part (executed by dedicated logic incorporated in FPGAs) is often denoted as hardware–software co-design [9].

FPGAs design cycle is relatively difficult and time-consuming. Therefore, a modular design approach is adopted. Modules, denoted as Intellectual Property (IP) cores, which functions are well-defined and tested, are supplied by different vendors. Connecting different modules to form a whole system is a fundamental problem which is tackled by Xilinx Embedded Development Kit (EDK) [10]. This software packet allows to graphically connect different modules and to integrate easily hardware and software part of the system.

In the proposed systems tens of different IP modules are employed. Some of then are supplied with the EDK. Nevertheless most of them are developed by the author of this paper; e.g. ADC interface with IIR filter, FIR filter (wavelet transform) with adjustable decimation rate.

It should be noted that the core of the whole system is the soft-processor MicroBlaze [4] which controls all hardware modules. Consequently, the final diagnostic procedures are designed in C/C++ language, which significantly simplifies the design cycle.

Programmable Unit for Diagnostics (PUD)

The proposed Programmable Unit for Diagnostic (PUD) is a device which core is FPGA XC3S1500 device [3]. The block diagram of the PUD is presented in Fig. 2. and it incorporates the following components:

- Four independent analog / digital modules on separate Printed Circuit Boards (PCB).
- Two independent SDRAM memory banks, 64MB each, employed to store acquired data from analogue / digital modules and other temporal data.
- Flash memory (4MB) to store FPGA configuration, MicroBlaze program and other non-volatile data e.g. filter coefficients, FFT coefficients.
- CPLD (Xilinx XC95144XL device) – module employed to configure FPGA and to control the PUD in power stand-by mode.
- LCD display employed to visualise the state of the device and results for acquired and processed data.

- Keyboard – allows user to control the PUD e.g. to start / stop data acquisition.
- PC computer communication by Ethernet, Parallel or Serial Ports.

Besides, the PUD incorporates some optional devices: hard disk driver (HDD), Compact Flash memory, VGA display, PC keyboard modules.

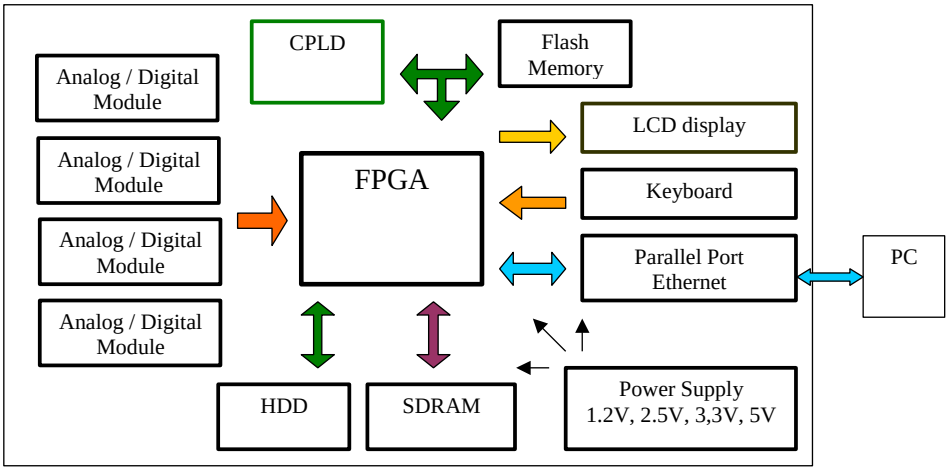

Fig. 2. Block diagram of the PUD

The PUD incorporates four independent analog (/ digital) boards. Each analog board incorporates either 10 MS/s 14-bit ADC or two channels 500kS/s 16-bit ADC or four channels 250kS/s. In total, the PUD can sample either four independent channels at 10MS/s each, 8 independent channels at 500 kS/s each or 16 independent channels at 250 kS/s.

Besides, the PUD incorporates a HDD, which allows for recoding acquired signals. In order to speed-up data transfer, a special HDD file format was designed to allow high speed sequential data writes to the HDD.

Conclusions

This paper presents a hardware solution for diagnostic systems. Several diagnostic procedures are adopted employing the PUD [11, 12, 13]. A detail description of them is outside the scope of this paper. Nevertheless by employing the PUD, input signal sampling frequency together with signal processing speed increases more than 10 times. This allows for significant increase of the overall diagnostic system performance. For example, the PUD adapts the Procedure of Linear Decimation (PLD) [12] directly in hardware which results in real-time signal processing, e.g. in switching-off a rotating machinery already in run-up

procedure in the case when an machine damage is detected [12].

References

[1] LIN T., ZHENGOU Z.: The implementation of 100MHz data acquisition based on FPGA, Proc. System-on-Chip for Real-Time Applications, pp. 287 – 291, **2003**

[2] TOSCHER, S. REINEMANN, T. KASPER, R.: An Adaptive FPGA-Based Mechatronic Control System Supporting Partial Reconfiguration of Controller Functionalities, NASA/ESA Conference on Adaptive Hardware and Systems, pp. 225 – 228, **2006**

[3] XILINX Inc.: Spartan-3 1.2V FPGA Family: Introduction and Ordering Information, DS099 (v1.0), www.xilinx.com, **2003**

[4] XILINX Inc.: *MicroBlaze Processor Reference Guide* Embedded Development Kit EDK 6.3i, Xilinx, **2004**

[5] JAMRO E., WIATR K.: Implementation of convolution operation on general purpose processors, Proceedings of the Euromicro Conf. Warszawa, pp. 410-417, **2001**

[6] JAMRO E.: Parameterised automated generation of convolvers implemented in FPGAs, Ph.D. Thesis, AGH University of Science and Technology, Kraków, **2001**.

[7] JAMRO E., WIATR K.: A Novel Parallel-Serial Architecture for Neural Networks Implemented in FPGAs, Proc. of IEEE Design and Diagnostics of Electronics Circuits and Systems Workshop, Sopron, pp.121-128, **2005**

[8] XILINX Inc.: Fast Fourier Transform v3.2, LogiCore, www.xilinx.com, **2006**

[9] STAUNSTRUP J., WOLF W.: Hardware/software co-design: principles and practice, Kluwer Academic, Boston, **1997**

[10] XILINX Inc.: Embedded System Tools Reference Manual, www.xilinx.com, **2006**

[11] BATKO W., MIKULSKI A.: New method of pit-shaft reinforcement evaluation with the application of pulse test. Problemy Eksploatacji, no. 4, Radom, pp. 173-180, **2006**.

[12] KRZYWORZEKA P., ADAMCZYK J., CIOCH W., JAMRO E.: Monitoring of nonstationary states in rotating machinery. Biblioteka Problemów Eksploatacji, Radom **2006**

[13] BATKO W., CIOCH W., JAMRO E.: Monitoring system for grinding machine of turbine-engine blades. Journal of Polish Cimac Explo – Diesel & Gas Turbine ’07. Gdańsk – Stockholm – Tumba, Poland – Sweden, pp. 37-42, **2007**