Piotr RZESZUT, Ernest JAMRO, Kazimierz WIATR
AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY
30 Adama Mickiewicza Ave, 30-059 Kraków, POLAND

Oscilloscope based on small-size FPGA with VGA display

Abstract

A simple single-chip, FPGA based oscilloscope was designed both to supply a user with a low-budget oscilloscope and to teach operation of such devices. The device implements basic functions of real oscilloscope, providing clear insight in processes of signal acquisition (employing FPGA built-in analog-to-digital converter with aggregated sampling rate equal 1MS/s), processing and displaying acquired signals. Also some effort was made in order to fit the design in limited resources of the selected FPGA device. The project is open source [1].

Keywords: FPGA, oscilloscope.

1. Introduction

A modern oscilloscope tends to use ASICs to realize key features instead of discrete components [2]. Some commercial oscilloscopes, such as Analog Discovery 2 [3], are FPGA-based devices. Some low-cost, non-professional oscilloscopes for educational purposes were previously produced [4], but often used advanced FPGA boards, equipped with external RAM memory and additional components.

The proposed oscilloscope is designed employing general purpose low-budget FPGA board: maXimatorwith Intel FPGA MAX10 10M08DAF256C8GES produced by Kamami.pl [5]. The maXimator board is targeted essentially for students, therefore this oscilloscope may be a side effect of the multi-function board. The maXimator board incorporates3-bit VGA output, 6 protected ADC inputs and some digital inputs arranged in Arduino compatible connectors.

In this paper, we describe the oscilloscope mentioning key components and challenges solved during the design process. The designed oscilloscope does not require any external module, it uses FPGA-embedded analog-to-digital converter (ADC) to acquire input signals and VGA connector to display these signals on an external VGA screen. The oscilloscope is controlled by only three on-board buttons and the oscilloscope mode and parameters are displayed on the VGA screen. A typical digital oscilloscope architecture is presented in Fig. 1 [6].

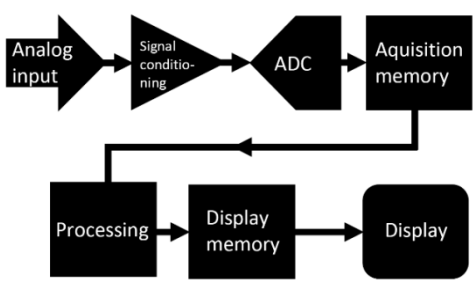

Fig. 1. Typical digital oscilloscope system diagram

Acquired signals are display on the external VGA screen, and the rest of the system, including ADC, is implemented inside FPGA chip. Block diagram of actual system is presented in Fig. 2.

2. Analog to digital converter

Intel FPGA MAX10 family devices incorporate single or dual ADC. 10M08DAF256C8GES device has one ADC converter with an analog multiplexer, allowing to use up to 17 external input channels and the internal temperature sensor [7]. Maximum aggregated sample rate of the converter is 1 MS/s. In this design only one or two channels are used, therefore maximum sample rate is 1 MS/s for a single channel, or 500 kS/s for two channels in chop mode (channel 2 sampling time is delayed by 1 µs in comparison to channel 1, thus only one channel is sampled at a time).

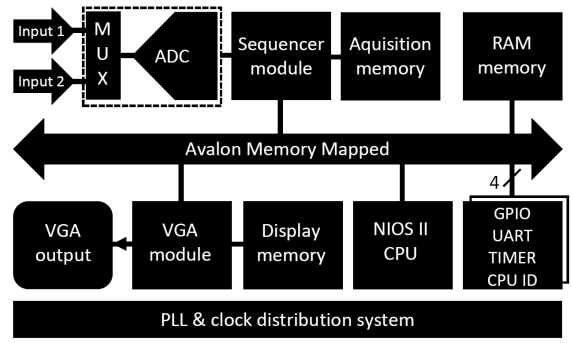

Fig. 2. Block diagram of the designed system

In this design, signal conditioning circuit was not implemented, therefore input voltages range is from 0 V to 2.5 V. Additional functions of the oscilloscope performed on the digital form of the acquired signals are: signals gain (vertical sensitivity) and shifting (offset); arithmetic operations: +, -, *, / performed between the channel 1 and 2.

3. Sequencer and triggering

In an acquisition mode, the ADC is continually sampling acquired signals and the trigger module is monitoring values taken from a triggering channel.

The fundamental task of the trigger module is to generate a trigger signal when the input signal value crosses up or down the trigger threshold. An important feature incorporated inside the trigger module is hysteresis which prevents trigger condition to be detected when noisy signal is applied, as shown in Fig. 3

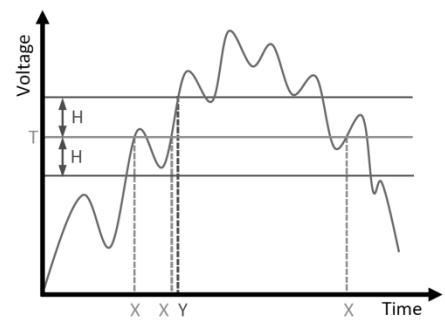

Fig. 3. Triggering on noisy signal. T designates trigger level, H – hysteresis value. X mark all possible trigger situations when using simple rising-edge trigger, while Y mark only trigger situation when using rising-edge trigger with hysteresis

After detecting trigger condition, the sequencer module uses dual port RAM memory and stores 1024 samples. If dual-channel operation is selected even and odd memory addresses are filled with first and second channel data, respectively.

All these features are encapsulated inside sequencer module, which can be interfaced by a microprocessor system (µC) using Avalon Memory Mapped interface. Through the interface, an user can select the sampling rate, trigger configuration, force trigger

and read collected samples. The interface also features interrupt output signal, that may be used by μC to detect end of acquisition without polling. Block diagram of sequencer module is presented in Fig. 4.

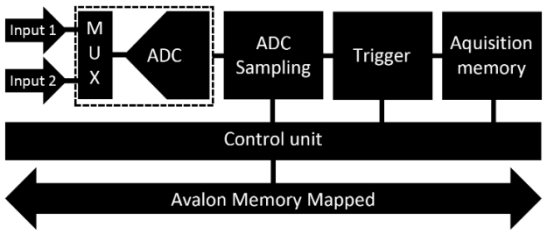

Fig. 4. Block diagram of the sequencer module

4. VGA module

The VGA resolution is 1024×768 pixels, and the maXimator evaluation board is capable of generating 3-bit colour depth(1 bit per red, green or blue colour). A RAM memory embedded in the FPGA device, however, is too small to store the whole VGA frame pixel-by-pixel, therefore special on-the-fly techniques were implemented to display acquired signals.

The VGA module is generating all VGA synchronisation signals and incorporates counters (row and column) to properly address the currently displayed pixel. The VGA module should be fed with the pixel value to be displayed on VGA indicated by the row and column counters. Based on the row and column addresses, the pixel value is generated by the VGA generator modules. Each channel has its own frame generation module. An additional priority multiplexer selects only one pixel value, generated by two or more VGA frame generation modules. Each VGA frame generation module generates pixel request signal, to indicate that it is generating signal (not background) pixel value. The block diagram of the VGA module is presented in Fig. 5. The frame generation module incorporates the following sub-modules: waveform, grid, line and text block. It is important to note that the frame generation module generates the pixel value on-the-fly, there is no need to store the whole frame image in a RAM memory. The VGA column counter addresses the acquired signal RAM to read different time samples. 1024 input signal values (samples) are stored for each channel RAM. The VGA row counter value is compared with the acquired signal value read from the RAM. In the case of the match, the frame generation module generates pixel value corresponding to the channel colour otherwise a background colour is used.

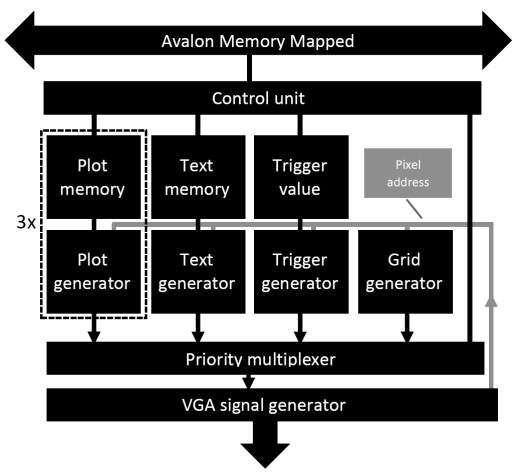

Fig. 5. Block diagram of the VGA module

There is a dedicated frame generation module for text. There are two possible options for drawing text on the screen. The first one assumes implementation character (font) generator ROM in FPGA and text RAM. This approach allow to place many characters using relatively small amount of RAM. The second approach uses a graphic mode to display texts. The microprocessor can set any pattern in the graphic RAM (characters or graphical symbols). As there are very limited resources of RAM available in the FPGA, the graphic mode covers only a fragment of the whole VGA frame. The graphic mode allows to place only limited amount of characters on the screen (as they are stored pixel by pixel), but on the other hand enables very flexible charset and font changes. For the project the second method was selected.

5. Nios II processor and ecosystem

In the design, the Nios II /e (Nios II Economy) soft processor is incorporated. This processor is a 32-bit RISC unit, that comes with JTAG debug support, interrupt controller and Avalon Memory Mapped Interfaces for interfacing other components.

To operate properly processor needs to be connected to instruction and data on-chip RAM memory. Instructions can be uploaded to the program memory by the JTAG during debugging, or during standard FPGA (and also the program memory) initialization.

To complete system design, additional components were added: three input pins for buttons, additional output pins for driving LEDs, interval timer for time measurement and debouncing, UART Serial Port for future development and providing standard output required by Hardware Abstraction Layer and System ID Peripheral to provide unique system ID required by the debugger.

6. Clock system

On maXimator board there is a 10 MHz oscillator connected to the FPGA. Using embedded Phase Locked Loops (PLLs) the following frequencies are generated:

- 10 MHz for the ADC (the converter clock must be fedby *c*0 output of the first PLL on MAX10 device [7]).
- 50 MHz for the Nios II Processor clock and other devices connected to Avalon Memory Mapped bus.
- 65 MHz for the VGA generation, to achieve the pixel clock for 1024×768 resolution with the 60 Hz refresh rate.

7. Software

The program running on Nios II processor, allows user to control behaviour of the oscilloscope using three buttons. The available settings are:

- enable or disable each of two channels
- change channel offset and gain which is performed on the digital signal form
- disable or set arithmetic channel operation
- change arithmetic channel gain and offset
- select trigger value, hysteresis and edge
- select trigger mode (implemented options are: auto, normal, single)
- select time base

The current settings are shown on the screen as texts. As graphic mode was selected to display texts, the processor is copying prepared characters patterns from its memory to the graphic memory. To allow the button operation, GPIO (General Purpose Input / Output)pins are sampled in timer interrupt, which also provides timing for debouncing.

In addition to providing the user interface, the program is processing data acquired by sampling module every time the module generates interrupt. The processor, if required, separates data for first and second channel and calculates arithmetic operations. The processor also controls gain and offset values which are further used in the VGA frame generation module.

8. Software versus hardware implementation

In many point of this paper, hardware and software implementations of some functions were mentioned. For example: text display, button debouncing, calculating arithmetic operations on the acquired channels could be implemented by the processor or by dedicated hardware module.

When making decisions on either hardware or software implementation, the below mentioned factors should be taken under consideration.

Required time-critical operation, such as acquisition of analog signals or generating signals to drive VGA display, strongly suggests implementation in hardware. Taking into consideration VGA frame generation, that is clocked at 65 MHz: a processor would need to read data from memory, set output pins and increment counters at each 65 MHz clock cycle. Assuming that each operation can be performed in a single cycle, that gives at least 195 MHz (in practice much more) clock frequency for a processor that is reserved only for VGA display. In the presented design, the VGA module needs only 65 MHz clock. Although practical comparison was not made between these methods, it seems clear that, due to lower frequency, such design is more power efficient.

Some operations, like character generation, are easier to be designed in software, because software design and compilation is much faster and straightforward than hardware (FPGA) counterpart.

Other operations, such as button debouncing, can be implemented either in software or in hardware. Only when high performance is required, the hardware implementation is preferred over software one.

9. Avalon Memory Mapped and Avalon Streaming interfaces

Intel FPGA provides two main interfaces to communicate with Nios II related IP cores: Avalon Memory Mapped and Avalon Streaming [8].

Avalon Memory Mapped interface is capable of addressing devices connected to the bus. It features both simple and pipelined bidirectional transfer modes. In this project simple transfer mode was used, with separate read and write buses, sharing address lines.

In contrast Avalon Streaming interface is unidirectional point-to-point interface. In the design it is used to transfer data from the ADC control core to sequencer module, and second such interface is used to transfer command (selecting ADC channel) to ADC control core. Both used in the design interfaces are very simple to understand and implement in hardware employing VHDL or Verilog.

10. Results and conclusions

Although generally the maXimator proved to be very good and low cost development board for learning FPGA, some minor issues related to the project were discovered. These issues are related to the ADC: the dedicated analog input pin is hardwired to a potentiometer, and due to the interference with VGA lines or a supply voltage, the noise level of the acquired signals is larger than expected.

The described oscilloscope was fully implemented and tested on maXimator development board. The occupied hardware resources for the oscilloscope are given in Tab. 1

Tab. 1. Occupied FPGA resources

Resource	Used	Available	% Used
Logic elements	3 252	8 064	40
Memory bits	326 400	387 072	84
M9K memory blocks	42	42	100

As the oscilloscope is implemented in FPGA, the functionality can be changed, e.g. number of channels, number of samples per channels. Also the microprocessor software can be modified in order to allow remote control by a PC or the acquired signals to be transferred over the UART.

Authors would like to acknowledge Faculty of Computer Science, Electronics and Telecommunications and Department of Electronics of AGH University of Science and Technology for funding the work.

11. References

[1] https://gitlab.com/piotrva/maximator-scope
[2] http://www.eenewseurope.com/news/novel-chipset-and-architecture-portable-oscilloscopes?news_id=96124
[3] https://reference.digilentinc.com/reference/instrumentation/analog-discovery-2/reference-manual
[4] Basa B.: Realization of digital Oscilloscope with FPGA for education, Procedia - Social and Behavioral Sciences 174 (2015) 814 – 820
[5] http://maximator-fpga.org/
[6] Kamieniecki A.: Współczesny oscyloskop cyfrowy – Budowa i pomiary, Wydawnictwo BTC (2009)
[7] https://www.altera.com/en_US/pdfs/literature/hb/max-10/ug_m10_adc.pdf
[8] https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/manual/mnl_avalon_spec.pdf

Received: 12.10.2017 Paper reviewed Accepted: 01.12.2017

Piotr RZESZUT, MSc

Is a PhD student in the AGH University of Science and Technology in Cracov, received MSc degree from the AGH in 2018. His research interests focus on the spintronic devices such as MRAM and Spin Torque Oscillators.

e-mail: piotrva@agh.edu.pl

Ernest JAMRO, PhD

Received MSc degree in electrical engineering from the AGH University of Science and Technology (AGH-UST), Cracov Poland in 1996; M.Phil. degree from the University of Huddersfield (U.K.) in 1997; PhD and habilitation degree from the AGH-UST in 2001 and 2014 respectively. He is currently an assistant professor in the Department of Electronics AGH-UST. His research interest includes reconfigurable hardware (esp. FPGAs), reconfigurable computing systems, System on Chip, artificial intelligence.

e-mail: jamro@agh.edu.pl

Prof. Kazimierz WIATR, PhD

Received the MSc and PhD degrees in electrical engineering from the AGH University of Science and Technology, Cracov, Poland, in 1980 and 1987, respectively, and the DSc degree in electronics from the University of Technology of Łódź in 1999. Received the professor degree in 2002. His research interests include design and performance of dedicated hardware structures and reconfigurable processors employing FPGAs for acceleration computing. He currently is a director of Academic Computing Centre CYFORNET AGH.

e-mail: wiatr@agh.edu.pl