Digital Oscilloscope Using FPGA

Mentors: Malkangowda S Patil, Nihar Gowda S
Mentees: Yash Malve , Bhaskara Akhila , Kunal Kumar , Mallishawri A , Vishnu Kumar
National Institute of Technology, Karnataka

Objectives

The goal of the project is to implement a 2 channel digital oscilloscope on a FPGA. FPGA's programmable nature make it more versatile than any other computer processor in terms of both speed and efficient utilization of hardware used.

Introduction

In electronics engineering, particularly analog electronics need to display a input or output signal of a circuit is must to analyse the functions and results of the circuit .
Oscilloscopes can be used to capture, process, display and analyze the waveform and bandwidth of electronic signal.The main purposes are to display repetitive or single waveforms on the screen that would otherwise occur too briefly to be perceived by the human eye. As to advantages of Field Programmable Gate Arrays (FPGAs) over the microcontroller is that it allows for parallel processes, you can implement more inputs with an FPGA than with a microcontroller, without running into a bottlenecking issue.

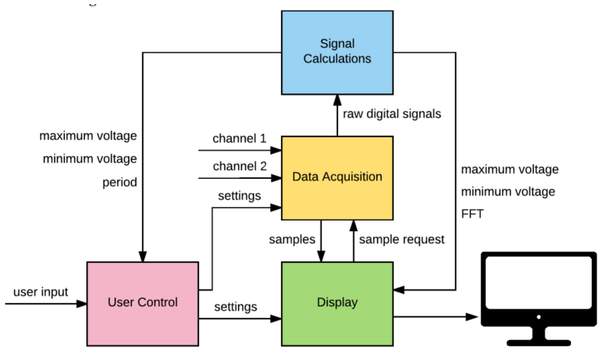
Dependencies and Hardware Specifications

The project is developed using Xilinx Vivado and Basys 3 FPGA BOARD
Dependencies:
Xilinx Vivado
BASYS 3 FPGA Board
PC Specifications:

- RAM:7.7 GiB,Processor: Intel®Core™ i5-8250U CPU @ 1.60GHz
- Graphics: 2GB GeForce 940MX/PCIe/SSE2
- OS: Windows, 64-bit

Block Diagram

Our oscilloscope consists of four major blocks. The Data Acquisition block converts analog signals from two channels into digital signals and stores them in memory. The Signal Calculations block computes the maxima, minima, and average of each signal. The User Control block adjusts the Oscilloscope's settings based on user input, and the Display block draws a waveform.

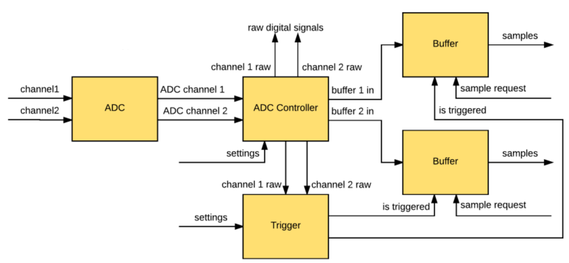


User Control

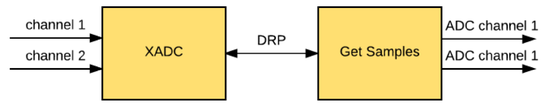
The user block is where we can adjust the setting to zoom the waveform or to display the waveform from channels.

Data Acquisition

The Data Acquisition block receives analog inputs from two channels, settings sent from the User Control block, and sampleRequests sent from Display block. The Data Acquisition block converts the analog input signals into digital signals and stores them in two buffers, one for each channel. Other blocks can access the samples stored in these buffers by sending sampleRequests to the Data Acquisition block



ADC Module



The ADC block instantiates the XADC core, reads the samples stored in the XADC's status registers, and outputs the samples so they can be used by other blocks.

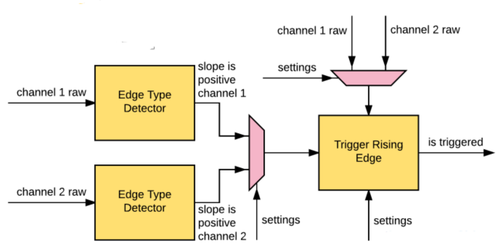
The XADC core is a building block available for all Xilinx 7 Series FPGAs. It includes a dual 12- bit, 1 Mega sample per second (MSPS) ADC and an on-chip analog sensor. The samples obtained by the ADC are stored in the XADC's status registers and they can be accessed through the FPGA's dynamic reconfiguration port (DRP).

ADC Controller

The ADC Controller module accepts samples from the ADC block. On each clock cycle, each input sample is copied directly out on the raw outputs. The non-raw outputs are subsampled and sent to the Buffer modules; the subsampling rate is determined by the settings sent from the User Control block.

Trigger

The trigger system processes the incoming data stream from either channel and asserts the isTriggered signal for one clock cycle when the input data is on a rising edge and crosses the user-defined trigger threshold (the trigger threshold is one of the settings outputted by the User Control block).

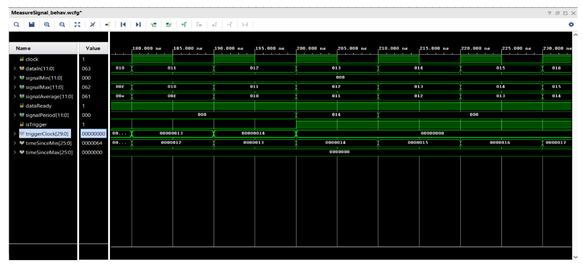


The edge type detector detects the edge is either positive or negative type and if the edge is positive isTriggered signal is set and samples from buffer are sent to display.

Buffer

The Buffer module stores ADC samples for further processing. Its primary input is the ADC Controller module's subsampled output, although its input could also come from a math module or other preprocessing unit.

Simulation Results



Conclusion

All the blocks such as data acquisition , ADC , trigger were written in Verilog Hardware Description language and simualted using Xilinix Vivado Software.During synthesis, the Verilog code is transformed to digital logic gates by minimizing use of resources on the FPGA and during the implementation the bitstream data is loaded onto the device of FPGA programming,

Future Work

To use a VGA monitor to display the waveform in real time and show various attributes such as max , min etc on the screen.

The input signal can further be processed using FFT to display its fourier transform.