

Zynq Audio System Demo

24 bit 48 KHz stereo audio synthesizer and effects processor

Elliot Schei (elliott.schei@zotechgroup.com)

11 September 2018

Platform

This project has been created to run on the Zedboard: <http://zedboard.org/product/zedboard>



Figure 1: ZedBoard used for the demo with speakers, rotary encoder and display connected

The current tool version used is Xilinx Vivado 2017.4, Vivado HLS 2017.4, and Xilinx SDK 2017.4. The design primarily consists of C, C++, VHDL code. Python and Scipy were also used for verification, filter coefficient design and other tasks, but these tools are not required to reproduce the design.

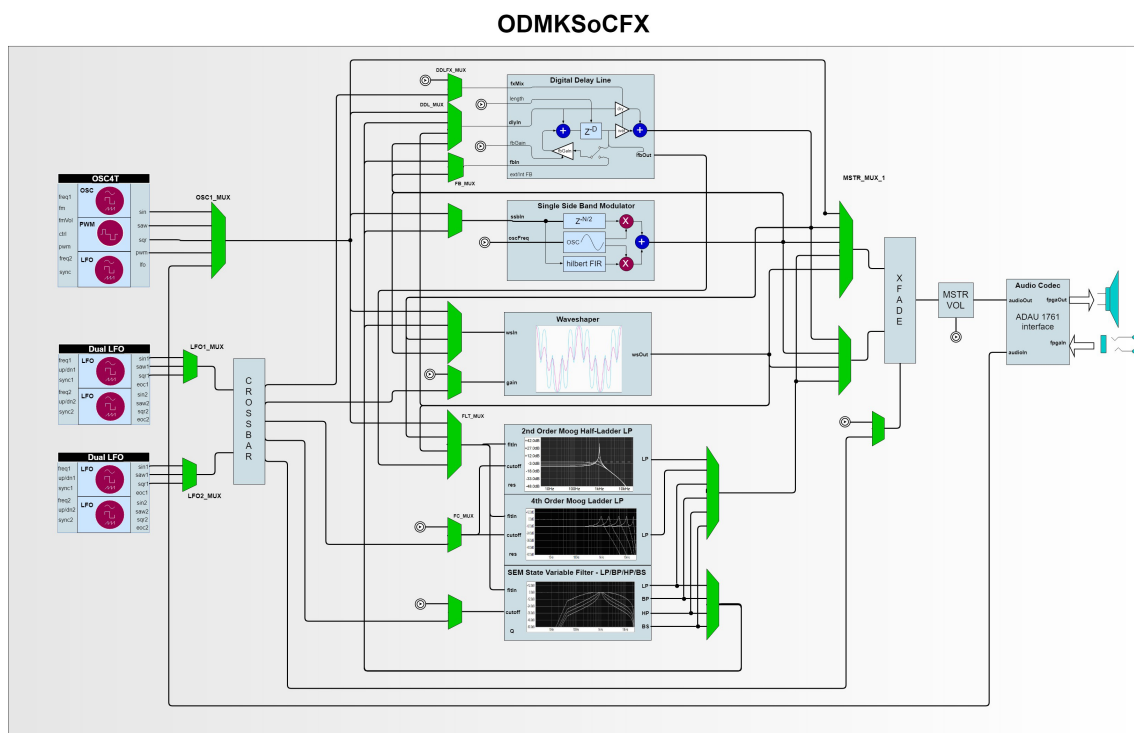
Goal

The purpose of this design is to demonstrate the functionality of ZoTech's IP, and to demonstrate how HLS based designs can be incorporated into a SoC embedded system. All the functions in the system except the low-level interface have been developed with synthesizable C++ using Xilinx's Vivado HLS tool. These functions are then packaged as IP and used together with an ARM processor in the Vivado IP Integrator design flow. Finally, Xilinx's SDK tool is

used for the software portion of the design to control the integration of the peripherals with the functions running in FPGA fabric.

Design Description

The current design is a basic audio synthesizer and effects processor. The main components of the design consist of: a stereo audio interface (1 stereo in, 1 stereo out), a basic audio synthesizer with sine, saw, square and pulse code modulated square waveforms, and a dynamically re-routable effects processor. The effects include single-side-band (SSB) modulation using a Hilbert transform for frequency shifting, a variable stereo digital delay with external feedback routing, a 2nd Order Moog "Half-Ladder" low-pass filter, a 4th order Moog Ladder Filter, a 2nd order SEM State Variable Filter, and a Waveshaper. These effects can be combined in many different configurations in real-time using simple switches and buttons. Rotary encoders are used to control effects parameters, oscillator frequency, volume, etc. The target parameters of the rotary encoders can also be changed in real-time. There is a 96x64 pixel OLED RGB display for visual feedback, and also a USB-UART for additional communication and feedback to and from a computer. The high-level block diagram of the Zynq PL side (FPGA processing) is shown below:



24 bit 48 KHz Stereo Audio Synthesizer and Effects Processor - FPGA (Zynq - PL) side

Implementation

The audio interface is designed to work with the ADAU 1761 audio codec that exists on the Zedboard. Audio input and output samples are clocked using a 48 KHz sample clock. Audio samples are then transferred to an internal 100 MHz FPGA clock while a 'new sample' signal is used to synchronize the design elements. The design in the FPGA runs at 100 MHz, and takes advantage of the fact that there is many cycles available (2083) per audio sample to efficiently use FPGA resources.

The design demonstrates how ZoTech's DDS IP cores can be used for several different functions. The DDS cores are used as audio oscillators, low-frequency control signals, and also for single-side-band modulation. The DDS core is implemented as a templated synthesizable C++ design that is efficiently compiled using Vivado HLS. Although the current demo uses a 100 MHz clock on a Zynq 7020 device, the DDS IP are able to run at greater than 500 MHz in faster FPGA (Kintex Ultrascale FPGA).

Both Fir and IIR Filters are also used in this demo. A frequency shifting function has been implemented using Single Side band Modulation and Hilbert Filtering. The Hilbert Filter is a 369 tap Multiply-Accumulate FIR filter. This MAC FIR can easily compute a new result within a single audio sample using only 1 DSP48 per stereo channel. The remaining filters are based on Virtual Analog filtering using Topology Preserving Transforms. There are two different Moog inspired Low-Pass filters, and a SEM inspired State Variable Filter that can do Low-Pass, Band-Pass, High-Pass, and Band-Reject filtering. The coefficients and other parametric values are partially calculated in floating point in the ARM processor, while the main datapath structures of the filter are implemented using fixed point, and are able to run at hundreds of MHz for very low-latency operation.

Simple audio waveform generation is another part of the demo. Sine, Square, and Pulse-width modulated (PWM) square waveforms are selected using an Audio multiplexor. The frequency of the waveforms is controlled using a rotary encoder connected to a Pmod input on the Zedboard. For the pulse-width modulated square waveform, an additional DDS core generated a low-frequency oscillator (LFO) that modulates the pulse-width. Changing the frequency of the LFO changes the timbre of the PWM waveform. The purpose of this basic waveform synthesizer is to show how the templated C++ DDS ip can be easily instantiated with a wide range of output frequencies, and also provide the ability to be controlled in real-time using an embedded processor (ARM).

The design makes use of one of the ARM processors in the zynq 7020 FPGA to interface with peripherals and also to provide real-time controls for the elements of the design. The ARM processor uses multiple AXI interfaces from the PL to the PS, and also multiple interrupts. Along with the on-board switches, LEDs and buttons, a Pmod 96x64 OLED RGB display is used for status messages and graphics, and a rotary encoder is used for Frequency and Volume control. The ARM processor manages a simple user-controlled interface, allowing the user to select from multiple configurations. The ARM also converts data from the rotary encoder into internal FPGA control signals in order to modify oscillator frequencies, master volume, and other settings.

Future enhancements

Further additions to the design currently in progress include a modulation matrix, reverberation, flywheel processing for the rotary encoders, programmable cross-fading, and doppler processing.