

AC'97 Codec Hardware Driver Example

- [Programmable Logic :: VHDL](#)
 - [LM4550 ac'97 Audio Codec](#)
 - [I/O](#)
 - [VHDL Hierarchy](#)
 - [Xilinx isim Waveform](#)
 - [Development](#)
- [Additional Information](#)
 - [VHD Files](#)
- [Feedback for Our Sponsor](#)

Programmable Logic :: VHDL

LM4550 ac'97 Audio Codec

This is an example hardware driver used to interface a National Semiconductor [LM4550 ac'97 audio codec](#) with an FPGA running at 100 MHz. The design can be scaled to other clock speeds by either scaling the internal counters, or instantiating an onboard PLL to attain a 100 MHz clock. A Spartan 6 FPGA was used to develop the ac'97 controller, but any FPGA can be used as long as the signal timing is taken into consideration with respect to the main system clock. The data sheet for the [LM4550 ac'97 codec](#) can be found below:

[LM4550.pdf](#)

I/O

The inputs to the controller include the main FPGA oscillator, an active low reset, a serial data in line, a 12.288 MHz bit clock from the ac'97 chip, a 3 bit source selector (slide switches) and a 5 bit volume control (slide switches). The controller's outputs include a sync signal, serial data output, and an ac'97 active low reset signal for initializing the ac'97. There are two internal signals to sync the main ac'97 controller with the command state machine. One of these signals pulses every 20us and the other is a signal used for error checking during the tag phase. Consult the LM4550 data sheet for details on the serial frame input/output.

VHDL Hierarchy

The controller has two main parts, the ac97 for generating signals, and converting 18 bit parallel data to serial data to interface the ac97 chip with the FPGA, and the command controller state machine used to configure the registers in the ac97 in a round robin fashion. The FSM can be changed to include user bus signal inputs, routed to each state corresponding to different register values, for on the fly configuration of the codec. The two parts of the ac97 hardware driver are synced with a one cycle, pulsed ready signal. The top of the hierarchy simply routes the parallel outputs of the controller to back into the parallel inputs of the controller on clock edges. This makes the ac97 talk through from input to output. This process in the top level file can be replaced by port mapping user components for various signal processing tasks. The VHDL system hierarchy can be seen in the RTL diagram in Figure 1 below.

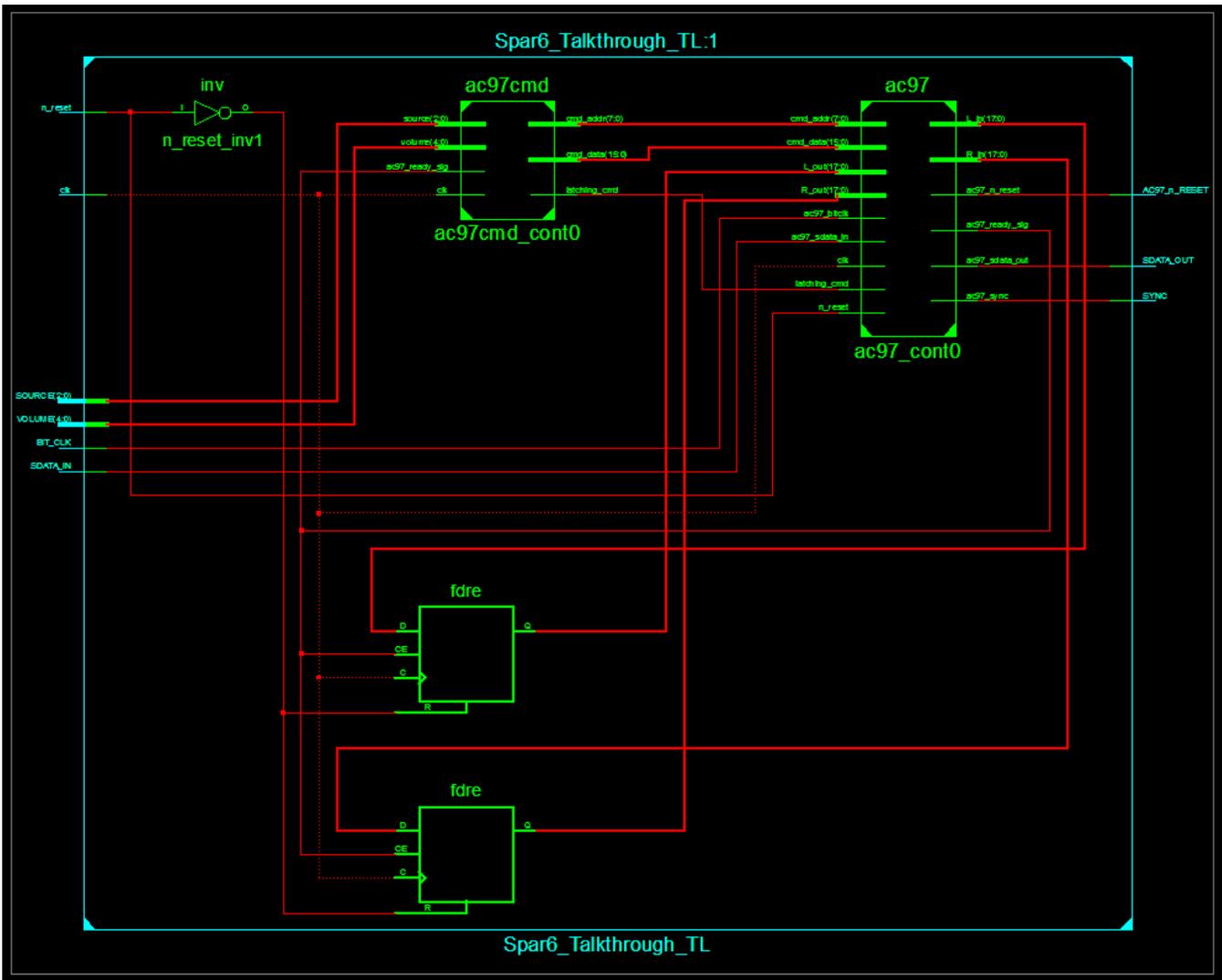


Figure 1 – RTL Block Diagram for AC'97 Talkthrough Circuit

Xilinx isim Waveform

Seen below, the red arrows show the register address and command data on the serial output stream to the ac97. The left and right channel data words can be seen in the serial output following the register and command data. The two bottom signals, ready and latching_cmd, are used to synchronize the two components in the bottom level of the hierarchy. The ready signal should also be used as a sample-ready signal when running the 18 bit parallel data bus to user components.

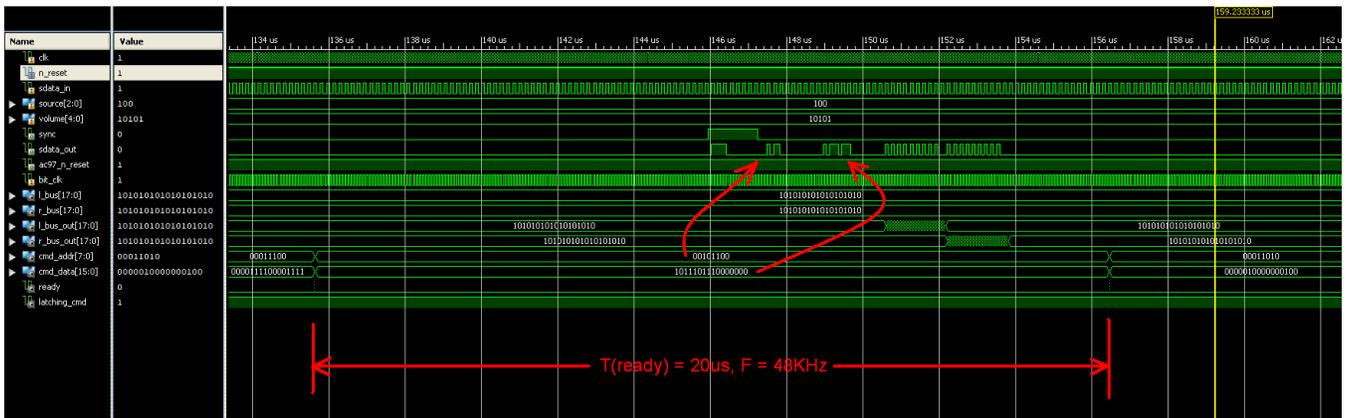


Figure 2 – AC'97 ISim Test Output

Development

This design has been varified in hardware with a [Xilinx Spartan-6 Atlys](#), by [Digilent development board](#).

Additional Information

Any questions regarding this eeWiki article can be directed to the author:

By Phone:

1-800-338-4105 x2969

By Email:

VHD Files

The attached .vhd files can be pasted directly into a new or existing project. The signals will require assignment via a .ucf file or through the use of a Plan Ahead style pin assignment tool.

[ac97_VHD_Files.zip](#)

Feedback for Our Sponsor

Please take a few seconds to help us justify the continued development and expansion of the eewiki.

Click on one of our [Digi-Key](#) links on your way to search for or purchase electronic components.

Is the eewiki helpful? Comments, feedback, and questions can be sent to eewiki@digkey.com.