Lab 7: Interfacing FPGA Spartan-6 with AC'97 Codec

EE-459/500 HDL Based Digital Design with Programmable Logic

Electrical Engineering Department, University at Buffalo

Last update: Cristinel Ababei, 2012

1. Objective

The objective of this lab is to demonstrate the use of the National Semiconductor LM4550 AC'97 audio codec (IC3), which is available on the Atlys board. We'll code in VHDL a driver and implement it on the FPGA to communicate with and control the codec. The driver can select the input into the codec (e.g., microphone, line-in) and set the volume – via the slide switches of the Atlys board.

2. Introduction

AC'97 (Audio Codec '97; also MC'97 for Modem Codec '97) is an audio codec standard developed by Intel Architecture Labs in 1997. The standard is used in motherboards, modems, and sound cards.
Read more about AC'97 here: http://www-inst.eecs.berkeley.edu/~cs150/Documents/ac97_r23.pdf

The Atlys board includes a National Semiconductor LM4550 AC'97 audio codec (IC3) with four 1/8" audio jacks for line-out (J5), headphone-out (J7), line-in (J4), and microphone-in (J6). Audio data at up to 18 bits and 48KHz sampling is supported, and the audio in (record) and audio out (playback) sampling rates can be different. The microphone jack is mono, all other jacks are stereo. The headphone jack is driven by the audio codec's internal 50mW amplifier. LM4550 basically serves as an interface between the analog world of traditional audio components (e.g., headphones and microphones) and the digital world of the FPGA.
Read more about LM4550 here: http://www.ti.com/lit/ds/symlink/lm4550.pdf

3. VHDL driver

This is an example hardware driver used to interface the AC97 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. The VHDL code and description of this controller is based on the work of Tony Storey and Scott Larson [1].

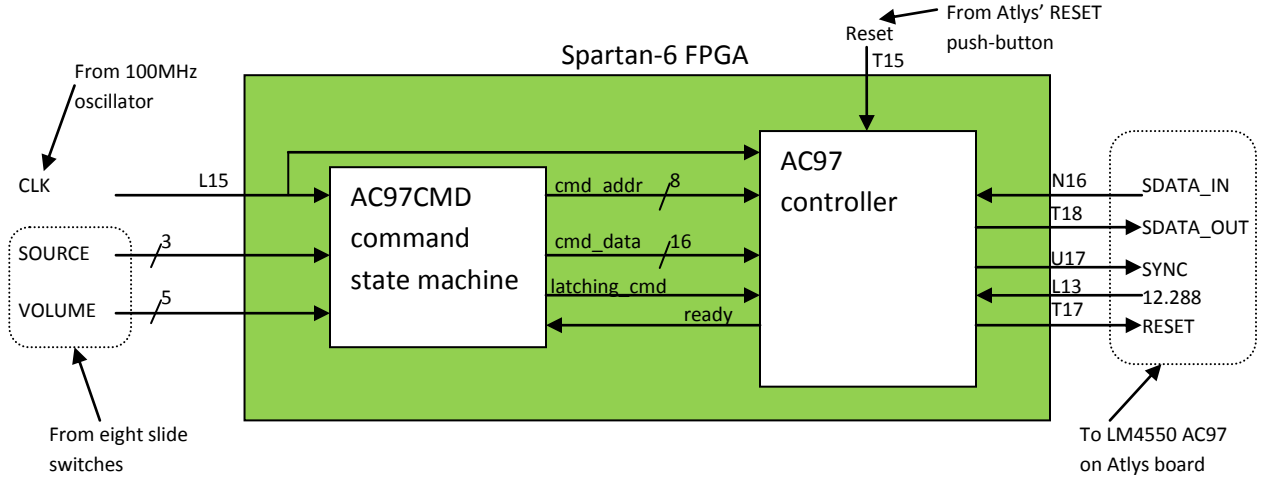

Figure 1: Block diagram of desired circuit

The inputs to the controller "AC97 controller" include the CLK (main FPGA oscillator), an active low reset, a serial data in line, a 12.288 MHz bit clock from the ac97 chip, a 3 bit source selector (slide switches SW7-5) and a 5 bit volume control (slide switches, SW4-0). The controller's outputs include a sync signal, serial data output, and an ac97 active low reset signal for initializing the ac97 (LM4550). There are two internal signals to sync the main ac97 controller with the "command state machine AC97CMD" (a small FSM to setup codec's registers). 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.
The VHDL files can be downloaded on the course website. The downloadable archive contains additional files (datasheets) including the .ucf file that must be utilized to assign FPGA I/O pins correctly. Its content is listed here:

```
# PlanAhead Generated physical constraints
NET "SOURCE[2]" LOC = E4;
NET "SOURCE[1]" LOC = T5;
NET "SOURCE[0]" LOC = R5;
NET "VOLUME[4]" LOC = P12;
NET "VOLUME[3]" LOC = P15;
NET "VOLUME[2]" LOC = C14;
NET "VOLUME[1]" LOC = D14;
NET "clk" LOC = L15;
NET "BIT_CLK" LOC = L13;
NET "SDATA_IN" LOC = T18;
NET "SDATA_OUT" LOC = N16;
NET "SYNC" LOC = U17;
NET "AC97_n_RESET" LOC = T17;
NET "n_reset" LOC = T15;
NET "VOLUME[0]" LOC = A10;
```

4. Synthesis and FPGA programming

Use ISE WebPack to synthesize the entire design and then program the FPGA. Test the whole system using a microphone and the audio signal from your favorite YouTube music video connected to the MIC and LINE IN of the Atlys board. Use the slide switches to select between the two inputs and vary the volume.

5. Lab assignment

Read the datasheets of AC97 and of LM4550 to get an understanding of the serial communication. Read the provided VHDL code and understand how it works – try to sketch the state graphs of the two FSM's from Fig.1 above.

Propose and implement a new VHDL design; you should reuse some or the entire VHDL code to do something different. The given VHDL design hierarchy simply routes the parallel outputs of the controller back to its parallel inputs. 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 for example. An excellent example is the following voice-recorder design:
http://web.mit.edu/6.111/www/f2008/handouts/labs/lab4.html
The top-level plan is pretty simple – when recording, store the stream of incoming samples in a memory (inside FPGA or on Atlys' memory?), when playing back feed the stored data stream back to the codec.

6. Credits and references

[1] Tony Storey and Scott Larson, AC'97 Codec Hardware Driver Example.
http://eewiki.net/display/LOGIC/AC%2797+Codec+Hardware+Driver+Example
[2] http://www.javiervalcarce.eu/wiki/VHDL_Macro:_DC97#cite_note-0
[3] http://www-mtl.mit.edu/Courses/6.111/labkit/audio.shtml
[4] http://web.mit.edu/6.111/www/f2008/handouts/labs/lab4.html