Audio Project

School of Computer Science and Electrical Engineering
University of Queensland, Brisbane, Australia.
http://www.csee.uq.edu.au/

THE UNIVERSITY OF QUEENSLAND

Last Modified: 23 February 2001

Contents

1.0 About this design

The audio project is rather simple as the ADC/DAC provided is easy to use and produces an easy to read output. Due to limitations in the clock speed multiplier selections, the audio project samples at a frequency of 48.8 kHz instead of the normal 48kHz that would be used. This project allows recording and playback of just over 10 seconds of sound and has options to play back the sound at doubled speed, reversed etc.

2.0 Files needed for this design

List of Files

- remap.vhd
- audiotop.vhd
- audio.vhd
- player.vhd
- recorder.vhd
- sram512k32bit50mhz-sv05
- audiopins.ucf

File Descriptions

remap.vhd

Renames ports to fit the ucf file. Top level file of the project

audiotop.vhd

Connects the audio entity to the ram controller.

audio.vhd

Creates clocks for the audio codec and connects the recorder and player devices together.

player.vhd

Reads audio data from RAM and coverts it into serial data for the audio codec. Requires the clocks SCLK and LRCK as input to produce data at the right time.

recorder.vhd

Converts serial data from the audio codec into data which is stored in RAM. Requires the clocks SCLK and LRCK as input to read data from the serial input at the right time.

sram512k32bit50mhz-sv05

RAM controller to read and write RAM.

audiopins.ucf

Constraints file for the audio design

3.0 Description of the design

Clocks

There are three clocks generated by the FPGA to control the AK4520A which are all generated by the 50MHz input clock. Their frequencies are:

MCLK = 12.5MHz
SCLK = 3.125MHz
LRCK = 48.828kHz

SCLK is also used to clock the audio playback and recording devices, and both of these read the value of LRCK to determine when to start each sample.

Serial Data

Serial data in and out is limited to 16 bits. The chip actually outputs 20 bits of data, but only the most significant 16 are stored during recording, and the last 4 bits in playback are always zero. This is required as the RAM is only 16 bits wide. Data is clocked into the recorder on the positive edge of SCLK, and data is changed during playback on the negative edge of SCLK. This means the data will always be ready at the right time. It is likely that some of the most significant bits of RAM are not being used under normal circumstances (as the sound doesn't get that loud). For this reason, a different slice of the data (e.g. bits 17 to 2) could be taken giving better precision, but a smaller range.

Interaction with RAM

The audio project uses the entirety of RAM for both reading and writing. The left bank is used for the left channel of data, the right bank is used for the right channel. Both banks are operated on simultaneously for both writes and reads, and the data bus is treated as being 32 bits wide. The RAM controller affects how the data is split up. At 48.828 kHz, the RAM can only store about 10.74 seconds of data. When this limit is reached, both the recording and playback devices will stop and wait to be told to start again. Also,

RAM operations occur at different times in the cycle, so playback and recording can happen simultaneously.

Controls

The four pushbuttons are mapped to control signals for the audio project and the reset button. The functions of the pushbuttons are:
1: Reset
2: Playback stop
3: Playback start
4: Record start

In the audio.vhd file, there is also some manipulation of the read address and read data of RAM. This allows changes to how the audio plays back without changing the data in RAM and keeping the playback program the same. The following table shows which choices are available for audio playback. All of these are separate from each other and can be used simultaneously. For normal operation, leave all inputs at 1.

Signal name	Switches	Value	Description
Direction	1	0	Playback is in the reversed direction
		1	Playback is in the forward direction
Speed	2,3	0,0	First half of buffer is played at ½ normal speed
		0,1	Second half of buffer is played at ½ normal speed
		1,0	Entire buffer is played twice at double speed
		1,1	Entire buffer is played once at normal speed
Precision	4	0	Playback uses 8-bit precision
		1	Playback uses 16-bit precision
Volume	5,6	0,0	Volume is quadrupled with capping for overflow
		0,1	Volume is doubled with capping for overflow
		1,0	Volume is halved
		1,1	Normal playback volume

Extensions

The following could be done to increase the amount of data that can be stored at one time:

Move to 8-bit precision. This would double the amount of data that can be stored. If this were done, it would be good to find exactly which bits are needed, as it seems the most significant 8 bits are probably not a good choice for most applications.

Only use mono sound rather than stereo. This will also leave twice as much room for data.

Change sampling speed. By only writing every second sample, but playing every sample twice, the sampling rate would be decreased and the storage space doubled.

It is also very easy to operate on data while it is not being used by the playback and recording devices. In this way, real time audio manipulation can be performed with simple algorithms.