Wiki »

EXERCISE: ST BUS

Exercise Goals

This exercise will through implementation of af FIR filter and an ST-bus to IIS interface converter introduce you to:

- Configurations as a means of testing in the ModelSim IDE
- The ST-bus protocol

Prerequisites

- Quartus and ModelSim software must be installed and working.
- The FIR Compiler II MegaCore Function User Guide
- SOPC Builder User Guide Chapter 12
- Introduction to the Qsys Tool
- Avalon Interface Specification chapter 6
- Skeleton code checked out from the repo (de1soc_audio)

Introduction

The Avalon ST-bus is used in SoC systems for low-latency data transfer. It is particularly well-suited for continuous (streamed) data.

In this exercise we'll create a bridge between the digital audio interface (IIS) and the ST-bus. Se the I2S-ST Functional Test Bench exercise for detail on how the interfaces are wired together.

The IIS bus, it's a 1-bit serial bus with two channels, left and right:

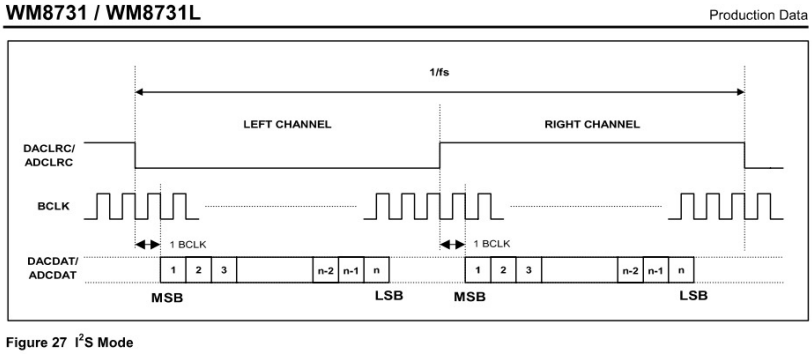
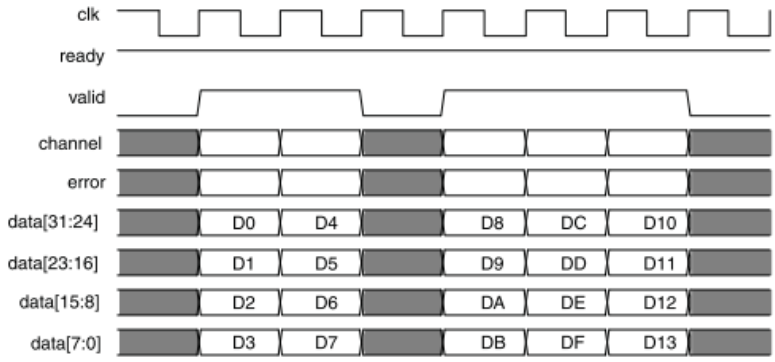


Figure 27 I²S Mode

Similar, but different is the ST-bus:



This example of a 32-bit/beat interface shows a sequential bus, yet data are transferred as parallel bits. The IIS bus is only 1-bit/beat, but 24-bits per package (compared to the ST-bus)

Exercise Steps

1 Create a Fir Filter

Locate the fir sub-folder in the de1soc_audio directory. In here you will find template code for the Fir filter interface, and a file containing an IIS <-> ST converter (iis2st.vhd) plus a test bench.

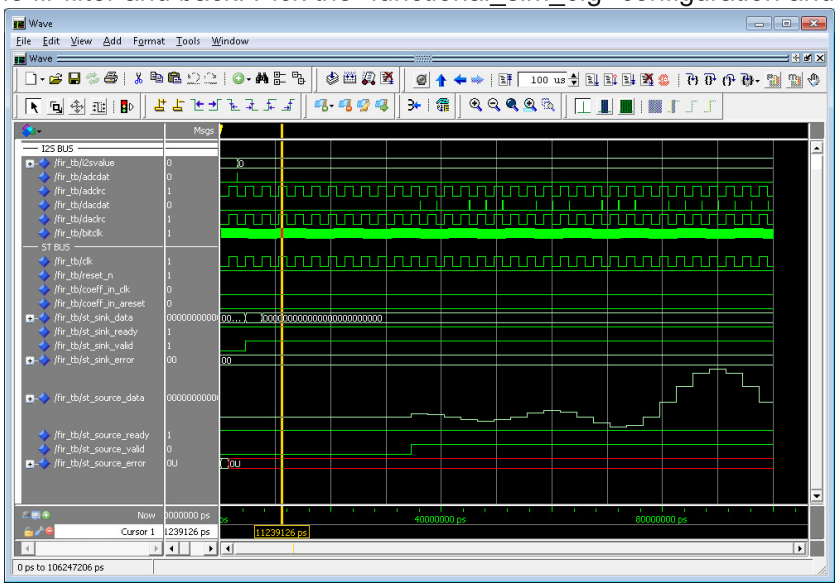
You can replace the content of the filter with a filter of your own or you can insert a filter generated by tQuartus' built-in Fir filter generator. Follow this Tutorial to create a FIR filter with Quartus II FIR compiler II if you want to do the latter.

2 Insert the FIR filter into iis2st_tb

A) Create a ModelSim project that includes the following (presuming you have called your fir filter for "fir_filter"):

- fir_filter.vhd
- iis2st.vhd
- iis2st_tb.vhd
- Plus all files generated by the FIR compiler, if you have used it

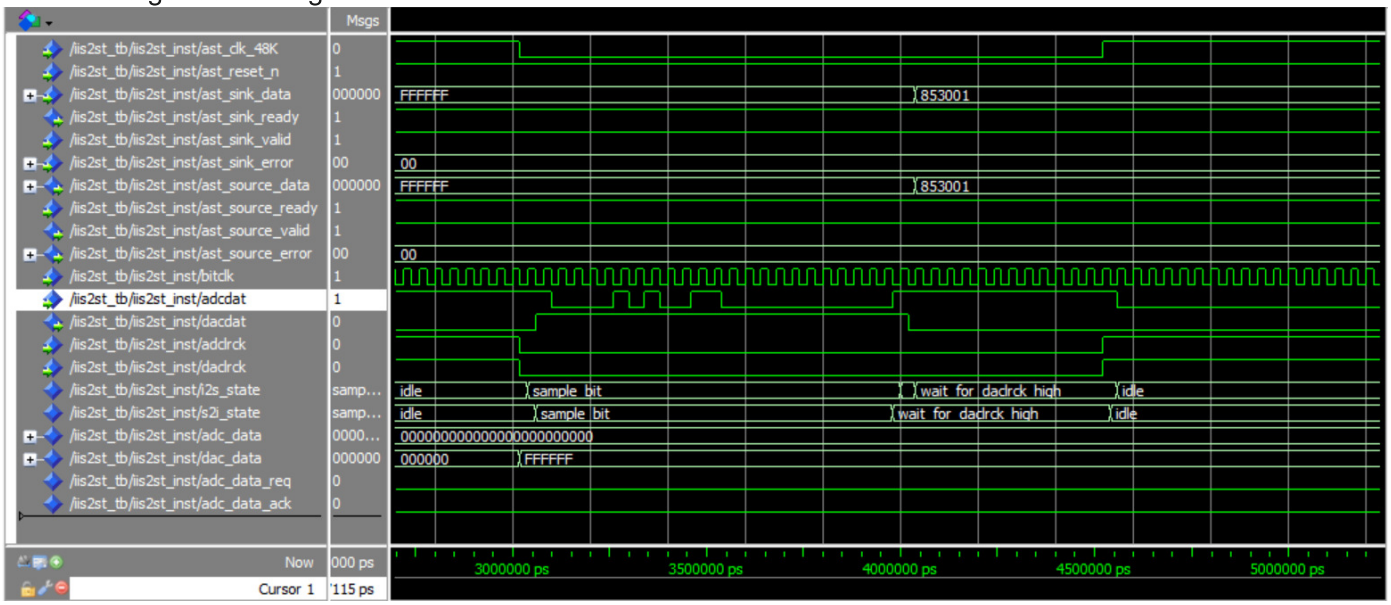
B) Simulate! The test bench generates an impulse signal on the IIS interface and passes it through the IIS2ST converter through the fir filter and back. Pick the "functional_sim_cfg" configuration and you should get something like this:



Yes, it's a Sinc function! You are using the functional model from iis2st.vhd and now have a basis for evaluating your next work.

C) If you are tough, then implement the RTL version of the ST to IIS converter in the iis2st architecture. Inspect the code of the IIS to ST converter and do the opposite. I you are weak ;-) then use the iis2st_sol.vhd file Verify by simulating, but this time use the "rtl_sim_cfg" configuration in ModelSim.

You should get something like this:



D) Integrate the iis2st and fir_filter into de1soc_audio_no_hps_top. The two modules should be wired together as seen in the test bench.

The IIS interface must be mapped like this:

```
bitclk              => AUD_BCLK,
adcdat             => AUD_ADCDAT,
dacdat             => AUD_DACDAT,
adclrck            => AUD_ADCLRCK,
daclrck            => AUD_DACLRCK
```

Remember to remove the ADC/DAC loopback at the bottom of de1soc_audio_no_hps_top

iis2st_wfm.png (159 KB) Peter Høgh Mikkelsen, 2017-02-17 14:07