EXERCISE: SLIP BUFFER

Exercise Goals

This exercise will through implementation of a slip buffer introduce you to:

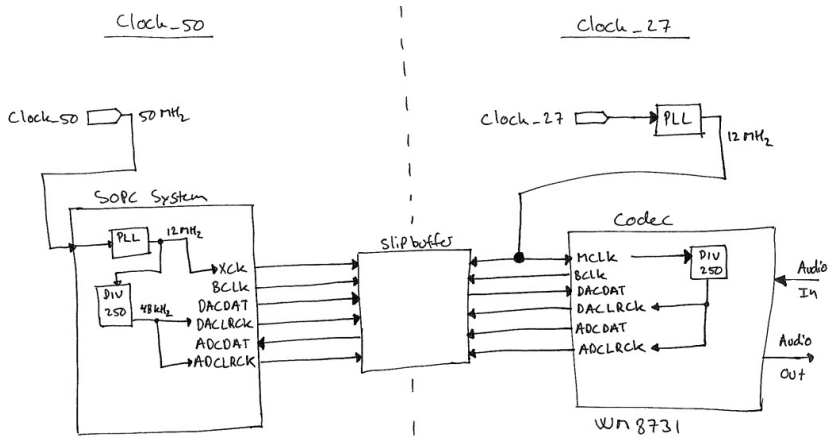
- Clock domain crossing issues
- More construction and test

Prerequisites

- Quartus II and ModelSim software must be installed and working.
- WM8731
- Skeleton code checked out from repository (exercise_slip_buffer)

Introduction

A slip buffer (also know as elastic buffer) provides a means to interface digital data streams between clock domains.



The above example shows two systems each working in their individual clock domain. The IIS signals transferring the digital audio between codec and SOPC system runs at 12 MHz at both sides, but sourced from different clock generators (50MHz and 27MHz oscillators). The slight difference in the two 12MHz clocks (Say +/- 10 ppm) will result in data building up on one side and data being sparse on the other. This will happen at a rate equal to the difference in frequency (At 48KHz sampling rate it will be 20 ppm of 48KHz ~ 0,5Hz). The result will be a sampling of erronous data. For audio it will result in a "pop" sound at the rate just mentioned. This effect is also know as a slip. I happens when one clock advances one cycle in common to the other clock.

To accomodate for this a slip buffer is used. Its function is very simple:

On the side where the RECIEVING CLOCK is too FAST:

- When lacking a data word, repeat the last value!

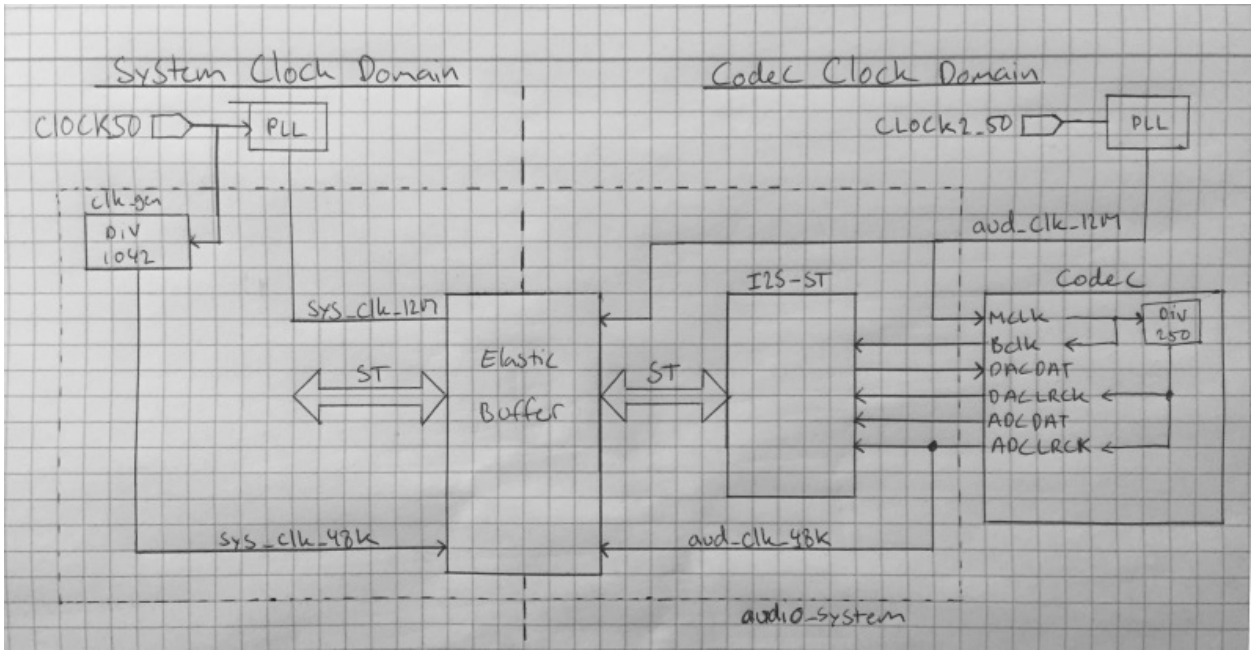
On the side where the RECIEVING CLOCK is too SLOW:

- When having an excessive a data word, throw it away!
  To detect that one clock has advanced its phase by 360 degrees in common to the other (it has progressed by one clock cycle), and that you have to take one of the two actions just mentioned, simply:

- Let Clock A sample Clock B, if Clock B's value changes, say from low to high, you know it has progressed one cycle. Do remember double flopping!

EXERCISE

BLOCK DIAGRAM for Exercise!



A) Open the Slip Buffer project in Quartus (de1soc_slip_buffer)

B) Build the project and download to the board. You should be able to loop-back audio on the DE board from Audio-In to Audio-Out. If SW is off, line-out is silent, otherwise audio should be looped back. In case audio is looped-back, you should be able to hear clicks and pops in the sound, it is probably most audible with the audio input silenced.

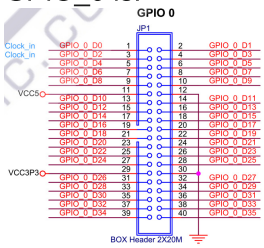
C) Open the project and investigate the source files.
In the project you will find a subfolder, audio_system, this contains all the files you will have to work on. It consists of the following source files:

- audio_system.vhd - Contains 48KHz clock generation and instantiation of the slip buffer
- elastic_buffer.vhd - Instantiates two push-synchronizers (commented out, since they are not yet implemented) and switching of the ST-bus signals, so that you can switch the slip buffer on/off.
- elastic_buffer_tb.vhd - Test bench, generates a signal on the "B" sink side, you can see in the waveform window, if your synchronizer manages to send it to "A" and back to "B" source.
- clock_gen.vhd - Simple clock generator. Divides the input clock but the ammount specified in the generic parameter
- iis2st.vhd - I2S to ST interface conversion
  In the main project folder you will find de1soc_audio_no_hps_top.vhd, which instantiates the audio_system and PLLs and wires it to the external interfaces (audio, clocks, switches etc).

The two clocks, aud_clk_48K and sys_clk_48K are wired out to the expansion header GPIO_0 for debugging.

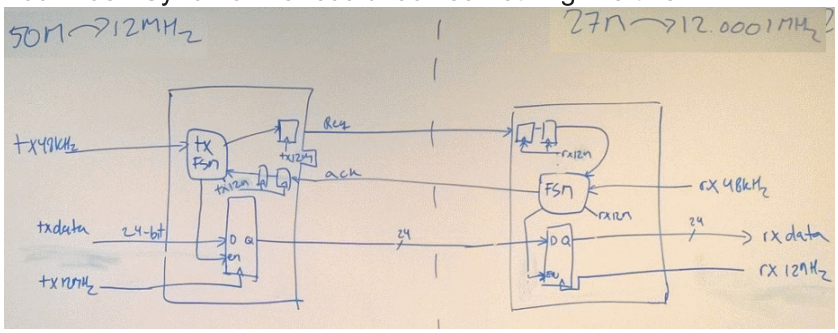
- GPIO_0(0)= aud_clk_48K
- GPIO_0(1)= sys_clk_48K
  You can connect these pins to your Analog Discovery (+GND) to se the clock relationship. Pinout for

GPIO_0 is:



D) Create a push-synchronizer to prevent audio cracks and pops. Simulate and Instantiate two in the elastic_buffer.vhd. Remove the two lines of code, that silences the output. Simulate the elastic buffer with the given test bench and when it works, compile the Quartus design, download and test on the DE board.

Your Push-Synchronizer could look something like this:



gpio_0_pinput.PNG (135 KB) Peter Høgh Mikkelsen, 2017-02-22 11:04

dspc_ex_slip_buffer_de1soc_block2.png (218 KB) Peter Høgh Mikkelsen, 2017-02-22 11:24