

LAB 4



Pmod-I2S2 IP-Core

The provided Pmod-I2S2 IP-Core has:

- Two I2S interfaces, to be connected to the external pins of the FPGA.
- Two AXI4-Stream interfaces, to read the data from the ADC or to send the data to the DAC.
- Two clocks, one for the I2S (**MUST BE 22.591 MHz**) and the other one for the AXI4-Stream (with a frequency of your choice).
- One input active-low reset signal.



Pmod-I2S2 AXI4-Stream format

The AXI4-Stream interface has an additional line called TLAST, which is used to determine the end of a packet.

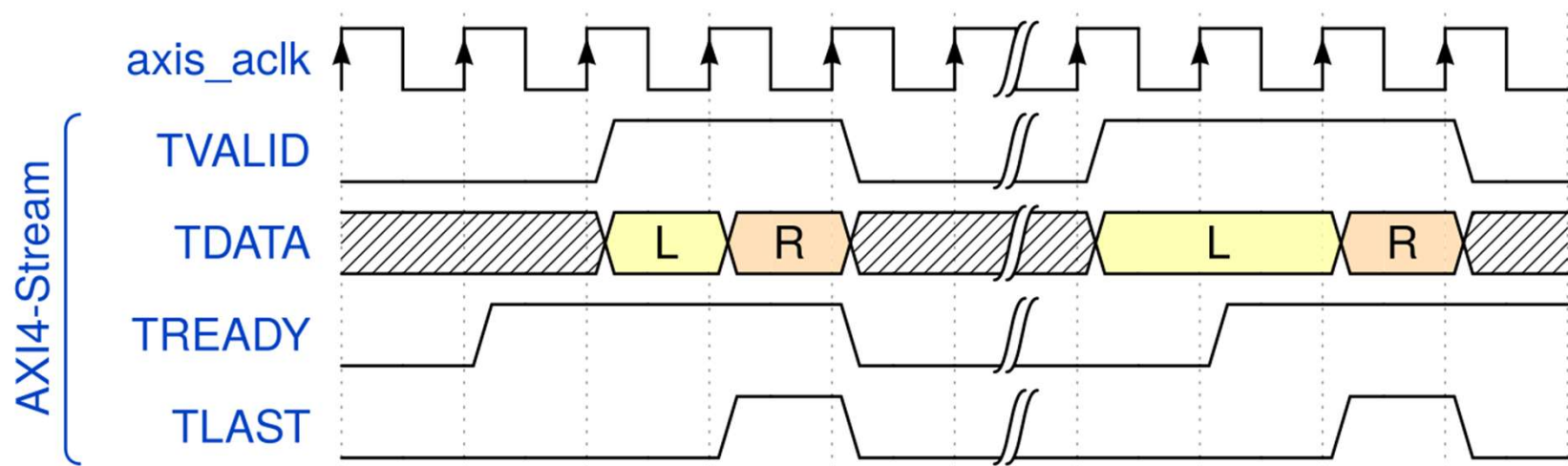
Each packet is composed by two 24-bits words: the first one is the audio data of the left channel, the second one the audio data of the right one.

TLAST is asserted on the second word; in other words:

- TLAST = 0: left channel
- TLAST = 1: right channel

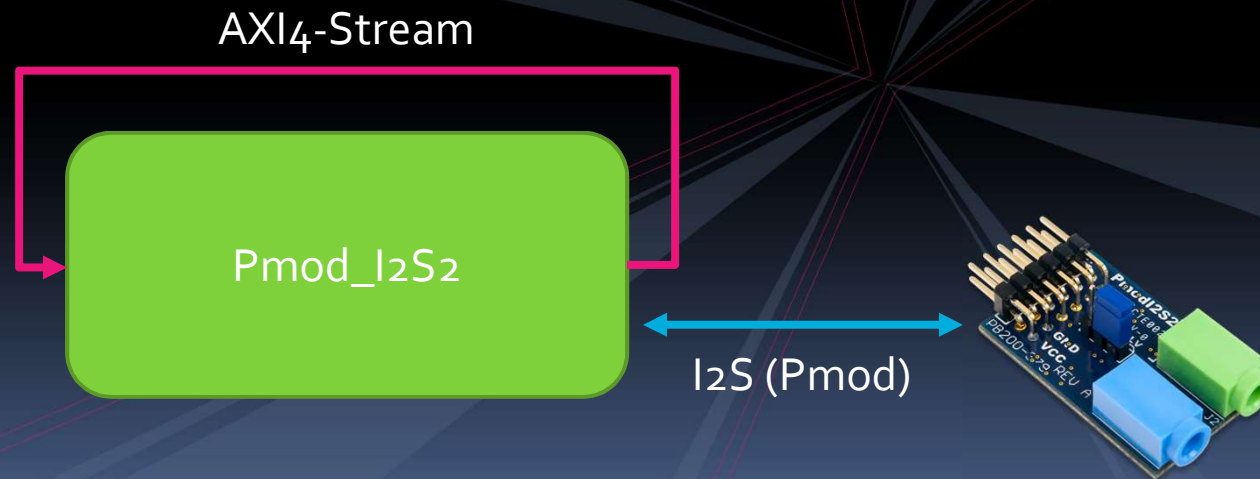
Pmod-I2S2 AXI4-Stream format

In this example, two «packets» have been transferred, first the left channel and then the right one for each one.



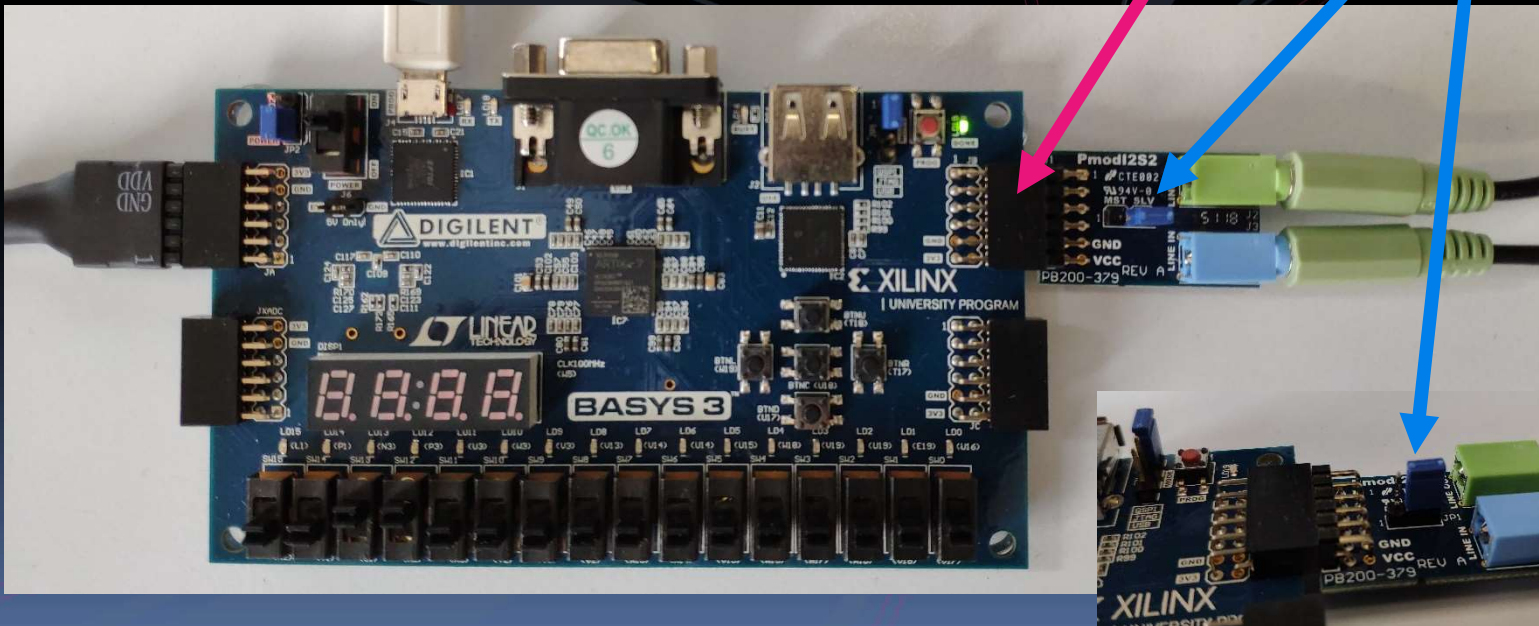
First steps

Like with the RS232 module, the first thing to do is to check whether the IP-Core and modules works by connecting the output stream to the input stream in a «loopback» configuration.



First steps: connection

The provided constraints are for the JB connector (top right of the board). Also make sure that the jumper on the PmodI2S2 module is on the SLV position (right position).



First steps: checklist (1/3)

- Create a new project and select «Basys 3» as the board of the design
- Create a new «Block Design»
- Instantiate our three IP-Cores: Pmod_I2S2, Clocking Wizard, Utility Vector Logic
- Configure the clocking wizard to generate a 22.591 MHz clock (to be connected to i2s_clk) and a 100 MHz clock (to be connected to every other clock input).

First steps: checklist (2/3)

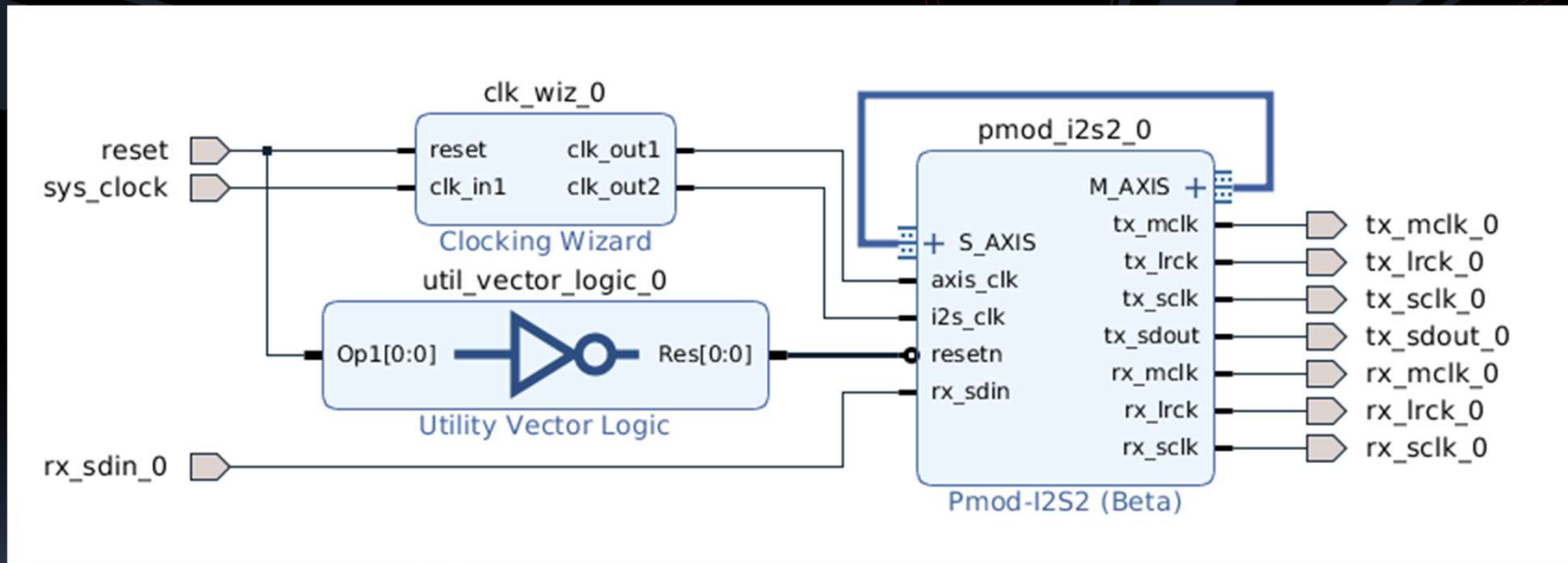
- Connect the Clocking Wizard inputs (clock and reset) to the correct pins (in the «Board» tab on the left).
- Configure the Utility Vector Logic as 1-bit «NOT» and use it to invert the value of the reset.
- Connect the output AXI4-Stream interface of the Pmod-I2S2 module to the input one
- Ctrl-click all the I2S signals (tx_mclk, ...), then right-click on one and «Make External».



First steps: checklist (2/3)

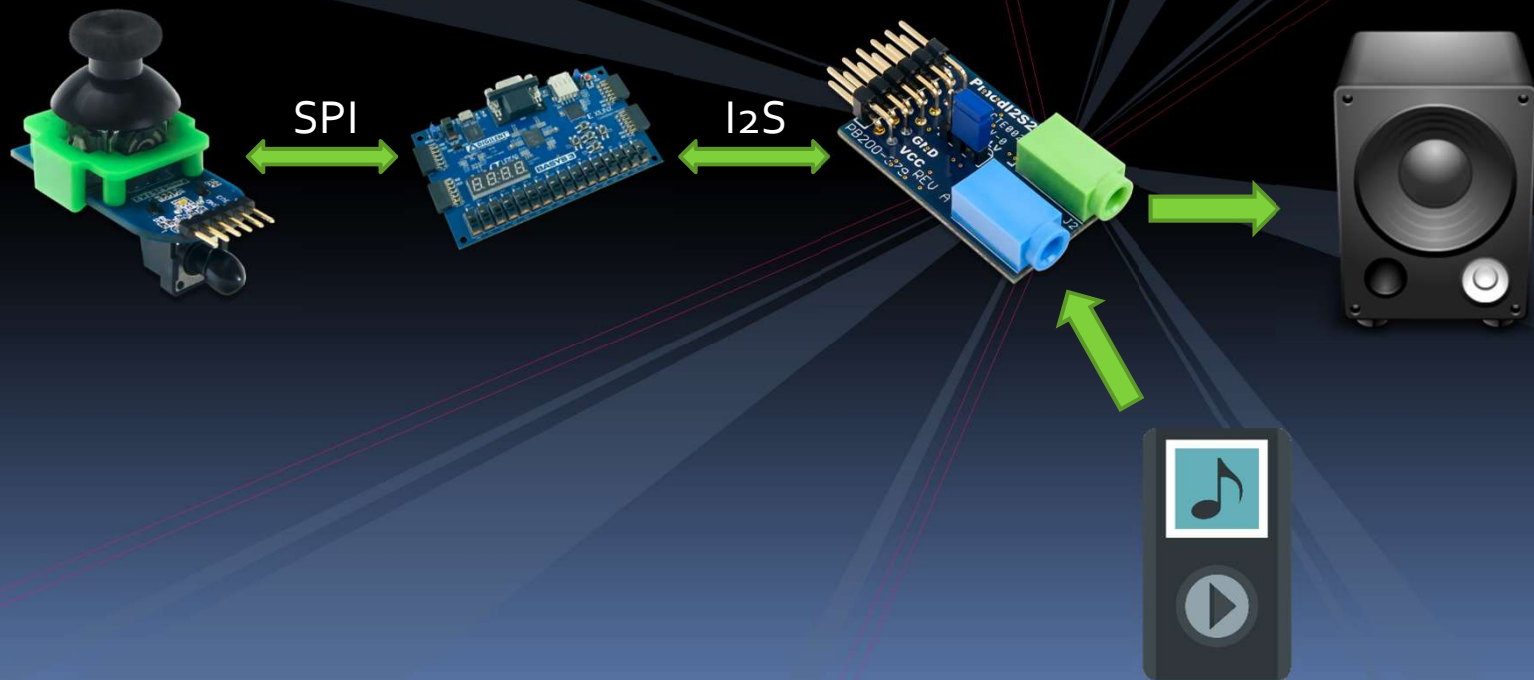
- Right-click on the block design file in the «Sources» tab and select «Create HDL wrapper» («Let Vivado manage...»).
- Add the constraints for the I2S pins.
- Generate the bitstream and check whether the system works as expected.


Expected Block Design



Assignment Requirements


Using the Pmod_I2S2 and Pmod_JSTK2 modules and IP-Cores, build a digital audio console.





Assignment Requirements (1/2)

The output audio should reproduce the input one with some “effects” applied:

- The vertical axis of the Joystick should control the volume of the output audio.
 - The horizontal axis should control the audio balance.
- 

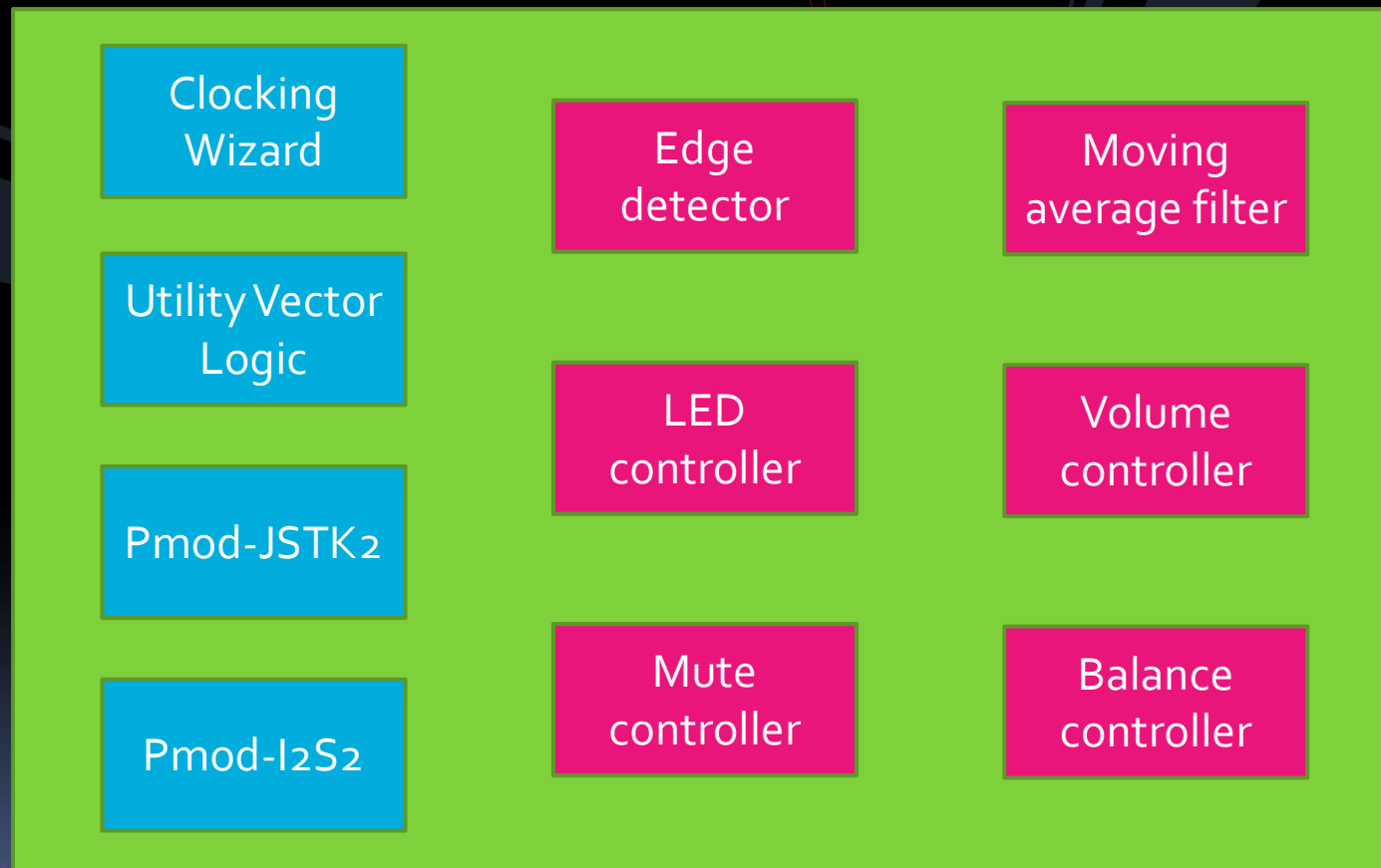
Assignment Requirements (2/2)

- Pushing* the “trigger button” mutes or unmutes the output channel.
- Pushing* the “joystick button” enables or disables a moving average filter (depth=32).
- The LED on the PmodJSTK2 module should show the status:
muted (red), filter active (blue),
no effects (green).

* toggles the status, not just “active when pressed”

Details: project structure

Top (Block Design)



Details: volume control

We perceive “loudness” in a logarithmic way, so the volume control should be exponential. The amplification factor should double every 2^N “joystick units” (with the center in the half of the joystick dynamic, with N as generic). N=6 returns good results.



Use an exponential amplification factor also for the balance control.

- Moving the joystick to the right decreases the left channel volume.
- Moving the joystick to the left decreases the right channel volume.



Details: other modules

- Edge detector: toggles its output when it detects an edge transition at its input
- LED controller: sets the Pmod-JSTK2 LEDs to the correct colors, depending on its inputs ("mute_on" and "filter_on")
- Moving average filter: applies a Moving Average filter (feedback implementation) on its input (depth configurable by generic)
- Mute controller: self-explanatory