Exercise

Implement a FIR filter co-processor in FPGA

Design an FIR filter both VHDL and in a programming environment of your choice (i.e. python) to compare the results from the two implementations (i.e. Frequency analysis). The FIR filter could have an arbitrary number of “taps” and an arbitrary behavior in the frequency domain (i.e. Low-pass, high-pass, band-pass, notch). The data width of the samples will be **8**. The size of the data processed by the FIR.

Block Diagram

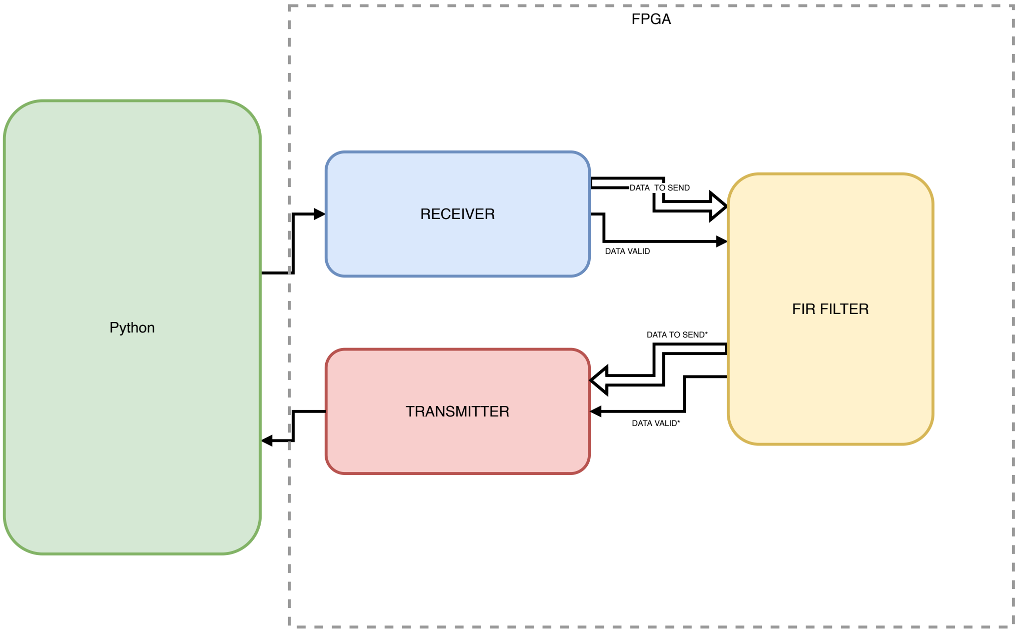

Figure 1: Block diagram

CernBox projects (click on them)

Download from the links below and copy the unzipped folder in your home directory at lxilinx1. (create a ssh tunnel and use the `scp` command with the option `-r` to transfer a whole folder)

1. Arty 100 project
2. Arty 35 project

Starting from the codebase included in the project "uart_complete" delivered in the CernBox (choose the project according with your board kind, A35 or A100, tips: to discover the model of your board, type `make discover`), you are requested to build a VHDL project able to run in the Arty7 Board with the following requirements:

1. The Arty7 board is in charge to apply the FIR filter to the data sent by the lxilinx1 server
2. It is possible to send the input data sequence (few hundred of samples) and get the result back using uart interface
3. The data are processed by the FIR filter in FPGA
4. The data results of the FIR operations have to be compared to the result obtained with the script environment and has to be the same.

Compose a technical report that explains the various developement phases, simulations, results.

Hints

1. Take care of the **data valid** signals coming from the uart receiver and the other similar **data valid** entering in the transmitter: they have to be connected to the FIR block.
2. Please study carefully the beaviour of the VHDL **signed** data type. In order to manage also the negative numbers the *2-complement* representation has to be adopted. To gain confidence with this data type dedicated preliminary simulation is strongly suggested. See the Free Range VHDL Book cap 11.9, page 150
3. Be careful about the FIR output data range: typically a filter has the output larger number of bits than the input. Suggestion: The function *shift_right* performs a "division by 2 at the power of.." useful to reduce the numbner of bits(even the precision, of course)

```
(...)
    Data    <= shift_right(Data, 1);
(...)
```

In the case of using python, the **scipy** library has a method *firwin* that helps to calculate the coefficients related to a specified frequency behavior. Take into account that the FIR filters have as many transient states as the number of taps.

https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.firwin.html

UART tx/rx example session in python

Change the *ttyUSB* number accorinding with the device that belongs to your group.

```
import serial
ser = serial.Serial('/dev/ttyUSB9', baudrate=115200)
for i in range(10):
    ser.write(chr(i))
    d = ser.read()
    print ord(d)
```

Useful links

1. Arty7 schematic https://reference.digilentinc.com/_media/reference/programmable-logic/arty/arty_sch.pdf

for any questions please write to: bergnoli@pd.infn.it , fmarini@pd.infn.it