Assignment -3

- Due Date : 23rd November 2017
- Please perform this assignment in Group of TWO
- Provide the Micro-Architecture (Good to use Powerpoint/Visio to make your diagrams)
- Then code it in Verilog & provide your simulation results in a PDF document.
- You also need to perform the simulation in MATLAB using the FIXED POINT TOOL BOX. The coefficients and x[n] shall be all in Q8.8 format (Shall be discussed in Class).
- The code for DAQ & HS is also required to be provided. DAQ shall use a ROM to initialize it with the same x[n] as for matlab. Helping material provided below to help you load your ROM/RAM from a file:
  - http://www.asic-world.com/verilog/memory_fsm1.html
  - https://www.xilinx.com/support/documentation/sw_manuals/xilinx13_1/xst_v6s6.pdf
- Your output should match that of a corresponding simulation performed by yourselves in MATLAB. Make your own evaluation framework that allows you to test your code.
  - Your MATLAB simulation should generate a x[n] data file for your DAQ. DAQ should import it using $readmemh. After your simulation your HS should store the results in a memory and you should dump the memory in a file. Then import it back in MATLAB to show that both the MATLAB and Xilinx results match.
- To be submitted:
  - Zipped Xilinx Project Folder with source codes & simulation
  - PDF report containing
    - Micro-Architecture (Data-path + Complete FSM)
    - Your MATLAB vs Xilinx Simulation Results
  - MATLAB code
  - Fixed Point Simulation of the functionality using the MATLAB's Fixed Point Toolbox

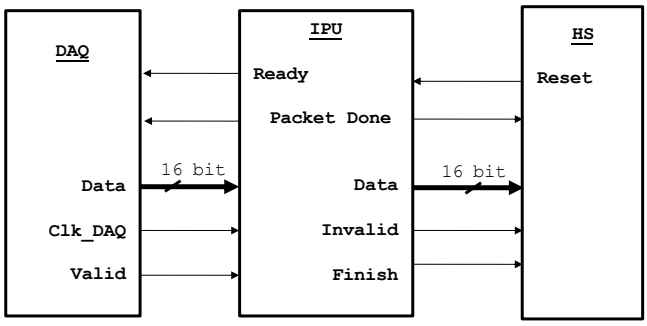
Design Specification

You have to design an IPU (Interfacing & Processing unit) that will provide an interface between a Data Acquisition Module (DAQ) and a Host System (HS). The IPU provides the functionality of applying the following processing on the received data.

$$y[n] = a\,x[n-2] + b\,x[n-4] + c\,x[n-6]$$

The figure below shows the interfacing blocks at a very abstract level.

The DAQ provides a new 16 bit sample (x[n]) on every positive edge of its clock on its primary Data line (Data). The data is organized in the form of packets where each packet consists of TWELVE 16-bit consecutive samples of x[n]. The DAQ also provides a "Valid" signal that defines the window in which the packets are valid. Only valid packet needs to be processed in the IPU. Non-valid packet/data will be discarded.



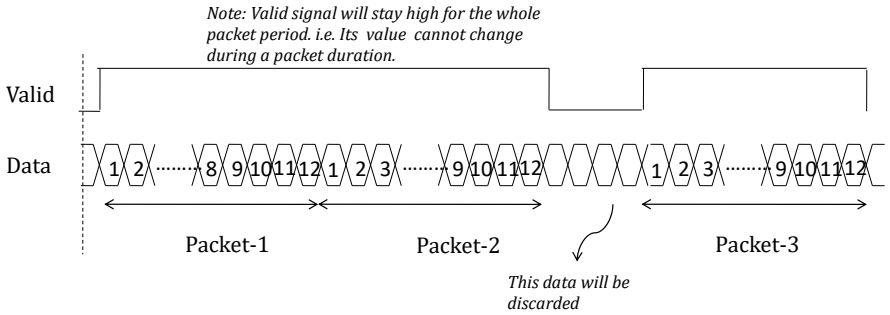
The required functionality of the IPU is explained below:

1. Whenever the HS wants to receive the data; or whenever the HS wants to reset the IPU; the HS gives a single cycle "reset" signal to IPU. This also initializes the IPU output to ZERO.
2. Starting from the next clock cycle; the IPU sends a "ready" signal to the DAQ. This informs the DAQ that the HS is ready to get the data from DAQ via IPU
3. When the DAQ has valid data ready; it responds by sending a continuous stream of data in the form of packets.

4. A single cycle “Packet Done” signal is generated by IPU after processing of every packet.
5. A single cycle “Finish” signal is generated by IPU after a batch of 32 Packets is processed.
6. The IPC shall enter a wait state after the Finish signal & wait for ‘reset’ signal from HS to start the processing again.
7. In-case the input data from DAQ is in-valid; the IPU simply freezes all the computational blocks in their current state and sets the Invalid flag for the HS.

In addition to the signals shown in the figure; following functionalities are also required. There is also an internal variable Average_Value that stores the average of all the y[n] till yet. There is an input signal to IPU named TEST and an output to the IPU named TEST_Result. The functionality of these signals is as follows:

- If (TEST=1); value of TEST_Result is ONE if current y[n] is greater than average y[n]
- If (TEST=1); value of TEST_Result is ZERO if current y[n] is less than or equal to average y[n]
- If(TEST=0); value of TEST Result is ZERO



(Design Question) *You are hired as a design engineer at Silicontron. Your latest job is to develop a Micro-Architecture [Datapath & Controller] of the IPU*

a) Labelled and complete Datapath showing all data & control signal routings & connection.
b) Correct & Complete Controller FSM