M.EEC041 - Digital Systems Design

2022/2023

Laboratory project 3 - V0.1
22 Nov 2022

(due date: Dec 19th, 2022)

PROFIR – Programmable finite response filter bank

Revision history

date	notes	author
Nov 22, 2022	V0.1 - Preliminary version with first spec of the filter bank	jca@fe.up.pt

1 – Introduction and functional description

This project will build a custom digital circuit for implementing a bank of eight different finite response digital filters for processing an ultrasonic signal sampled at 1953.1 kHz (250 MHz/128), with 16 bits per sample. Each digital FIR filter can have up to 128 coefficients represented by 24 fixed-point signed numbers, with 6 integer bits and 18 fractional bits.

The filter coefficients h_k are read from a bank of 8 independent RAM memories. Each memory is organized as 32 words of 48 bits (two coefficients per memory location) and is read by a clocked synchronous interface, as described below. The memories will be implemented as dual-port memory blocks, using one write port to upload the memories with the filter coefficients (this part is not included in your project).

For each input sample, the system must compute the outputs of the 8 digital filters, producing 8 digital signals with the same sampling frequency as the input signal. The output is produced by calculating the discrete convolution between the input signal (represented by samples x_j) and the filter impulse response h_k (the filter coefficients):

$$y_n = \sum_{k=0}^{N-1} h_k \, .x_{n-k}$$

In this expression, the values x_{n-k} represent the N previous input samples, h_k are the N filter coefficients and y_n is the output sample to calculate. The maximum FIR length is 128 and sorter filters can be implemented just by filling the non-existent coefficients with zeros.

Your design will implement only the digital filters and will be integrated in a complete design that includes the memories holding the filter coefficients, the interface to upload those memories from an external system and the rest of the digital processing system.

The system must be designed as single clock domain synchronous digital circuit, clocked at 250 MHz. The Verilog code below represents the interface of the system to design:

```
module profir(
       input                clock,        // Master 250 MHz clock, active in the rising edge
       input                reset,        // Master reset, synchronous, active high
       input signed  [15:0] datain,       // Input signal sample
       input                din_enable,   // When 1, a new sample is present at datain input (lasts 1 clock)
       output         [5:0] coeffaddress, // Address to read all the coefficient memories
       input signed  [47:0] coeff0,       // Coefficient read data for filter 0
       input signed  [47:0] coeff1,       // Coefficient read data for filter 1
       input signed  [47:0] coeff2,       // Coefficient read data for filter 2
       input signed  [47:0] coeff3,       // Coefficient read data for filter 3
       input signed  [47:0] coeff4,       // Coefficient read data for filter 4
       input signed  [47:0] coeff5,       // Coefficient read data for filter 5
       input signed  [47:0] coeff6,       // Coefficient read data for filter 6
       input signed  [47:0] coeff7,       // Coefficient read data for filter 7
       output signed [15:0] dataout0,     // Output data of filter 0
       output signed [15:0] dataout1,     // Output data of filter 1
       output signed [15:0] dataout2,     // Output data of filter 2
       output signed [15:0] dataout3,     // Output data of filter 3
       output signed [15:0] dataout4,     // Output data of filter 4
       output signed [15:0] dataout5,     // Output data of filter 5
       output signed [15:0] dataout6,     // Output data of filter 6
       output signed [15:0] dataout7      // Output data of filter 7
       );
```

The 8 RAMs holding the filter coefficients can be read in parallel with a single address that is connected to the address buses of all memories. The read port of the memories that provide the coefficient data to the filter bank is implemented by the following Verilog process:

```
always @(posedge clock)
  coeffout <= RAMCOEF[ coeffaddress ];
```

If the coefficient memory address **coeffaddress** is set by your circuit in clock cycle K, the memory output data register **coeffout** is loaded in clock cycle K+1 and your system can only use that value in clock cycle K+2. As the memory read access is pipelined, a different memory address can be read at each clock cycle.

The calculation of all the 8 output samples must be completed before the arrival a new input sample. Thus, the computation of the 8 digital filters must be completed within 128 clock cycles (equivalent to a computational complexity of 4 giga integer operations per second).

The design must target the Spartan6 FPGA and minimize the global logic complexity, measured as the number of main FPGA resources: lookup-tables, flip-flops and DSP48 blocks.

The project stages to complete are:

1. Design of the Verilog HDL code and functional verification with logic simulation
2. RTL Synthesis
3. Simulation post-synthesis
4. Integration in a FPGA project (a complete project will be provided, ready to receive your module)
5. Final synthesis and place and route
6. Post-P&R simulation
7. Final report