SystemC and Electronic System Level Design Methodology

Assignment 4, 2024-03-18

Abstract

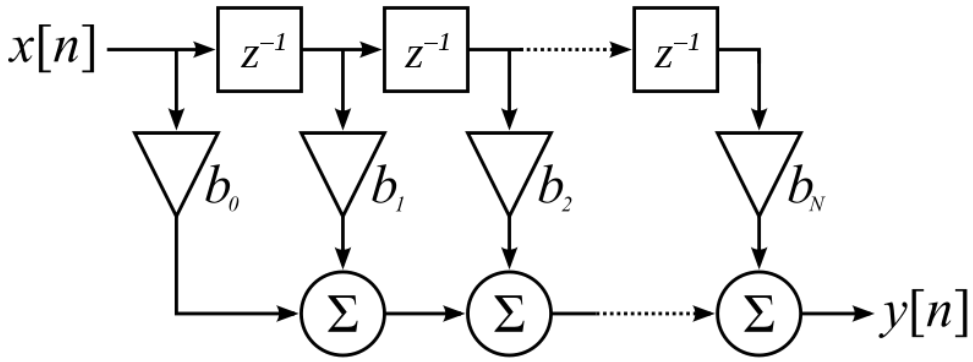
Implement a 16th-order discrete-time Finite Impulse Response (FIR) digital filter, in 200MHz clock.

Please read carefully. All outputs required are described in the text. Five (5) points will be taken for each bug, missing required output and behavior.

The 16th-order FIR module with a `SC_CTHREAD` process

Description

1. A schematic of N^{th}-order discrete-time FIR filter is given below



2. Use above schematic as the specification and implement a non-pipeline 16th-order FIR filter, in `SC_MODULE` with a `SC_CTHREAD` process, for which the module must be named as `FIR16`. You also must name SystemC file as `FIR16.h` and `FIR16.cpp`. This is to make it easier to compile your code using my makefile. Fail to do so will be penalized with 5 points.
3. The input port is named `x` and its data type is `sc_uint<32>`.
4. The output port is named `y` and its data type is `sc_uint<32>`.
5. The positive triggering clock port is named `clk`. The synchronous active-low reset pin is named `rst`.
6. Let us do a *Moving Average Filter*, a.k.a. *boxcar filter*, where all $b_i = 1/(N + 1)$. Therefore, in this case $b_i = 1/17 \approx 0.05882353$.
7. Let us use a fixed point system of `wl=32` and `iwl=16` for all

computations. Then b_i = `0x00000F0F`. However, this is just to explain how you going to set b_i values in the module. Notice that in your code there is no need to use fixed point data types. Just use `sc_uint<32>`.

8. The reset behavior is to reset all delays to 0.

sc_main

Description

1. Create a test suite, i.e., `sc_main`, and you must name the file `main.cpp`, that
   - Instantiate the `FIR16` modules
   - Read 64 input data, one-by-one and feed into the `x` port of the `FIR16`, from a file named "`firData`." Again, you must name the file exactly as specified above to get away from a 5 points penalty.
2. Create a trace file named `RESULT.vcd`. And trace ports are shown in the following order:
   - `clk`
   - `rst`
   - `x`
   - `y`

makefile

Description

1. A `makefile` must be provided, with proper modifications to your environment.

Please turn in the source codes and `makefile` only. Do not turn in the executable.

Due date

2:00 PM, March 25, 2024

Score weight (towards the final grade) 5%