Final Project: Design of an FIR Filter

Tasks:

1. Design an FIR filter from Scratch
    - Design a Bandpass filter that passes 35-200Hz
    - Use the Matlab script provided to determine the correct coefficients
    - Write a SystemVerilog module that computes the FIR filter output
    - Write a top module that instantiates FIR filter with SSE module
        i. Takes the input and golden output together
        ii. Pass the FIR filter output to SSE module to compute the SSE
        iii. Output both the FIR output and SSE output together
    - Module name: FIR
    - Input: clk, rst, stop
      [31:0] in
    - Output: next, ready
      [31:0] out
    - Handshaking:
        - rst initializes the module
        - ready goes high when ready to give result
            - give result at the clock cycle that ready goes high
            - Don't go high when no result to give
        - next goes high when ready to receive the next input
            - receive the next input at the next clock cycle
        - stop goes high the clock cycle after next goes high and there's no input to give
    - Module name: FIR_SSE
    - Input: clk, rst, stop
      [31:0] in, out_gold
    - Output: next, ready
      [31:0] out_filt, out_sse
    - Handshaking:
        - rst initializes the module
        - ready goes high when ready to give result
            - give result at the clock cycle that ready goes high
            - Don't go high when no result to give
            - Output the filter output and the corresponding SSE together
        - next goes high when ready to receive the next input
            - receive the next input at the next clock cycle
        - receives the input and the golden output together
        - stop goes high the clock cycle after next goes high and there's no input to give
2. Synthesize the top module
    - Use **ZYNQ-7 ZC702 Evaluation Board**

- Submit a screenshot of the utilization table of post-implementation
    - Make sure the numbers aren't greater than 100%

Note:

- This is a group project with **two people per group.** Please do not switch between groups
- Use 1ns timescale; **10 ns clock cycle** (i.e. 5ns high, 5ns low)
- Objective is to optimize your speed in terms of clock cycle, use whatever method you can think of.
- Hint: your throughput does not have to be equal to your latency

Submission:

- Please only submit one project per group, specify your partner's full name, PID in screenshot.pdf
- The module name, inputs and outputs are all provided in the write-up. Please do name your module accordingly.
- Your file name should be exactly same as the module name, with all the submodules within the same file.
    - adder, multiplier, and SSE module should be at separate files
- Please make sure the codes you turn in are compilable.
- Please only submit the required files specified, all other files will be ignored.
- Do not turn in your testbench