VLSI Architectures
Assignment #3
Due Date: Mon, Feb 26th, 2018

1. Consider the 8 tap FIR filter which you implemented in Assignment #1.

    a. Design the isomorphic architecture of the filter and implement it in Verilog.
    b. Design the iteratively decomposed architecture for the above isomorphic architecture with only one multiplier and one adder and other additional circuitries and implement it in Verilog.
    c. Design coarse-grain pipelined architecture for the filter such that only one multiplier lies along its critical path and implement the design in Verilog.
    d. Calculate area, throughput, latency, critical path delay and maximum operating frequency of all the architectures manually (show all the calculations) and compare the results through a table. For area calculations assume area of multiplier to be 2 units and area of adder to be 1 unit. Delay of multiplier is 3ns, adder is 2 ns and rest of the components is 1ns.

2. Suppose you are asked to design a real-time sequential VLSI architecture for a mathematical expression $y(k) = \max\{ x(k), m(k)\} + \max[0, \{23 – 5 * | x(k) – m(k)|\}]$ where $k = \{0. 1. …. N-1\}$.

    a. Design a sequential isomorphic architecture for this mathematical expression. Please assume that the quantization is of 8 bits and implement it in Verilog.
    b. Pipeline the above isomorphic architecture and implement it in Verilog.
    c. Calculate area, throughput, latency, critical path delay and maximum operating frequency of all the architectures manually (show all the calculations) and compare the results through a table. For area calculations assume area of multiplier to be 2 units and area of adder and rest of the components to be 1 unit. Delay of multiplier is 3ns, adder is 2 ns and rest of the components is 1ns.
    d. Verify whether the pipelined architecture is hardware efficient or not by making use of the analytical formula.

Deliverables:

1. Include architecture diagrams in the report (neatly hand drawn or generated on a PC).
2. Write test benches to test the above HDL implementations and include them in your simulations.
3. Clearly show the simulation waveforms in the report and compare the results with the timing results of the isomorphic architecture.
4. Feed same input to all the architectures and show that the output is same irrespective of the architecture by making use of the simulation waveforms.