System-on-Chip (SoC) Design

EE382M.20, Fall 2018

Homework #3

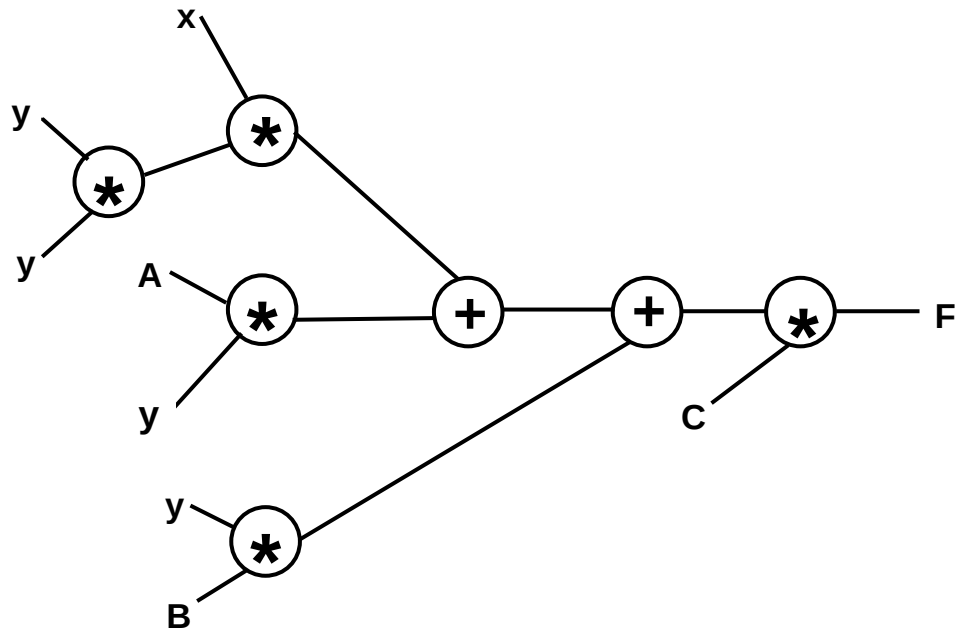
Assigned: October 16, 2018
Due: November 1, 2018

Instructions:

- Please submit your solutions via Canvas. Submissions should include a single PDF with the writeup and single Zip or Tar archive for source code.
- You may discuss the problems with your classmates but make sure to submit your own independent and individual solutions.

Problem 1: High-Level Synthesis (100 points)

Consider the following dataflow graph (DFG) of a computation and an RTL resource library that contains 2-input adders with one cycle latency consuming *E* units of energy per addition and non-pipelined 2-input multipliers that require two cycles and 4*E* energy per multiplication.



(a) Apply behavioral optimizations to the DFG in order to minimize the tree height in cycles, i.e. the ASAP schedule length. How many cycles does it take to execute the computation assuming unlimited resources?

(b) Apply behavioral optimizations in order to minimize the energy consumption without increasing the minimal latency required for the overall computation. What is the energy required for the original DFG, your DFG from (a) and the energy-optimized DFG?

(c) Assuming a resource allocation of one adder and one multiplier, apply a list scheduling algorithm using operation mobility as priority to schedule the graph into a minimum number of cycles. Show the steps of the algorithm and the final schedule and latency obtained.

(d) Derive the ILP formulation for a minimum-latency, resource-constrained scheduling problem from (c). Use binary decision variables $x_{i,j}$ to indicate whether operation i starts in cycle j. Formulate an objective function that minimizes latency and show the ILP equations/inequalities for unique start times, dependencies and resource constraints. Make sure that your solution from (c) satisfies all your constraints.

(e) Use the left-edge algorithm to determine a minimal set of registers and a corresponding register binding for your solution from (c).

(f) For your implementation in (e), draw a multiplexer-based realization of your final datapath and show the state machine of the controller driving the datapath computation.