System-on-Chip (SoC) Design

ECE382M.20, Fall 2023

Homework #3

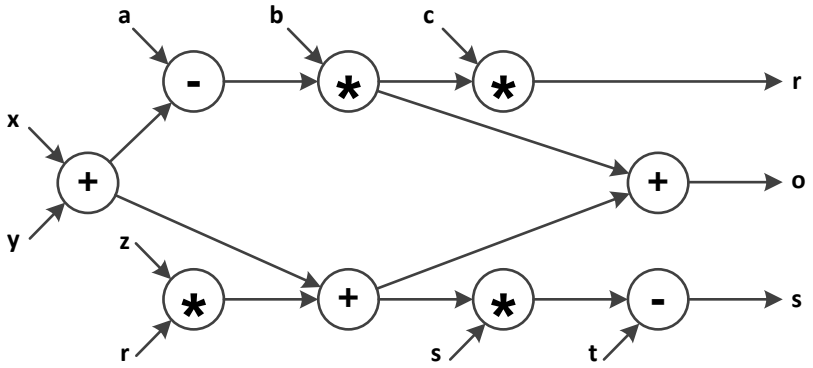
Assigned: October 12, 2023
Due: October 26, 2023

Instructions:

- Please submit your solutions via Canvas. Submissions should include a single PDF with the writeup and single Zip or Tar archive for any 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):



(a) Assuming all operations execute in one cycle, can the graph be optimized and rewritten to minimize the ASAP schedule length? Show any such modifications and the ASAP and ALAP schedule of the final graph including all mobilities?

(b) Given a resource library that contains (non-pipelined) multipliers with 2 cycles latency and ALUs that require one clock cycle to execute. Assuming a resource allocation of one ALU and one multiplier, apply a list scheduling algorithm using an operation's distance to the sink as priority to schedule the original (unoptimized) graph into a minimum number of cycles. Show the final schedule and latency obtained.

(c) Assuming that input variables are already stored in separate external registers, use the left-edge algorithm to determine a minimal set of registers and a corresponding register binding for the intermediate variables in the scheduled DFG obtained in (b). Is a binding of the given schedule into a fewer number of registers possible or is the result optimal?

(d) Sketch the ILP formulation for a minimum-latency, resource-constrained scheduling problem from (b). 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 (you don't need to show every equation if it is just repetitive, but indicate if and how that is the case). Validate that your solution from (b) satisfies all your constraints.

(e) **(Extra credit)** Encode and solve the ILP formulation from (d) using an ILP solver program, and verify that the solution is a valid schedule. For example, you can use the `lp_solve` program installed on the LRC machines. Make sure to submit all ILP solver input and output files.