Embedded System Design and Modeling

EE382N.23, Fall 2019

Homework #3

Synthesis

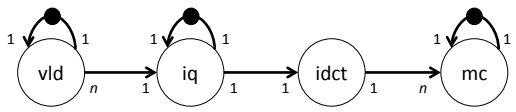
Assigned: October 31, 2019
Due: November 12, 2019

Instructions:

- Please submit your solutions via Canvas. Submissions should include a single PDF with the writeup and a single Zip or Tar archive for any supplementary files (e.g. source files, which has to be compilable by simply running 'make' and should include a README with instructions for running each model).
- You may discuss the problems with your classmates but make sure to submit your own independent and individual solutions.
- Some questions might not have a clearly correct or wrong answer. In general, grading is based on your arguments and reasoning for arriving at a solution.

Problem 3.1: Multi-Processor Mapping

Given the following dataflow graph of a H.263 decoder, where *n* is the number of macro-blocks in each image frame. Explore mappings of this graph onto a heterogeneous smartphone platform consisting of a quad-core ARM processor and a GPU with the following actor execution times (N/A means that the actor can not be mapped onto the respective processor type):



	ARM	GPU
vld	25	10
iq	1	N/A
idct	1	N/A
mc	10	5

(a) What is the repetition vector for this SDF graph? Convert the graph into an equivalent homogeneous SDF (HSDF) model according to its repetition vector, i.e. show the precedence graph for this example.

(b) Set $n = 3$ and find a latency-minimized mapping and scheduling of one iteration of the graph onto the quad-core ARM processor without utilizing the GPU. You can, for example, apply a list scheduler using the maximum distance to the sink (i.e. length of the longest/critical path with the maximum total execution time to reach the sink) as priority function. What is the latency of your schedule and how many ARM cores do you need?

(c) Schedule the graph (with $n = 3$) in a pipelined fashion, where one or more consecutive iterations can overlap. What is the highest throughput that can be achieved by running on the quad-core ARM processor only? Show a schedule that achieves this throughput while minimizing latency. What is the latency of your schedule and how many cores do you need?

(d) Now schedule the graph (with $n = 3$) utilizing both the quad-core ARM and the GPU. Show a schedule that achieves minimal latency, and a schedule that achieves maximal throughput with minimized latency. What is the latency and throughput, and how many cores do you need in each case?