System-on-Chip (SoC) Design

EE382M.20, Fall 2023

Homework #2

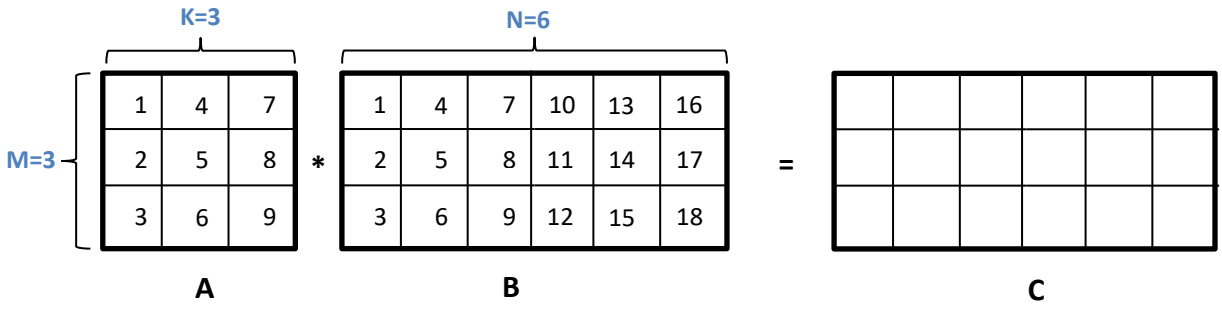
Assigned: September 21, 2023
Due: October 5, 2023

Instructions:

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

Problem 1: Execution Time Analysis (30 points)

Assume that we have a fully associative cache with 120 bytes capacity, where each cache line is 3 integers (i.e., 3*4 = 12 bytes) wide. The cache is initially empty and uses an LRU replacement policy with write-back. Given the following matrix multiplication using matrices with column-major layout, i.e. where elements are laid out in memory in the order as shown in the figure:

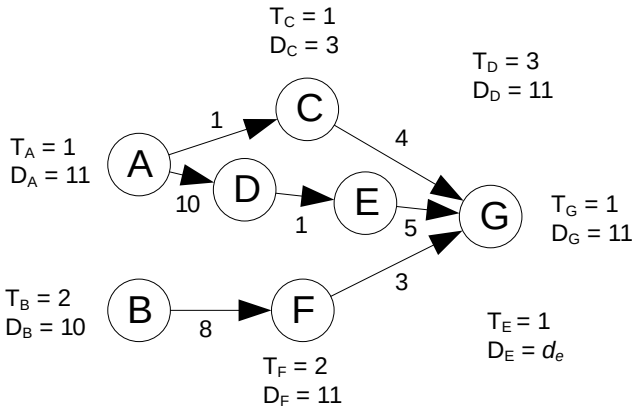


(a) What is the execution time of performing this matrix multiplication using the naïve GEMM algorithm from Problem 1(c) in Homework 1, assuming each multiply-accumulate operation and each cache hit takes 1 cycle, and each cache miss takes 10 cycles?

(b) What is the execution time of a blocked GEMM as described in Problem 1(d) in Homework 1, with a block size adjusted to the cache size to minimize execution time?

Problem 2: Partitioning (35 points)

Consider the following task graph where communication costs indicate the number of kBytes transferred between tasks (ignore task execution times and task deadlines for this problem).



(a) Apply a hierarchical clustering algorithm to partition the graph onto two processors with a shared system bus where the cost of clustered nodes is the sum of communication delays with other nodes. Show the graphs and costs after each clustering step and the final partition.

(b) Apply a Kernighan-Lin algorithm, appropriately adapted to account for partitions of unequal size if necessary, on the two-processor solution from (a). Can Kernighan-Lin improve the communication cost further? Is there a different cut that achieves a lower communication cost?

Problem 3: Scheduling (35 points)

(a) Assuming a partitioning of the graph from Problem 2 onto two processors connected via a single shared bus with a bandwidth of 1kB per time unit such that tasks A, D run on processor P0 and tasks B, C, F, E, G on processor P1, schedule the partitioned graph with task execution times, communication times and deadlines as shown in Problem 2 using an EDF* strategy on each of the two processors. You can assume that any communication within processors takes zero time and that independent communication and computation can occur overlapped at the same time. Show the schedule for the smallest d_e for which the partitioned graph remains schedulable.

(b) Can a smaller d_e be achieved using a different partitioning and scheduling? If yes, show a mapping that also minimizes total schedule length/makespan (i.e. finish time of sink task G).