Massachusetts Institute of Technology
6.848J/18.435J: Theory of Parallel and VLSI Computation
Professor Tom Leighton

Handout 2
September 3, 1997

Tentative Outline of Lectures

Lecture 1: Organization and Overview. Introduction to parallel computation on fixed-connection networks. Notions of global and local control. The bit and word models of computation. Pipelining. Elementary algorithms for sorting on linear arrays. Measures of performance such as total time, processor count, work, speedup and efficiency. Scaling and simulating large networks on smaller networks.

Lecture 2: Introduction to lower bounds and proofs of optimality. Elementary algorithms for sorting and counting on trees. Properties of the fixed-connection network model. Carry-lookahead addition and parallel prefix computations. Segmented prefix. Index computations.

Lecture 3: Carry-save addition. Integer multiplication and convolution on a linear array. Division and Newton's method. Matrix multiplication, solving systems of equations, and Gaussian elimination on systolic arrays. Matrix inverse and determinant. Stability considerations.

Lecture 4: Solution of tridiagonal systems using parallel prefix and odd-even reduction. Iterative methods. X-trees and pyramid/multigrid networks. Retiming semisystolic circuits to obtain systolic circuits.

Lecture 5: Retiming, continued. Elimination of broadcasting. Application of retiming to palindrome recognition, and priority queues. Analysis of slowdown caused by retiming.

Lecture 6: Derivation of Gaussian elimination algorithm by retiming. Graph algorithms on two-dimensional arrays: transitive closure, shortest paths, breadth-first search and minimum weight spanning trees. Sorting algorithms on two-dimensional arrays. The 0–1 sorting lemma. Odd-even transposition sort.

Lecture 7: Sorting, continued. Packet routing on a two-dimensional array. Algorithms on higher-dimensional arrays. Efficient simulations on high-dimensional arrays on low-dimensional arrays.

Lecture 8: The two-dimensional mesh of trees. $O(logN)$-step algorithms for packet routing, sorting, matrix-vector multiplication, Jacobi relaxation, pivoting, convolution, integer multiplication, division, root finding, iterated products, and related problems on a two-dimensional mesh of trees.

Lecture 9: Minimum spanning trees and connected components. Pointer jumping. Three-dimensional meshes of trees. Fast algorithms for matrix multiplication, and graph problems on a 3-d mesh of trees.

Lecture 10: Matrix inversion and related problems from linear algebra. Multidimensional meshes of trees. Hypercubes. Simulation of arrays and trees on a hypercube.

Lecture 11: Simulation of meshes of trees by a hypercube. Levelled algorithms and normal algorithms. Derivation and structure of the butterfly, cube-connected–cycles, shuffle–exchange graph, and de–Bruijn graph. Similarities among the networks. Simulation of normal algorithms. Some card tricks.

Lecture 12: Fast Fourier Transforms on a butterfly. Applications to polynomial arithmetic and convolution. Odd-even merge sort on hypercubic networks.

Lecture 13: Sorting circuits. Packet routing on hypercubic networks. Many-to-one routing and combining. One-to-many routing and broadcasting. h–relations. Routing models: store–and–forward, circuit-switching, wormhole. Off-line routing on a Beneš network. Universality of hypercubic networks.

Lecture 14: The greedy algorithm; analysis of worst-case behavior. Lower bounds on oblivious routing. Packing, spreading, and monotone routing. Reduction of one–to–many routing to one–to–one routing. Reduction of many–to–one routing to sorting. Batcher–Banyan networks. Average case analysis of greedy routing algorithms for circuit switching–the BBN algorithm.

Lecture 15: Average-case bounds on queuesize for store–and–forward routing on butterflies. Delay sequence arguments and average–case bounds on time for the greedy algorithm.

Lecture 16: Ranade's algorithm and routing with $O(1)$–size queues on arbitrary levelled networks. Randomized routing algorithms. Locality–preserving randomized routing–the CM–5 approach. Fat–trees. The advantages and disadvantages of randomly hasing the memory. Hashing with only $logN$–wise independence. Randomly-wired variations of the butterfly.

Lecture 17: Expanders and multibutterflies. Deterministic routing in $O(logN)$–steps on a multibutterfly. Routing around faults on a multibutterfly. Flow–control protocols. Circuit–switching with 100% throughput on a multibutterfly. Nonblocking algorithms for a multiBeneš network.

Lecture 18: PRAM models. Variations on concurrent writes. Efficient implementation of matrix and graph algorithms on PRAMs. Simulation by hypercubic and multibutterfly

networks. Pointer jumping and parallel prefix. Reducing the number of processors to achieve efficient speedups. Symmetry breaking.

Lecture 19: Treefix computations and expression evaluation. Tree contraction and the Euler tour technique. Other abstract models of parallelism. The scan and multiprefix models of computation. Application to quicksort and radix sort.

Lecture 20: Formal definitions of models for parallel and sequential computation from complexity theory. Uniformity. Reductions, and P–completeness of straightline code. P–completeness of ordered Lexicographically–First Maximal Independent Set (LFMIS) and other problems. Lower bounds on memory required for shifting, convolution, addition, sorting and matrix multiplication. Compact list representation schemes. Crossing sequence arguments, and time lower bounds based on communication complexity. Time–bandwidth tradeoff for transitive functions. Application to convolution, integer multiplication, sorting and matrix multiplication.

Lecture 21: Review of VLSI technology and Mead–Conway design rules. Abstractions of VLSI circuits. Discussion of the Thompson grid model. Tradeoffs between time and geometric measures such as chip area, wire crossings, and wire length. The planar separator theorem.

Lecture 22: The H–tree layout. Optimal layouts for arrays, meshes of trees, multidimensional arrays, hypercubic networks, and fat–trees. Measuring computation performance in terms of area.

Lecture 23: Area–universality. The DRAM model.

Lecture 24: Topics and open questions.