CS256
Applied Theory of Computation

Parallel Computation I

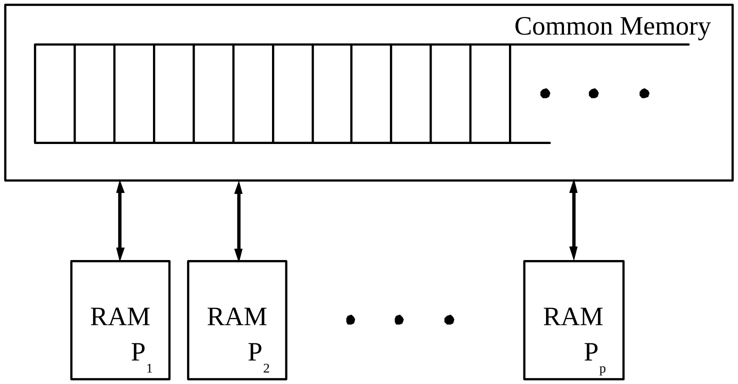
John E Savage

Overview

- Machine and network models
- Performance metrics
- Flynn's taxonomy
- Networked computers
- Amdahl's law
- Multidimensional mesh models
- Sorting on 1-D arrays
- Matrix multiplication on 2D arrays

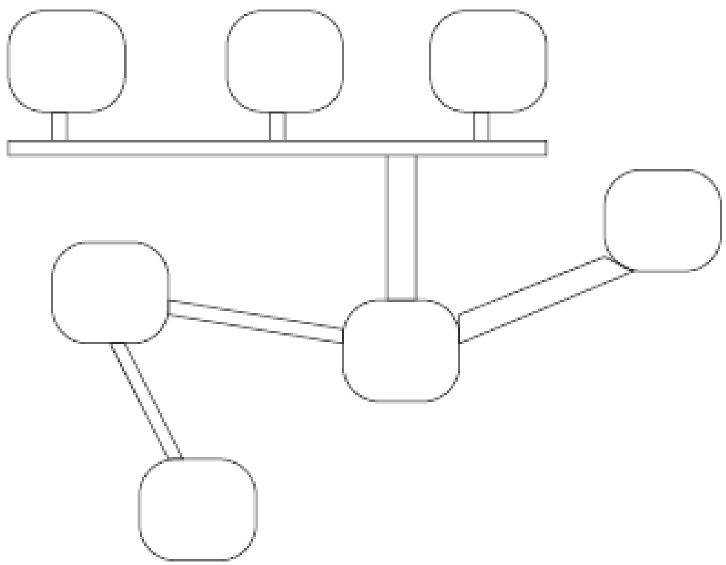
Machine Models

- Memoryless serial and parallel machines
- Serial machines: RAM & TM
- Parallel machines with memory
  - Fine- vs coarse-grained computers
  - PRAM: p RAMs with shared memory

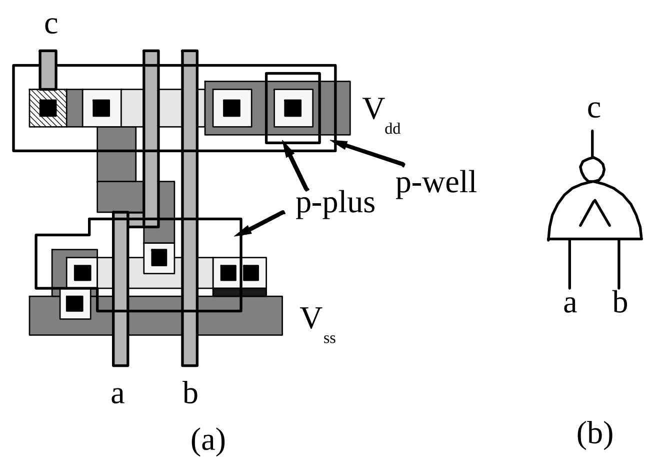


Network and VLSI Models

- Loosely coupled computers



NAND



Performance Metrics

- Logical and algebraic circuits
  - Circuit size and depth
- RAM and TM
  - Time (no. steps) & space (no. locations)
- Parallel machines
  - Time, no. processors, & space
- Memory hierarchies
  - I/O time vs primary storage space
- Distributed computing
  - Time $T(n)$ to send length n message over single channel satisfies $T = l + nb$ where l is latency and b is bandwidth.

Flynn’s Taxonomy

- SISD (single instruction, single data)
  - Single thread of control accessing one datum on each time step
- SIMD (single instruction, multiple data)
- MISD (multiple instruction, single data)
- MIMD (multiple instruction, mult data)
  - Multiple threads of control accessing multiple data on each time step
- The *data parallel model* realizes the SIMD style of programming. One thread of control but with many parallel operations, such as *parallel prefix*.

Networked Computers

- Such computers are modeled by a graph.
  - Vertices represent processors and edges denote connections between processors.
- Examples of important networks:
  - Trees, linear and multidimensional arrays. Examples of the latter coming.

Crude Bounds on Parallel Computing

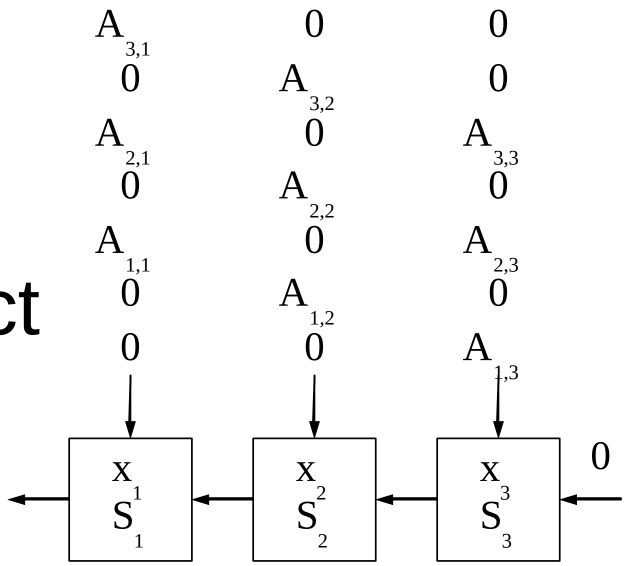
- **Amdahl's Law** If a fraction *f* of a (legacy) program's execution time on a serial machine is parallelizable, the *speedup S* achievable by the program on a *p*-processor RAM satisfies the following inequality.

$$S \le \frac{1}{\left((1-f) + \frac{f}{p}\right)}$$

- Thus, if *f* = 90% (which is large), $S \le 10$, which is not big even if *p* is infinite.
- **Efficient parallel programs are generally very different from efficient serial ones.**

Multidimensional Mesh Models

- Matrix-vector multiplication on a 1D systolic array.
- On each cycle S_i is equal to the sum of S_{i+1} and the product of x_i with the vertical input.

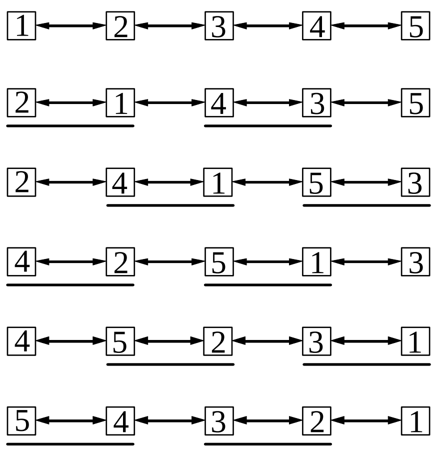


Multidimensional Mesh Models

- Systolic arrays - cells operate in synch, as shown above.
- 2D array processors have connections along NSEW axes. Toroidal connections possible.
- Higher dimensional meshes possible
- How can cells be numbered in an array?

Sorting on a 1D Array

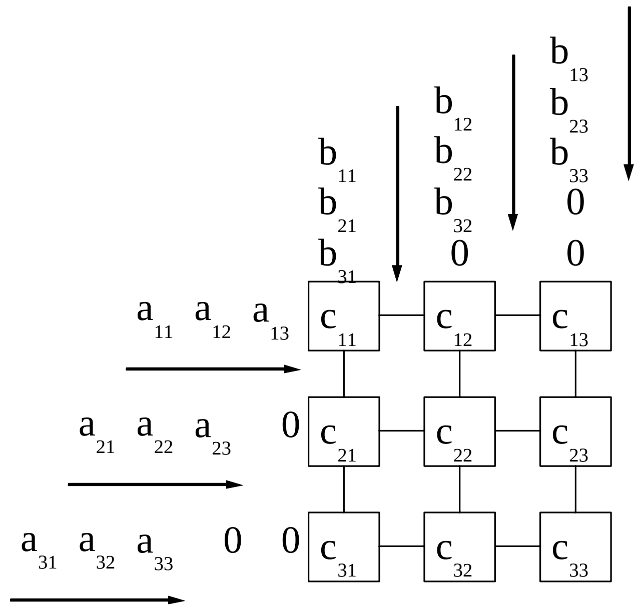
- Bubble sort compares adjacent elements and "bubbles up" the largest elements.



Sorting on a 1D Array

- This can be implemented on a linear array whose elements are numbered *1, 2, ,..., n* when *n* is even by alternating the following operations:
  - Ccomparing and swapping, if necessary, *i* and *i+1* for *i* odd
  - Comparing and swapping, if necessary, *i* and *i+1* for *i* even
- The textbook gives a proof that this procedure correctly sorts.

Matrix Multiplication on a 2D Array



- As two values enter a cell they are multiplied and added to the current value which is 0 initially.
  - Why does this algorithm correctly compute the matrix product?
- This algorithm multiplies two $n \times n$ matrices in $4n - 2$ steps.
  - How many steps are needed to compute the last of the results and then deliver the results to the output?