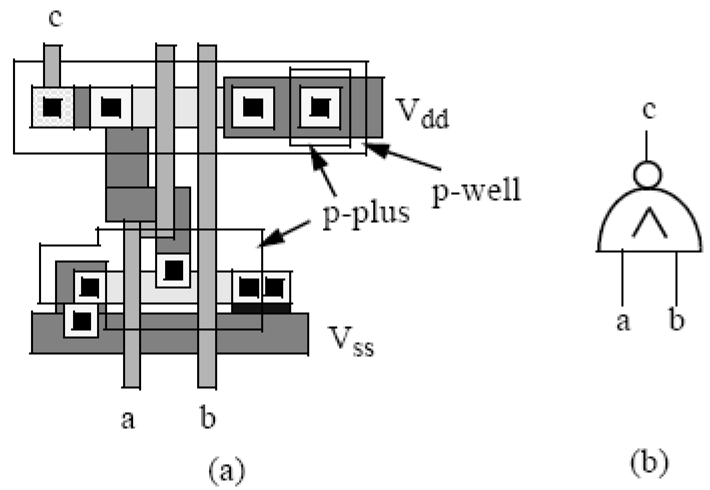
CS159 Introduction to Computational Complexity

The VLSI Model I

The VLSI Revolution

- **Time Line**:
  - 1947 - transistor invented
  - 1958-9 - integrated circuit
  - 1970's - small CPU on chip
  - 2000's - > 10^8 transistors/chip
- Gates constructed of overlays of rectangular sections of materials.



The VLSI Model

- **Architectural Model**:
  - Chips realize FSMs
  - Wires are rectilinear.
  - Wires have bounded width and separation λ.
  - Gates can be binary or non-binary.
  - Gates/memory cells occupy area proportional to λ^2.
  - I/O pads also occupy area proportional to λ^2.
  - Wires on at most $v \geq 1$ levels. Gates on one level.
  - Time for signal to travel wire of length l is constant.
  - (Time for *diffusion model* is proportional to l^2!)

The VLSI Model

- **Performance Measures**:
  - Chip area A - manufacturing cost increases with A
  - Number of steps T - reflects computational cost

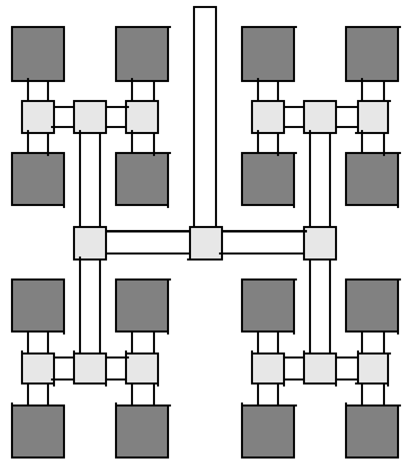
The VLSI Model

- **Functional Model**: Finite functions computed $f : B^n \rightarrow B^m$ where B is any set, usually {0,1}.

- **Algorithmic Model**: Limitations on I/O
  - Inputs provided at times and places on the chip that are data-independent.
  - Each input is supplied once (*semellective alg.*) or provided multiple times (*multilective alg.*).
  - Outputs are produced once at times and places on the chip that are data-independent.

Chip Layout

- Trees are very important. The H-tree uses area well. Leaves & interior nodes are square



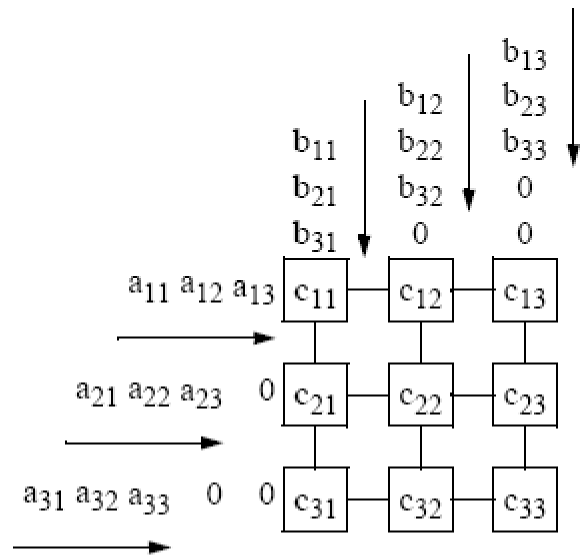
- Let H-tree have 4^k leaves.

H-Tree Layout

- Let H-tree **leaves** have area b and **interior nodes** have area c.
- Let S(k) = length of H-tree side with 4^k leaves
  - S(1) = 2b + 1. H-tree has 4 leaves
  - $S(k) = 2S(k-1) + c = (b+c)2^k - c$
- Area of H-tree with $n = 4^k$ leaves, $A(n) = S^2(k)$
  - $A(n) \leq n(b+c)^2$

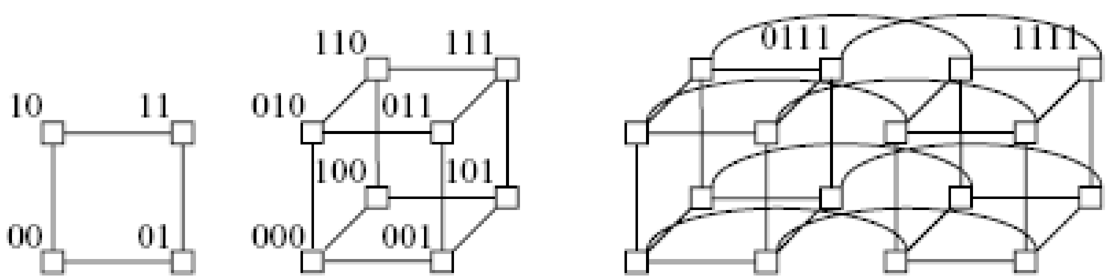
VLSI Architectures

- Tree-based computations
  - Matrix-vector multiplication
  - Prefix computations
- Meshes (see textbook for other examples)
  - Matrix multiplication
  - Bubble sort
- Systolic array



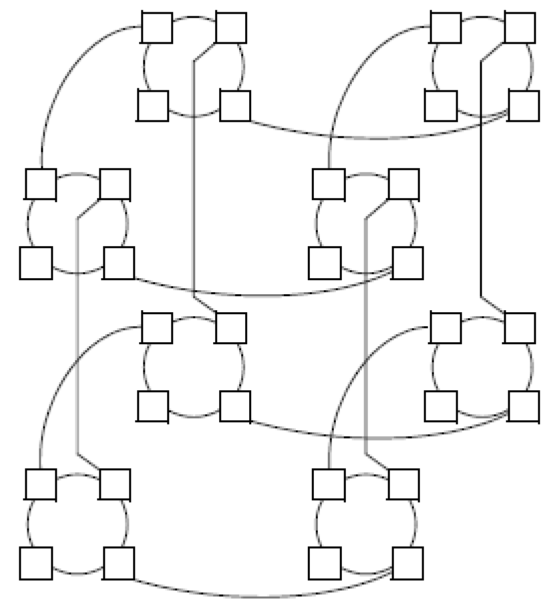
Hypercube-Based Machines

- A d-dimensional hypercube has 2^d vertices labeled by binary d-tuples. Edges between vertices whose d-tuples differ in one position.
- A d-D hypercube has $(d/2)\ 2^d$ edges. It can be formed by adding edges between corresponding vertices of two (d-1)-D hypercubes.



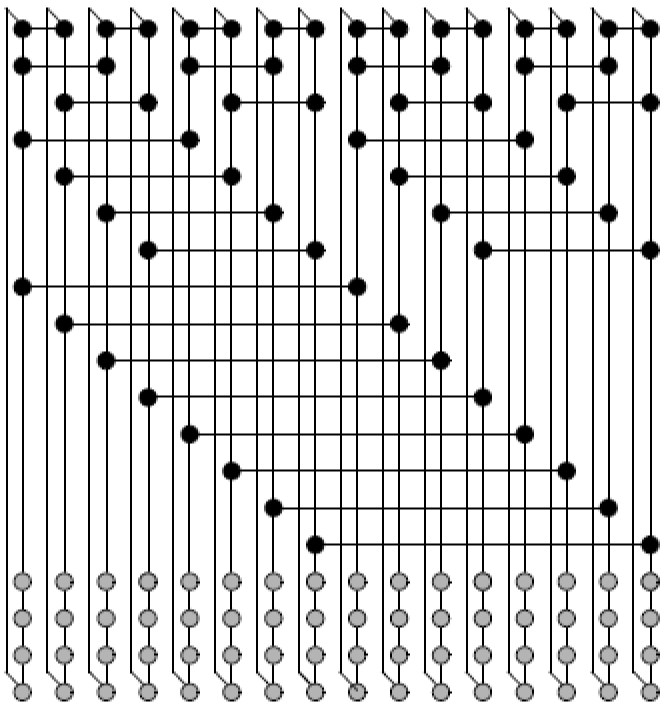
CCC Network

- Efficient layout of hypercube.
- The (k,d)-CCC network (it has 2^d cycles, 2^k vertices per cycle) shown for $k = 2$, $d = 3$.
- The jth vertex on each cycle connected to jth vertex on another cycle.



Planar Layout of CCC Network

- (k,d)-CCC network with 2^d cycles & 2^k vertices/cycle.
- Shown with $k = 3$, $d = 4$.



CCC Layout

- The network has $n = 2^{d+k}$ vertices, $2^k \geq d$, & 2^d cols.
- Index columns from left by binary tuples starting with (000...0). First (*k*th) dimension connections are made using one (2^{k-1}) row(s).
- $R_c = 1 + 2 + ... + 2^{d-1} = 2^d - 1$ rows are used for connections. $2^k - d$ more rows are used for the $2^k - d$ extra processors on each cycle.
- $R \leq 2^d + 2^k - d$ and **area** $A = R \times 2^d \leq 2^d(2^d + 2^k - d)$.
- $T = O(d + 2^k)$ steps for normal algorithm to move data across d dimensions and within cycle of length 2^k.
- When $2^k \approx d$, following theorem holds.

CCC Layout

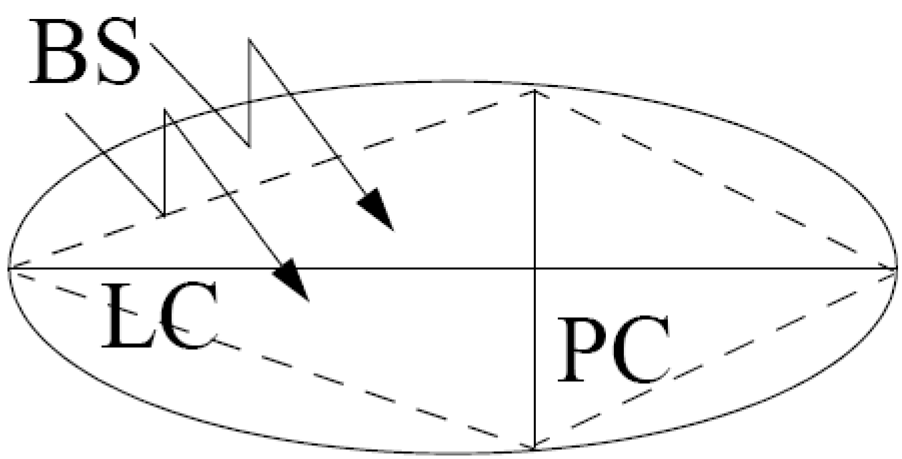
Definition A **normal hypercube algorithm** swaps information between neighbors in the same dimension of the hypercube per step.

Theorem Every fully normal algorithm for a n-processor hypercube can be implemented on a VLSI CCC network with area A and using time T satisfying the following bound when $\Omega(\log n) \leq T = O(\sqrt{n})$.

$$AT^2 = O(n^2).$$

Basic Area-Time Tradeoffs Ideas

- Convex chip boundary, feature size = λ
- LC = longest chord, PC = chord ⊥ to LC divides inputs into two equal size sets. One side, BS, has at least half of outputs.



Basic Area-Time Tradeoffs Ideas

- Find amount of information, $\mathcal{I}$, that must move to outputs in BS from inputs on other side.
- At most $|PC|/2\lambda$ wires cross PC.
- Number of steps $T \geq \mathcal{I} /(|PC|/2\lambda)$.
- Area $A \geq |PC||LC|/2 \geq |PC|^2/2$. Thus,

$$AT^2 \geq (2\lambda^2)\, \mathcal{I}^2$$

Planar Circuit Size

Definition A **planar circuit** over set X is a circuit over X that has been embedded in the plane in a way that gates do not overlap but edges may cross.

A planar circuit is **semellective** if there is a unique vertex at which each input variable is supplied. Otherwise it is **multilective**.

The **size** of a planar circuit is the number of inputs, edge crossings, and gates drawn in a planar circuit in which gates realize functions over X.

Planar Circuit Size

Definition The **planar circuit size** of a function $f : X^n \rightarrow X^m$, $C_p(f)$ (when the basis is understood), is the size of the smallest planar circuit for f over the given basis.

Planar Circuit Size

Lemma The planar circuit size $C_p(f)$ and standard circuit size $C(f)$ of $f : B^n \rightarrow B^m$ over a basis Ω are related as follows where r is the fan-in of Ω.

$$C(f) + n \leq C_p(f) \leq (rC(f))^2/2 + C(f) + n$$

Proof The first inequality is obvious. The second is straightforward to obtain. Note that there are at most $rC(f)$ wires connecting inputs and gates to other gates.

Planar Circuit Size

Proof Consider any planar embedding of a minimal circuit for *f* (containing C(*f*)) gates. It isn't necessary for two wires to cross more than once; if they cross more than once, swap their respective segments.

Since that are at most *q(q-1)/2* pairs of elements from a set of *q* elements, it follows that there are at most $(rC(f))^2/2$ crossings from which the result follows. QED

- In section 12.7 of *Models of Computation* we show cyclic shifting function nearly meets upper bound.