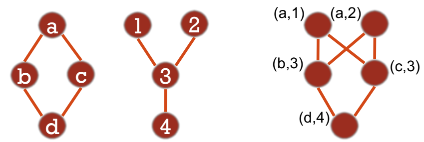
EVEN AND EVEN LAYOUT (1)

- Layout presented by G. Even and S. Even ['00], and based on the notion of Layered Cross Product
- **Def.** A layered graph *of* $l+1$ *layers* $G=(V_0, V_1, \ldots, V_l, E)$ consists of $l+1$ layers of nodes; V_i is the (non-empty) set of nodes in layer i; E is a set of directed edges: edge (u,v) connects two nodes of two adjacent layers, that is, if u lies on layer i then v lies on layer $i+1$.

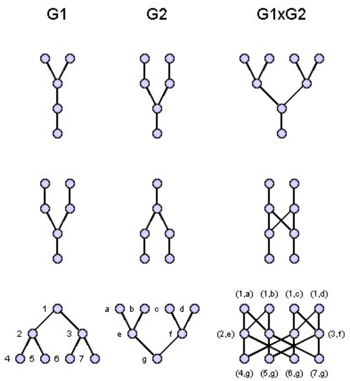
EVEN AND EVEN LAYOUT (2)

- **Def.** [Even & Litman '92] The Layered Cross Product (LCP) of two layered graphs of $l+1$ layers each, $G^1=(V_0^1, V_1^1, \ldots, V_l^1, E^1)$ and $G^2=(V_0^2, V_1^2, \ldots, V_l^2, E^2)$, is a layered graph of $l+1$ layers, $G=(V_0, V_1, \ldots, V_l, E)$, where:
  - For every $i=0, \ldots, l$, $V_i = V_i^1 \times V_i^2$ (i.e. each layer is the cartesian product of the corresponding layers in G^1 and G^2);
  - There is an edge (u,v) in G connecting nodes (u^1,u^2) and (v^1,v^2) iff (u^1,v^1) and (u^2,v^2) are edges in G^1 and G^2, respectively.



EVEN AND EVEN LAYOUT (3)

Exemples of LCP



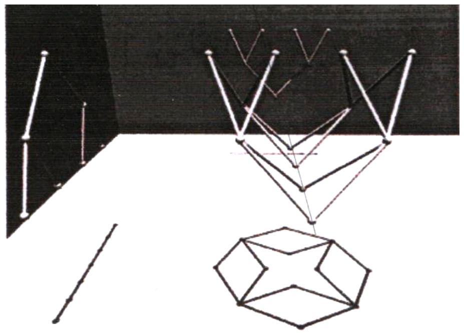
Even and Litman proved that many well known topologies are the LCP of simple structures (e.g. trees).

Specifically, the butterfly network is the LCP of two complete binary trees, an upward one and a downward one.

EVEN AND EVEN LAYOUT (4)

The Projection Methodology (PM):

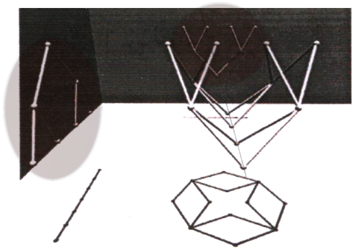
- Let G^1 and G^2 two layered graphs of $l+1$ layers each and let G denote their LCP. A layout of G is obtained with the PM as follows:



EVEN AND EVEN LAYOUT (5)

The Projection Methodology (contd)

- Consider a cube and draw the graph G^1 on the *xy* face so that
  - (a) the *y*-coordinate of every node *u* e V_i^1 equals *i*
  - (b) the *x*-coordinate of every node is an integer.
- Similarly, draw the graph G^2 on the *yz* face
- …



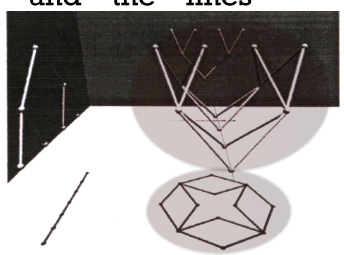
EVEN AND EVEN LAYOUT (6)

The Projection Methodology (cntd)

- … A three-dimensional drawing of the LCP *G* is constructed in the cube as follows:
  - if $u \in V_i^1$ is drawn in coordinates $(x_u, i, 0)$ and $v \in V_i^2$ is drawn in coordinates $(0, i, z_v)$, then the coordinates of node $(u,v) \in V_i$ are (x_u, i, z_v).

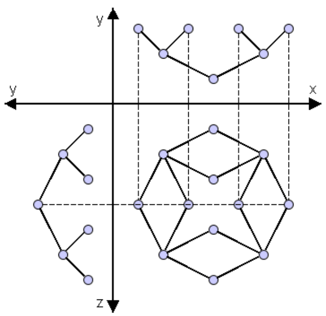
    In other words, the nodes of *G* are the intersections between the lines orthogonal to plane *xy* and passing through nodes of G^1 and the lines orthogonal to plane *yz*

    and passing through nodes of G^2.
- A 2D drawing of *G* is obtained by projecting the 3D drawing to the *xz* plane.

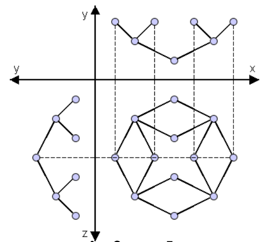


EVEN AND EVEN LAYOUT (7)

Obs. It is possible to avoid to construct the 3D representation by immediately using the prolongations on plane *xz* of the projections of nodes in layer *i* of G^1 on the *x* axis and of node in layer *i* of G^2 on the *z* axis, $i=0,\dots,l$



EVEN AND EVEN LAYOUT (8)

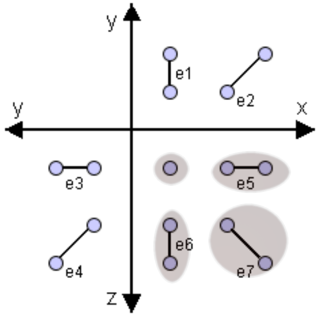


- The PM may produce layouts that do not satisfy the constraints required by the Thompson model.
  - For example, the drawing above is a grid drawing but it is not an orthogonal drawing.
- We now describe how rectilinear layouts of *G* can be obtained via the PM. First, we formalize necessary and sufficient conditions:
  - for the edges of the *xz* projection of *G* to be along grid paths,
  - for nodes to be mapped to different grid points, and
  - for not using any grid edge more than once.

EVEN AND EVEN LAYOUT (9)

Four types of edges in the product graph G:

1. The product of two diagonal edges yields a diagonal edge;
2. The product of a vertical edge and a diagonal edge yields a vertical edge;
3. The product of a diagonal edge and a vertical edge yields a horizontal edge;
4. The product of two vertical edges yelds a single grid point.



EVEN AND EVEN LAYOUT (10)

In order to get a feasible layout through the PM, we have to impose that the product of either two diagonal edges or two vertical edges never occurs.

More precisely:

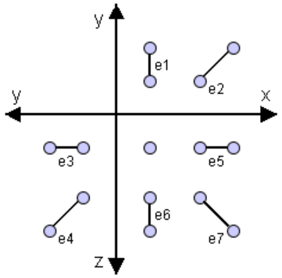
***1.** The PM generates a layout of G in which the edges are grid lines if and only if the drawings of G^1 and G^2 on the faces of the cube satisfy the following condition: For every edge e e E, exactly one of its factor is drawn diagonally.*

This claim avoids overlappings of nodes of the same layer, too.

EVEN AND EVEN LAYOUT (11)

We need to impose that nodes in different layers do not overlap:

***2.** The PM generates a layout of G in which at most one node is mapped to each grid point if and only if the sets*$\{(x_u, z_v): u \in V_i^1$ *e* $v \in V_i^2\}$ *are disjoint, for each* $i=0, \ldots, l$.

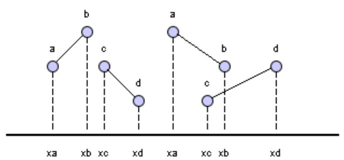


EVEN AND EVEN LAYOUT (12)

Consider now two diagonal edges (a,b) and (c,d) in G^1; the coordinates of nodes a, b, c, d are:

- node a: $(x_a, i, 0)$;
- node b: $(x_b, i+1, 0)$;
- node c: $(x_c, j, 0)$;
- node d: $(x_d, j+1, 0)$.

We say that these two edges are consistent if the open intervals (x_a, x_b) and (x_c, x_d) are disjoint.



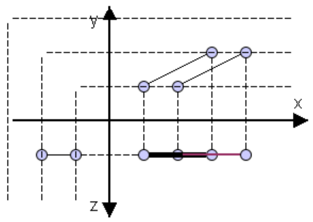
EVEN AND EVEN LAYOUT (13)

***3.** The PM generates a layout of G in which no grid edge is used twice if and only if for every two inconsistent edges of one of the multiplicands the following condition holds:*

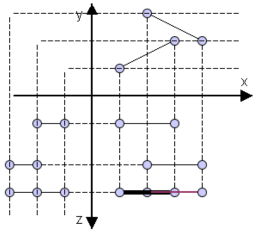
The two edges are not in the same layer of the multiplicand, and

on the two layers in which they appear, there are no (straight) edges of the other multiplicand which are collinear.

Inconsistent edges on the same layer



Inconsistent edges on different layers



EVEN AND EVEN LAYOUT (14)

In order to produce a feasible layout, we need to impose that all the three claims are satisfied.

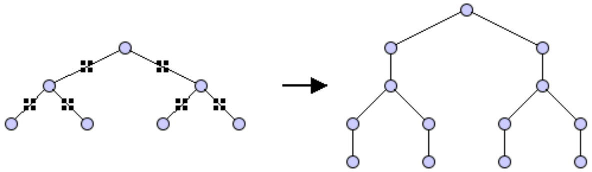
Let us consider the Claims one by one:

EVEN AND EVEN LAYOUT (15)

***1.** The PM generates a layout of G in which the edges are grid lines if and only if the drawings of G^1 and G^2 on the faces of the cube satisfy the following condition: For every edge e e E, exactly one of its factor is drawn diagonally.*

A solution is to double the number of edge levels so that edges in the drawing of G^1 are diagonal in odd layers and straight in the even layers, while the edges in the drawing of G^2 are straight in the odd layers and diagonal in the even layers.

EVEN AND EVEN LAYOUT (16)



The doubling of the number of edge levels is achieved by stretching each edge of the two multiplicands to become a path of two edges.

In this way we simulate the creations of edge bends.

EVEN AND EVEN LAYOUT (17)

2. The PM generates a layout of G in which at most one node is mapped to each grid point if and only if the sets$\{(x_u, z_v): u \in V_i^1$ e $v \in V_i^2\}$ are disjoint, for each i=0, …, l.

A simple way to guarantee that this condition will hold is to make sure that no two nodes in the drawing of G^1 (G^2), except for the two end-points of the same straight edge, share the *x*-coordinate (*z*-coordinate).

This is always possible if we opportunely enlarge the drawings of the two factors.

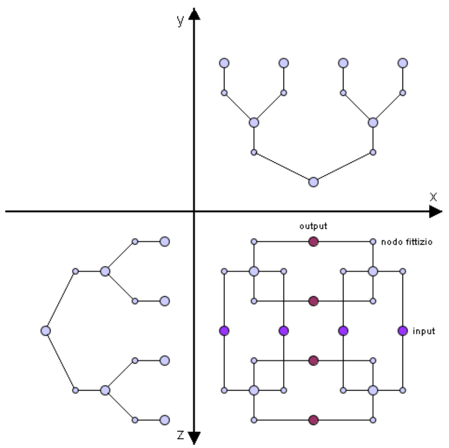
EVEN AND EVEN LAYOUT (18)

***3.** The PM generates a layout of G in which no grid edge is used twice if and only if for every two inconsistent edges of one of the multiplicands the following condition holds:*

- *The two edges are not in the same layer of the multiplicand, and*
- *on the two layers in which they appear, there are no (straight) edges of the other multiplicand which are collinear.*

This condition is harder to enforce and is a severe limitation on this technique. For this reason, we limit to networks, each of which is the LCP of two trees.

EVEN AND EVEN LAYOUT (19)



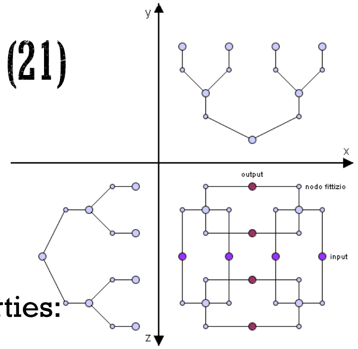
EVEN AND EVEN LAYOUT (20)

The butterfly network is the LCP of two binary trees, one drawn upward and one drawn downward. (We dedicate a column to each vertex to prevent vertices of the layout from colliding.)

Proceed as follows:

- Draw one tree next to the *xy* plane and the other next to the *yz* plane
- Construct their LCP in 3D inside the cube, in such a way that the two trees are the projections of the resulting butterfly on the *xy* and *yz* planes
- The projection of this 3D figure on the floor is a planar layout of the butterfly

EVEN AND EVEN LAYOUT (21)



This layout has the following properties:

- It's symmetric 👍
- Its height is $H=2(N-1)$
- Its width is $W=2(N-1)$
- Its area is $4N^2+o(N^2)$ 👎
- Input and output nodes are not on the boundary 👎
- All the edges on the same layer has the same length 👍

COMPARING THE TWO TECHNIQUES

- WISE - PROs:
  - relatively small area
  - It "looks like" a butterfly
  - Input/output nodes on the boundary
- WISE - CONs:
  - knok-knees
  - "slanted" grid
- EVEN & EVEN - PROs:
  - It eliminates all the flaws
- EVEN & EVEN - CONs:
  - Larger area
  - input/output nodes inside the layout

OTHER RESULTS (1)

With the aim of optimizing the layout area, other layout algorithms have been proposed:

- Dinitz ['98] proves that the area of the Even & Even layout can be decreased by means of some local adjustments, so to achieve area $11/6\ N^2+o(N^2)$
- Later, Avior et al. ['98] prove that any butterfly layout cannot have area smaller than $N^2 + o(N^2)$ if "slanted" drawing is not allowed, and they provide an algorithm producing a layout of optimal area.

OTHER RESULTS (2)

Finally, Dinitz et al. ['99] prove that, if a "slanted" drawing is allowed, area $1/2\ N^2+o(N^2)$ is necessary and sufficient.

These works definitively close the optimal area layout problem of the Butterfly network.

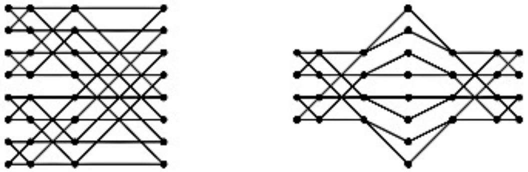
OPTIMAL AREA LAYOUT OF THE BUTTERFLY NETWORK

OPTIMAL AREA LAYOUT - IDEA (1)

- The two papers that provide an optimal area layout base their results on the following lemma:

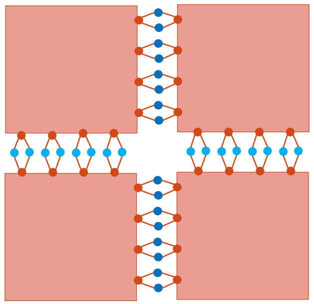
Lemma: For any non-negative integers j, k, $0 \leq j \leq j+k \leq n$, the subgraph of the n-dim. Butterfly induced by the nodes of levels $j, j+1, \ldots, j+k$ is the disjoint union of 2^{n-k} copies of k-dimensional butterflies.

- In particular, if $j=0$ and $k=n-1$:



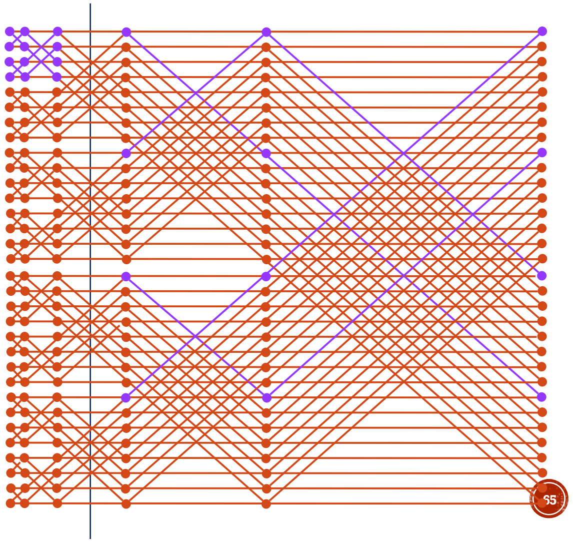
OPTIMAL AREA LAYOUT - IDEA (2)

- Hence, an $(n-1)$-dimensional Butterfly can be built as a pair of $(n-2)$-dim. Butterflies connected by one node layer and one edge layer.
- If we cut out the input and output nodes from an n-dim. Butterfly, we get:



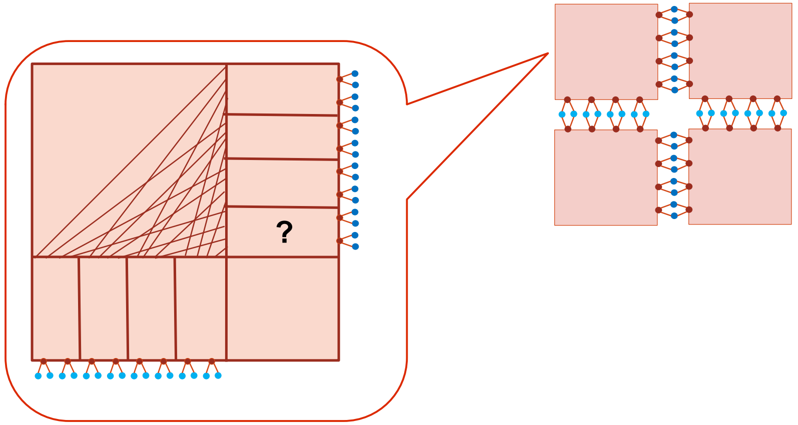
OPTIMAL AREA LAYOUT - IDEA (3)

- Each one of these $(n-2)$-dim. Butterflies can be, in turn, cut into many smaller butterflies:



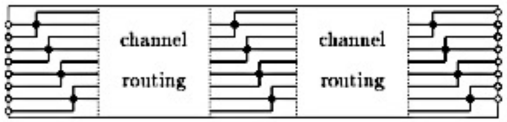
OPTIMAL AREA LAYOUT - IDEA (4)

The previous layout can be better specified as follows:



OPTIMAL AREA LAYOUT - IDEA (5)

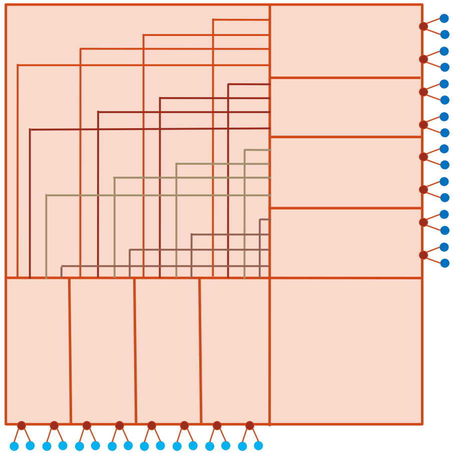
Each rectangle contains a Butterfly that can be represented, either horizontally or vertically, layer by layer as follows:



Obs.: this layout is far from being optimal; nevertheless it allows to produce a final optimal layout.

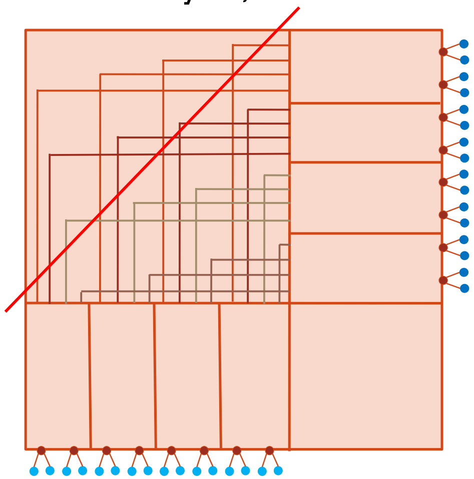
OPTIMAL AREA LAYOUT - IDEA (6)

It remains to connect the small rectangular butterflies:



OPTIMAL AREA LAYOUT - IDEA (7)

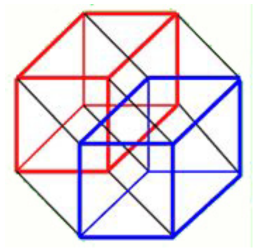
In the case of slanted layout, it can be bent along the line:



OPTIMAL AREA LAYOUT - IDEA (8)

- It is possible to prove tight lower and upper bounds on the layout area for both the models (usual and slanted).
- The interested students can look at:
  - A. Avior, T.C., S. Even, A. Litman, A.L. Rosenberg: A Tight Layout of the Butterfly Network. *Theory of Computing Systems* 31, 1998.
  - Y. Dinitz, S. Even, M. Zapolotsky: A Compact Layout of the Butterfly. *J. of Interconnection Networks* 4, 2003.

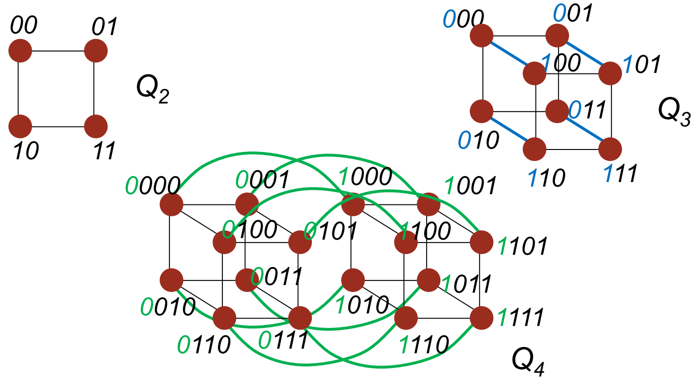
LAYOUT OF THE HYPERCUBE NETWORK



THE HYPERCUBE (1)

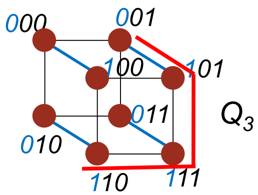
- Widely used for parallel computation, thanks to its nice properties (high regularity, logarithmic diameter, good fault tolerance, …).
- **Def.** The n-dimensional Hypercube, Q_n, has $N=2^n$ nodes and $\frac{1}{2}\, n2^n$ edges. Each node is labeled with an n-bit binary string, and two nodes are linked with an edge iff their binary strings differ in precisely one bit.
- The edges of the hypercube can be naturally partitioned according to the dimensions that they traverse and Q_n can be seen as $Q_{n-1} \equiv Q_{n-1} \cdots$

THE HYPERCUBE (2)



Q_n can be built by joining with an edge nodes in two different copies of Q_{n-1} if they have the same label.
Obs.: These edges form a perfect matching.

THE HYPERCUBE (3)

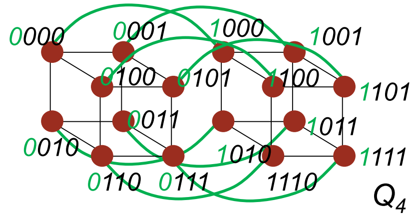


Property: Q_n has diameter *log N.*
Sketch of proof. Any two nodes
$u=u_1u_2\dots u_{logN}$ and $v=v_1v_2\dots v_{logN}$ are connected by the path:
$u_1u_2\dots u_{logN}\rightarrow v_1u_2\dots u_{logN}\rightarrow v_1v_2\dots u_{logN}\rightarrow\dots\rightarrow v_1v_2\dots v_{logN}$
There are pairs of nodes requiring exactly *log N* steps. ■

THE HYPERCUBE (3)

Reminder: The bisection width of a network is the minimum number of edges one has to cut to disconnect the network into two equally sized subnetworks.
Property. $BW(Q_n)=N/2$.

Sketch of proof. the green edges (=edges in a single dimension) divide the hypercube into two equally sized sub-networks; they are *N/2* and it is not possible to cut a smaller number of edges to get the same result.



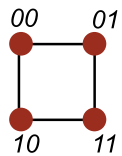
THE HYPERCUBE (5)

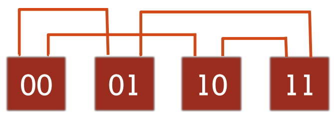
Th. A lower bound on the layout area of a network is the square of its bisection width (already proved).
Cor. Each layout of Q_n has area at least $N^2/4$.

In the following: layout with area $4/9N^2+o(N^2)$, that hence is almost optimal (far from the lower bound by a factor of 1.7) [Yeh, Varvarigos, Parhami, '99].

COLLINEAR LAYOUT (1)

- Reminder: In a collinear layout all nodes are placed on the same line. Instead of computing its area, it is usual to count the number of necessary tracks.
- We start with a 2-dim. Hypercube, and inductively move to hypercubes of higher dimensions:
  - Q_2:

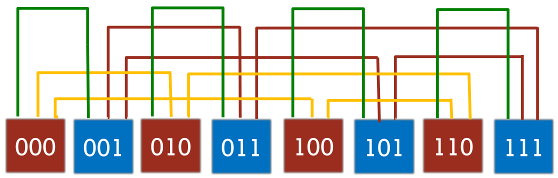




2 tracks

COLLINEAR LAYOUT (2)

- <u>If n odd</u>: Assume that we have a collinear layout for Q_{n-1} that requires $f(n-1)$ tracks: Q_n

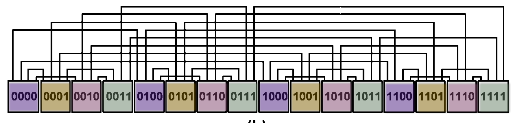


2 tracks 2 tracks 1 track

Tot. $f(n)=2f(n-1)+1$ tracks

COLLINEAR LAYOUT (3)

- <u>If n is even</u>: To obtain the collinear layout of Q_n we start with the layouts of four Q_{n-2}s:



$f(n+2)=4f(n)+2$

COLLINEAR LAYOUT (4)

- **Th.** The number of tracks required for the collinear layout of Q_n is $2/3N$ (where $N=2^n$ is the number of nodes).
- Proof. We solve the following recurrence equation:
  - $f(n)=2f(n-1)+1$ if n odd
  - $f(n)=4f(n-2)+2$ if n even
  - $f(2)=2$

  Even case:

  $f(n)=4f(n-2)+2=4^2f(n-4)+4x2+2=$

  $=4^3f(n-6)+2^5+2^3+2=\ldots=$

  $=\ldots$when $n-2k=2$ iff $k=(n-2)/2\ldots= 4^k f(n-2k) + \sum_{i=0}^{k-1} 2\cdot 4^i =$

  $$= 4^{\frac{n-2}{2}} f(2) + 2\sum_{i=0}^{\frac{n-2}{2}-1} 4^i = 2\sum_{i=0}^{\frac{n-2}{2}} 4^i \cong 2\cdot\frac{4^{\frac{n-2}{2}+1}}{3} = \frac{2}{3}2^n = \frac{2}{3}N$$

COLLINEAR LAYOUT (5)

(proof cntd)

The odd case is analogous. ■

The area of this layout is $(2/3N+n) \times (nN)$.

Th. Q_n can be laid out in $4/9N^2+o(N^2)$ area.

Sketch of Proof. Let $n=n_1+n_2$.

Let us use 2^{n1} copies of the collinear layout of Q_{n2}, each placed along a row.

We connect the 2^{n1} nodes that belong to the same column vertically according to the collinear layout of a Q_{n1}.

COLLINEAR LAYOUT (6)

(proof cntd)

Reminder: Q_k needs of

${}_{2/3}2^k$ tracks

# of horiz. tracks (rows) :

2^{n1} copies x ${}_{2/3}2^{n2}$ tracks

Additive height (nodes):

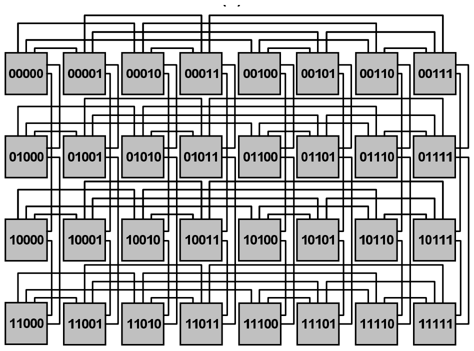
2^{n1} copies x n_2

Total heigth:

2^{n1} x $({}_{2/3}2^{n2}+ n_2) =$

${}_{2/3}2^{n1+n2}+n_2\,2^{n1}$

…



COLLINEAR LAYOUT (7)

(proof cntd)

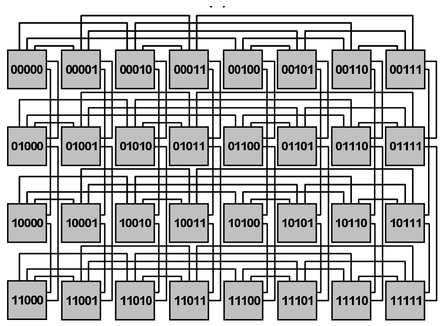
Reminder:

heigth= ${}_{2/3}2^{n1+n2}+n_2\,2^{n1}$

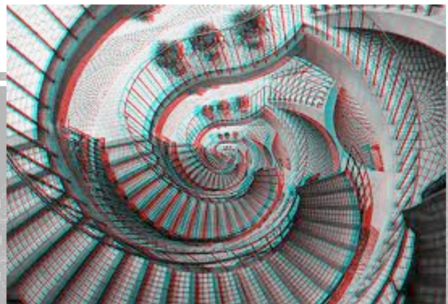
The width is computed analogously, switching the roles of n_1 and n_2.

Total width:

${}_{2/3}2^{n1+n2}+n_1\,2^{n2}$



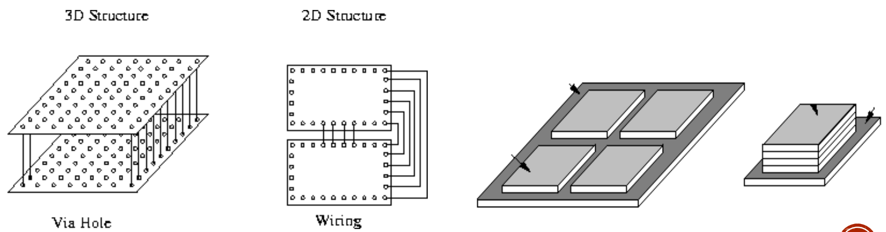
Area$=({}_{2/3}N+n_1\,2^{n2})({}_{2/3}N+n_2\,2^{n1})=$
$=4/9N^2+o(N^2)$ if n_1 and n_2 are $o(N)$, e.g. if $n_1=\Theta(n_2)\cong n_{/2}$.



3D LAYOUT

3D LAYOUT PROBLEM (1)

- The diffusion of the 3D layout has increased in the last thirty years.
- The topology is lain out on a series of slices.
- Further optimization of the wire length and number of bends
- Less silicon used.



3D LAYOUT PROBLEM (2)

Def. A 3D layout of a topology *G* is a 1-1 function between *G* and the 3D grid such that:

- the nodes are mapped into grid points
  - it is better if the nodes lie on the external slice in order to minimize: energy consuming, production of heat and difficulty of connection with other devices
- the wires are mapped on independent grid paths so that:
  - these paths are edge-disjoint;
  - there are no "knock-knees"
  - these paths do not cross any mapping of a node that is not an extreme of the corresponding wire.

Aim: minimizing the volume and keeping wires short.

3D LAYOUT PROBLEM (3)

The students interested in this topic can look at:

- L. Torok and I. Vrto. Layout Volumes of the Hypercube. *Proc. Graph Drawing '04*.
- T.C. and A. Massini. Three Dimensional Layout of Hypercube Networks. *Networks* 47, 2006.

-> possible lessons