A Hierarchical Three-Way Interconnect Architecture for Hexagonal Processors

Feng Zhou, Esther Y. Cheng, Bo Yao, Chung-Kuan Cheng, Ronald Graham

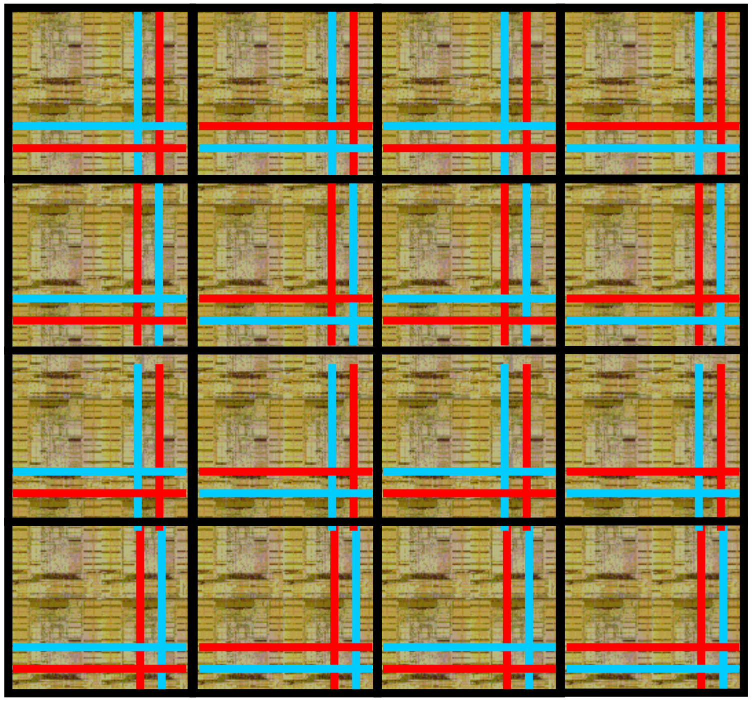
University of California, San Diego

Outline

- Background
- X-Trees and Y-Trees
- Performance Evaluation
- Representation of Hexagonal Cells
- Conclusions

Multi-Processors on a Chip

- Small microprocessor size makes multi-processors on a chip possible
- 100k trans. => Embedded MPU
- Billions trans. on a chip nowadays
- Hundreds MPUs on a chip



An example: RAW chip (MIT)

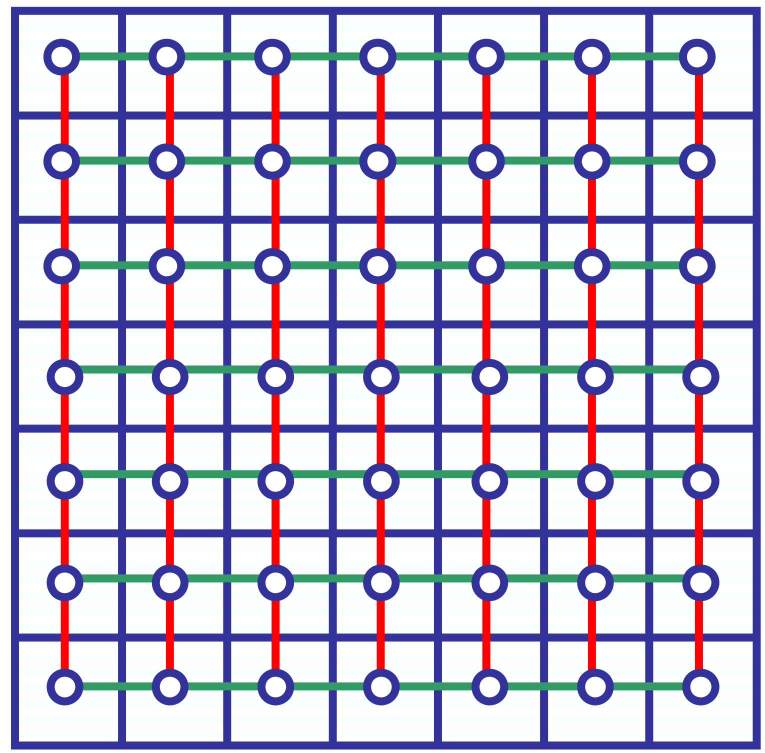
(http://cag-www.lcs.mit.edu/raw/)

Importance of Interconnect Architecture

- Device is cheap while interconnect is expensive
  - Limited routing resource for global interconnect between processors
  - Long wire in global interconnect => large delay
- Interconnect architecture determines the communication efficiency to a large extent.

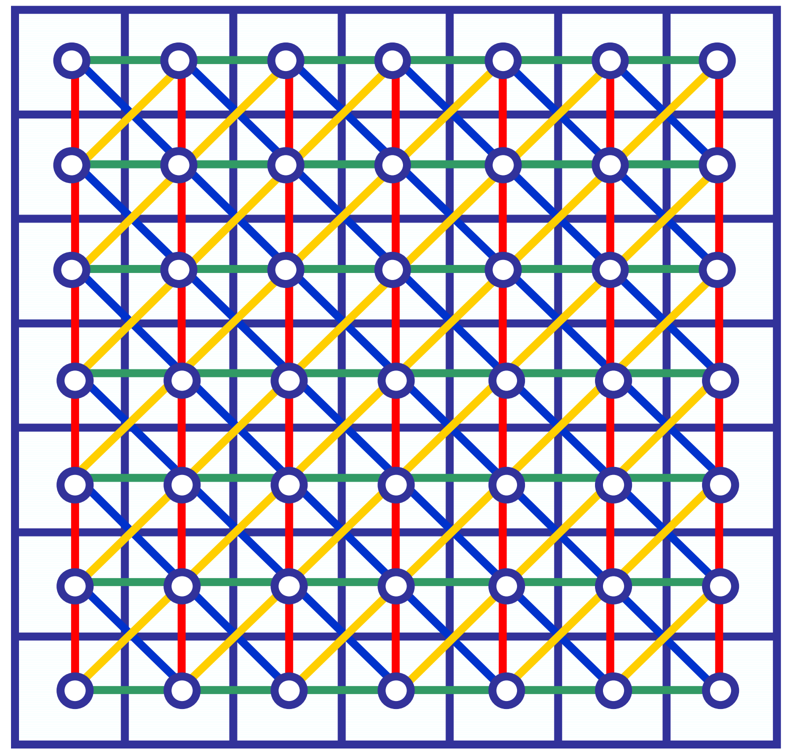
Interconnect Architectures (1)

- Manhattan
  - Two direction routing:
    - — Horizontal
    - | Vertical
  - Square cell



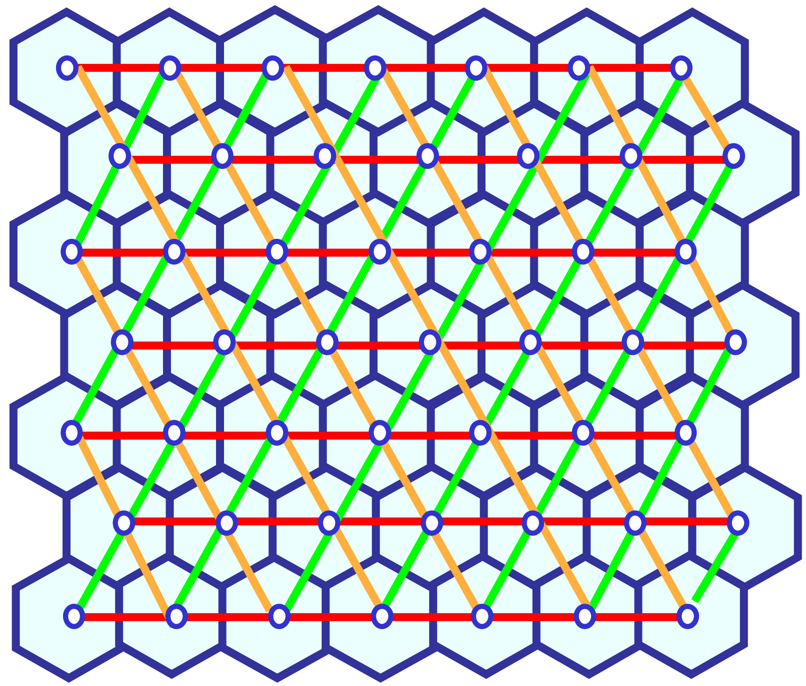
Interconnect Architectures (2)

- X architecture
  - Four direction routing:
    - Horizontal
    - Vertical
    - 45°
    - 135°
  - Square cell



Interconnect Architectures (3)

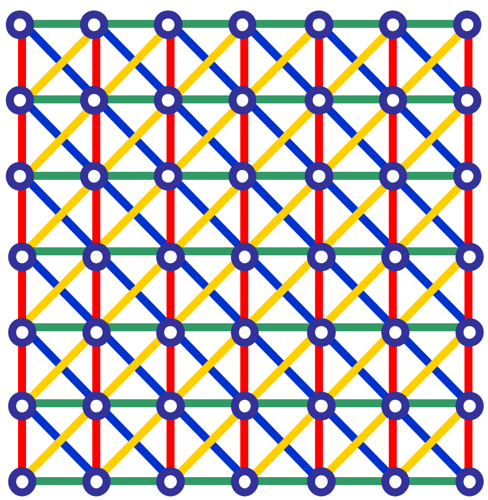
- Y-Architecture
  - Three routing directions:
    - 0°
    - 60°
    - 120°
  - Hexagon cells
  - Proposed by Chen, et al, in ASP-DAC '03

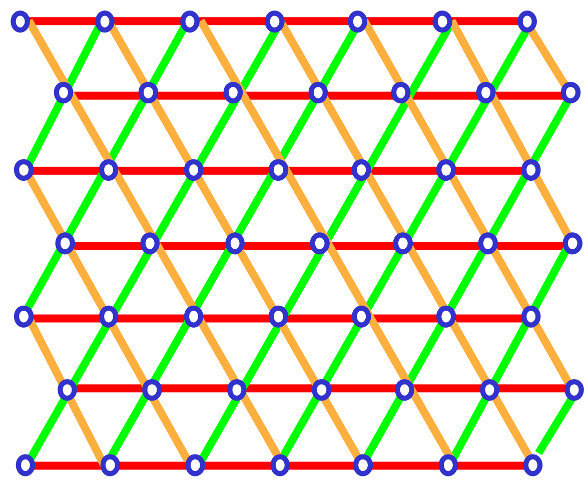


Advantage of Y-Architecture

- More routing direction => Better throughput over Manhattan (24% more)

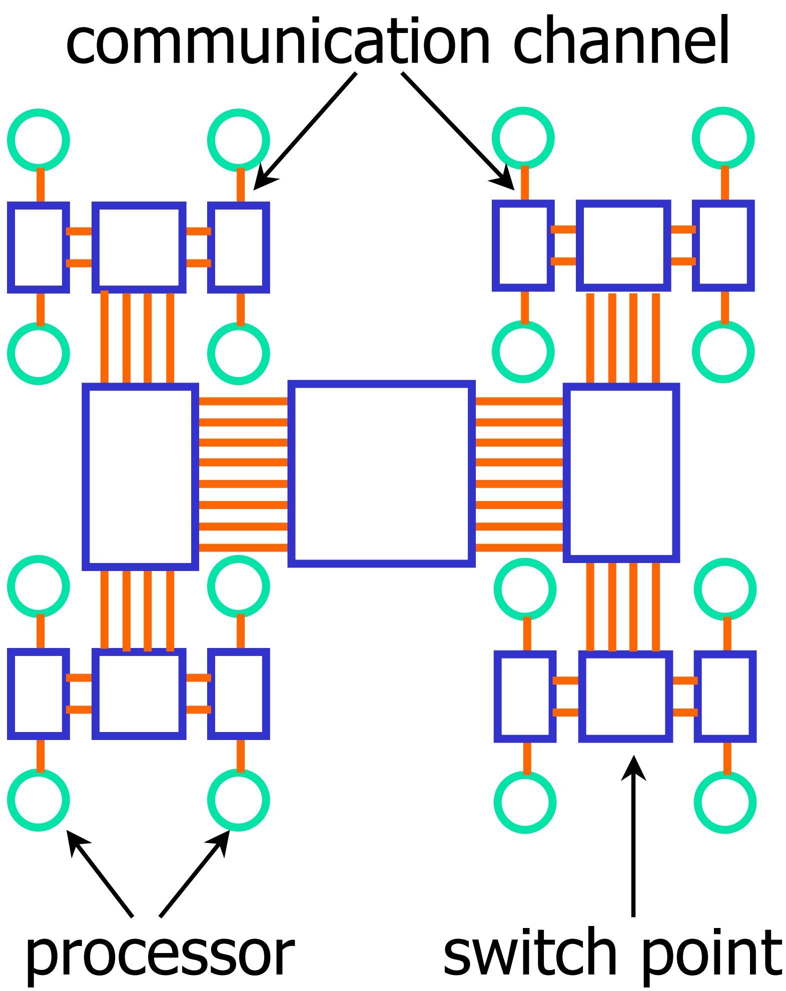
  Comparable with X (12.6% less)
- Same pitch for all routing directions. (X must use different pitch)





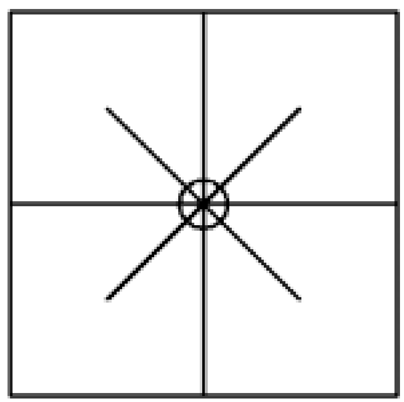
Universal Communication Networks – Fat trees

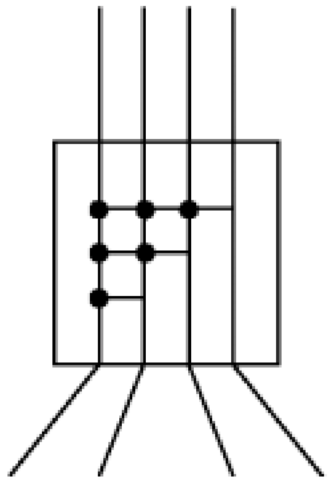
- Introduced by Leiserson, 1985
- General structure
  - Complete binary tree
  - Leaf nodes are processors
  - Internal nodes are switch points
  - Capacity of the channel increases as we go up the tree



X-Trees (1)

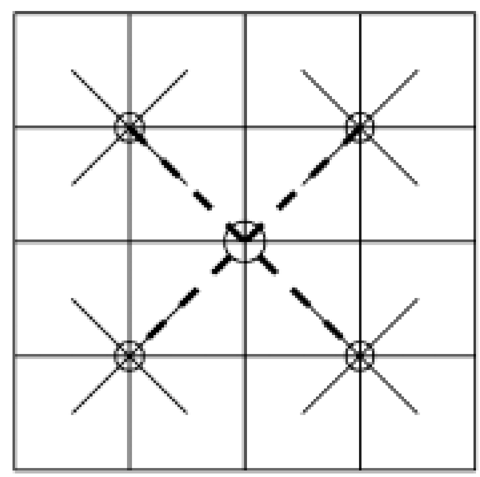
- Elements:
- A X-tree connect 4 cells
- Switch at the center



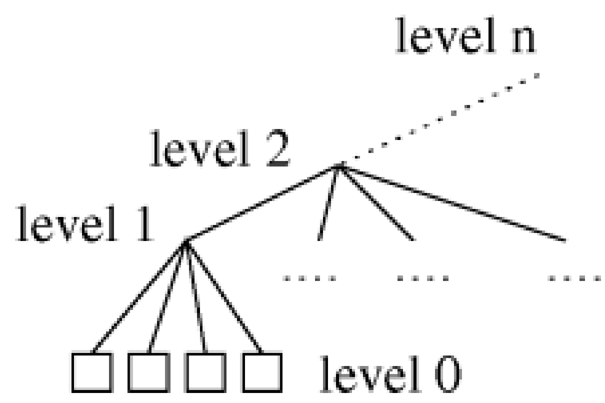


X-Trees (2)

- Expending hierarchically



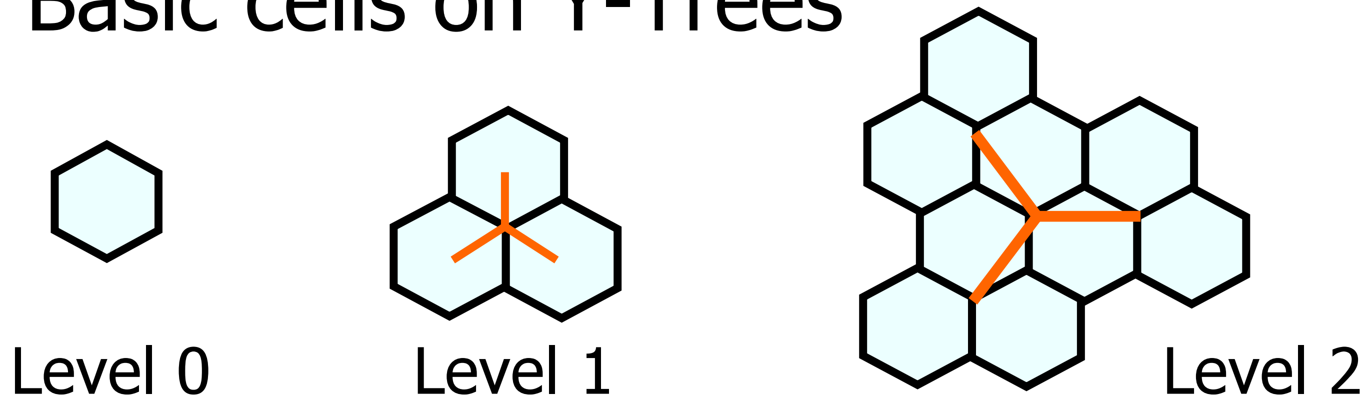
2-level X-Tree



- Can be embedded in X architecture

Definition of Y-Trees (1)

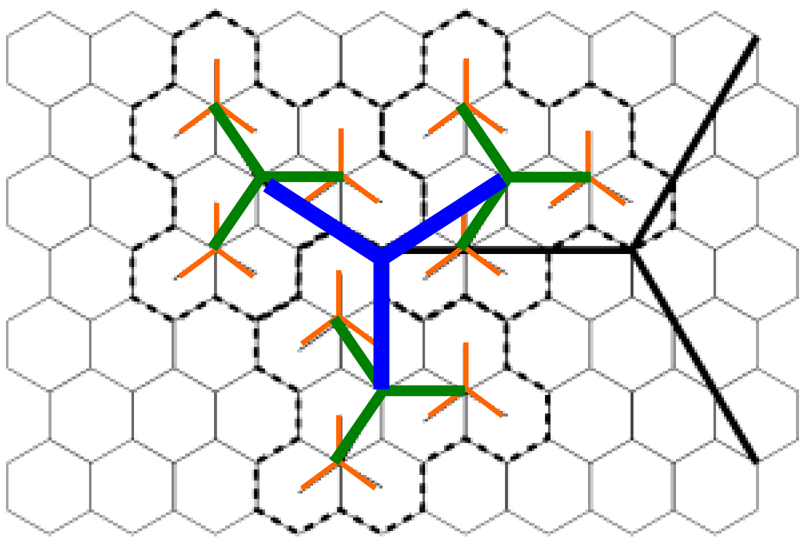
- Basic cells on Y-Trees

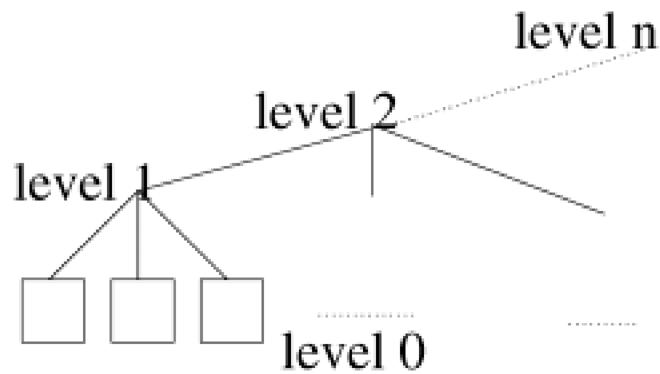


- Connect 3 cells with a “Y” structure to form a higher level cell
- Four “Y” directions

Definition of Y-Trees (2)

- Hierarchical expending





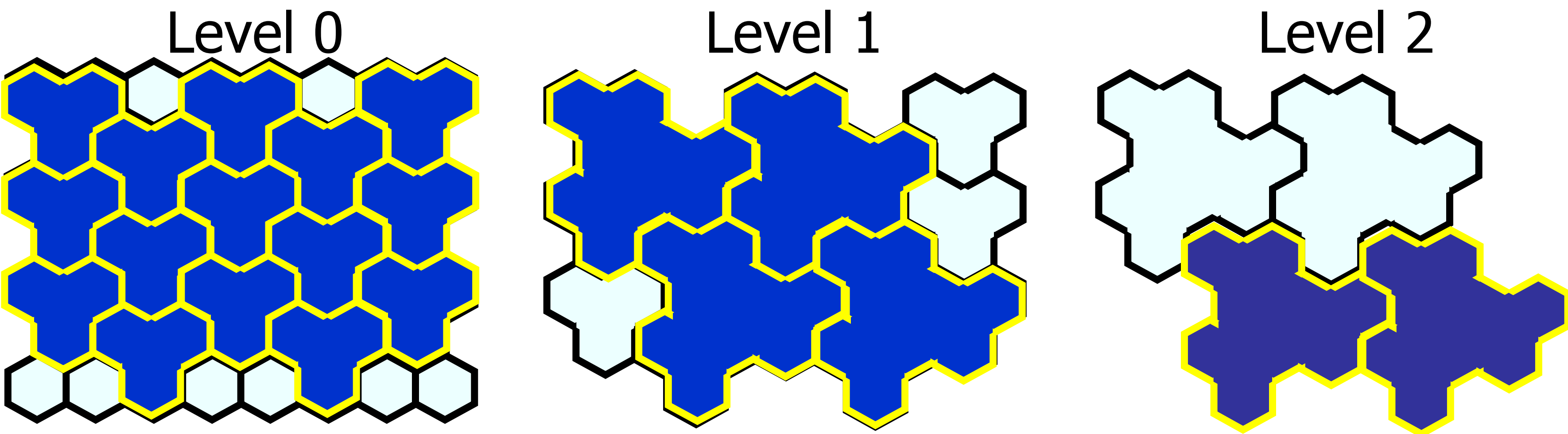
- Can be embedded in Y Architecture

Growth of Y-Trees

- Connect 3 (k-1)-level cells with a “Y” to form a k-level cell.
- Same direction of Y in each level. Direction of Y must rotate 90 degrees (positive or negative) between adjacent levels.
- Y-Tree can grow hierarchically without any empty space.

Intuitive Proof of the growth of Y-Tree

- Properties of the cell array
  1. ½ grid shift between rows (columns)
  2. Each cell is adj. to 2 cells in same row
  3. Each cell is adj. to 2 cells in each of the neighboring row (column)



Properties of Y-trees

- For a Y-Tree of n levels, there are 2^n combinations.
- a cell at the k-th level in the Y-Tree contains 3^k hexagons.
- Y-Tree can grow hierarchically and cover all the hexagons in the array without empty holes.

Performance Evaluation

- Object function: M = L * D

  where $L = \sum \text{length of each wire segment}$

  $$D = \sum_{1 \le i < j \le P} d_{ij},$$

  (d_{ij} is the distance of leaf node i and j on the tree)

  Cheng, et al, ICCD 2002
- L for the wiring resource cost
- D for power consumption due to wire capacitance
- M is smaller the better

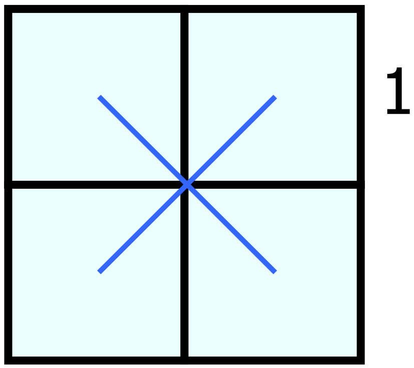
Deriving L and D for X-Trees

- Recurrence form

L_x	D_x
$L_1 = 2\sqrt{2}$ $L_n = 4L_{n-1} + 2^{3n-2}\sqrt{2}$	$D_1 = 6\sqrt{2}$ $D_n = 4D_{n-1} + 6 \cdot 2^{4n-4}\sqrt{2}(2^n - 1)$

- Closed form

$$L_x(n) = \sqrt{2}(2^{3n-1} - 2^{2n-1})$$
$$D_x(n) = \tfrac{\sqrt{2}}{14} 4^n (6 \cdot 2^{3n} - 7 \cdot 2^{2n} + 1)$$



Deriving L and D for Y-Trees

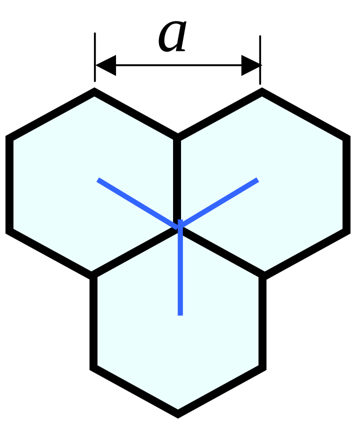
- Recurrence form

L_y	D_y
$L_1 = \sqrt{3}a$ $L_n = 3L_{n-1} + 3^{\frac{3}{2}n-1}a$	$D_1 = 2\sqrt{3}a$ $D_n = 3D_{n-1} + (3+\sqrt{3})(3^{\frac{n}{2}} - 1)3^{2n-2}a$

- Closed form

$$L_x(n) = \frac{3^n(\sqrt{3}^n - 1)}{3-\sqrt{3}}a$$

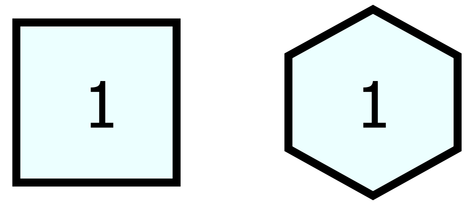
$$D_x(n) = \tfrac{3+\sqrt{3}}{78}3^n\left[(9+\sqrt{3})(3\sqrt{3})^n - 1) - 13(3^n - 1)\right]a$$



Normalization by Area

- Cell area is 1

  $$=> a = \sqrt{2} / \sqrt[4]{3}$$



- X Trees and Y Trees covers different area with same level =>
  - Normalize L and D with $A^{3/2}$ and $A^{5/2}$ (area of the tree)

    $$L_{norm} = L/A^{3/2}, \quad D_{norm} = D/A^{5/2}$$

  - Normalize M with A^2

    $$M_{norm} = M/A^2$$

  - A is the area of the tree, $A_X=4^n$, $A_Y=3^n$

Comparing X- and Y-Trees with the M Metric

- Normalized value

$$M_{Xnorm}(n) \approx \tfrac{3}{7} = 0.43 \qquad M_{Ynorm}(n) \approx \frac{11+7\sqrt{3}}{39} = 0.59$$

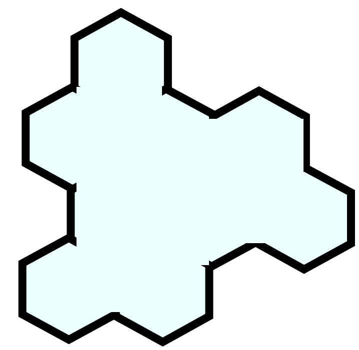
$$L_{Xnorm}(n) \approx \tfrac{\sqrt{2}}{2} = 0.71 \qquad L_{Ynorm}(n) \approx \tfrac{1}{3-\sqrt{3}} a = 0.85$$

$$D_{Xnorm}(n) \approx \tfrac{3\sqrt{2}}{7} = 0.61 \qquad D_{Ynorm}(n) \approx \tfrac{5+2\sqrt{3}}{13} a = 0.70$$

- Y-Trees are comparable to X-Trees

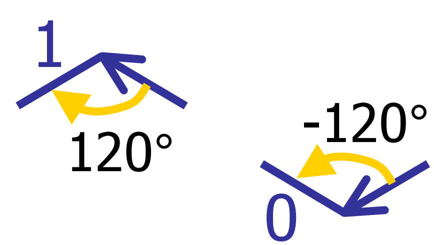
Representation of Merged Hexagonal cells

- Basic observation:
  - Only 3 directions of edge
  - Each edge makes either a 120-degree or minus 120-degree turn from the previous edge



A level 2 cell

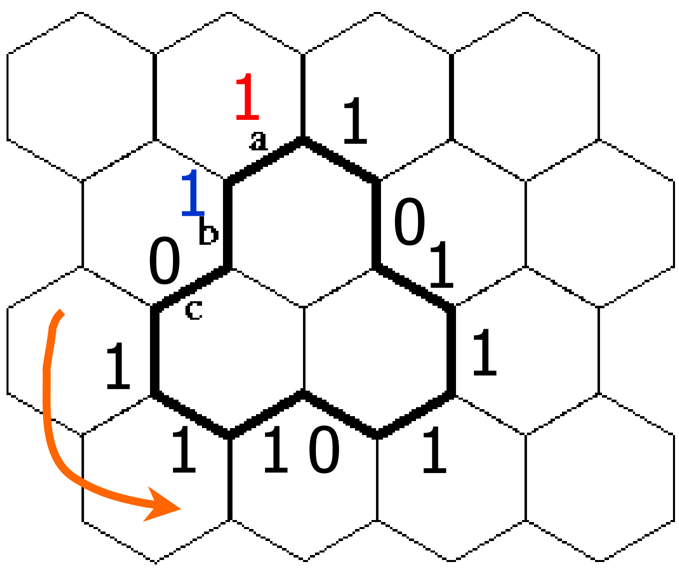
- Representation:
  - Mark each edge with
    - "1" – 120-degree
    - "0" – minus 120-degree



  - Start with the first vertical edge going down.

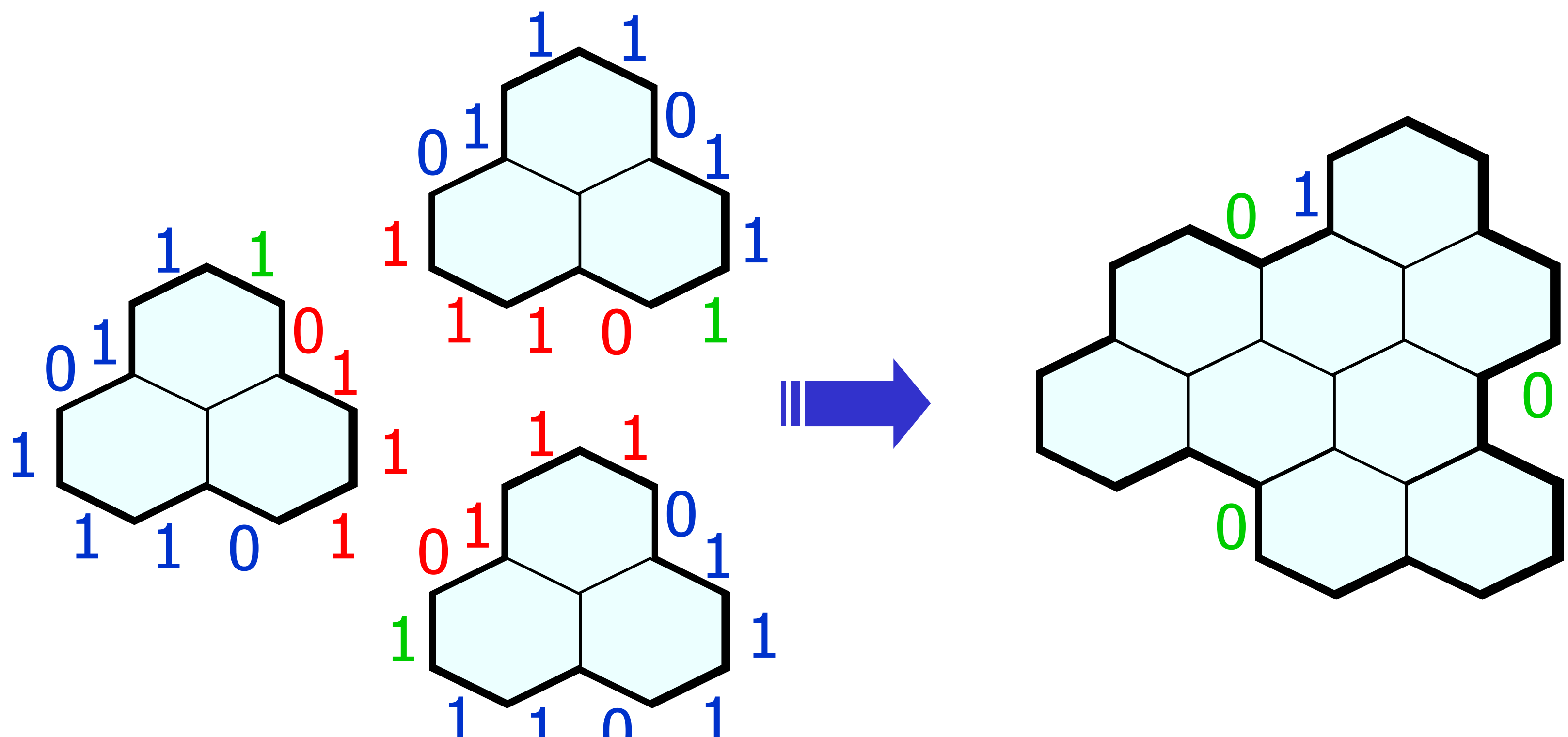
Example of Hexagonal Cell Representation

- A level 1 cell
- Start with b, end at a, we get 101110111011



Example of Cell Merging

- Merging 3 level 1 cells
- 1001101110011011100110 11



Conclusions

- A three way on-chip interconnect architecture, Y-Trees, is proposed.
- Y-Trees have compactable performance with X-Trees under M metrics.
- Y-Trees can grow hierarchically and can cover all the hexagonal cells in the array.

Thanks !

Questions?