*International Symposium on Physical Design 2009*

# Transistor-Level Layout of High-Density Regular Circuits

**Yi-Wei Lin[1], Malgorzata Marek-Sadowska[1] and Wojciech Maly[2]**

*[1]Dept. Of ECE, University of California, Santa Barbara*
*[2]Dept. Of ECE, Carnegie Mellon University*

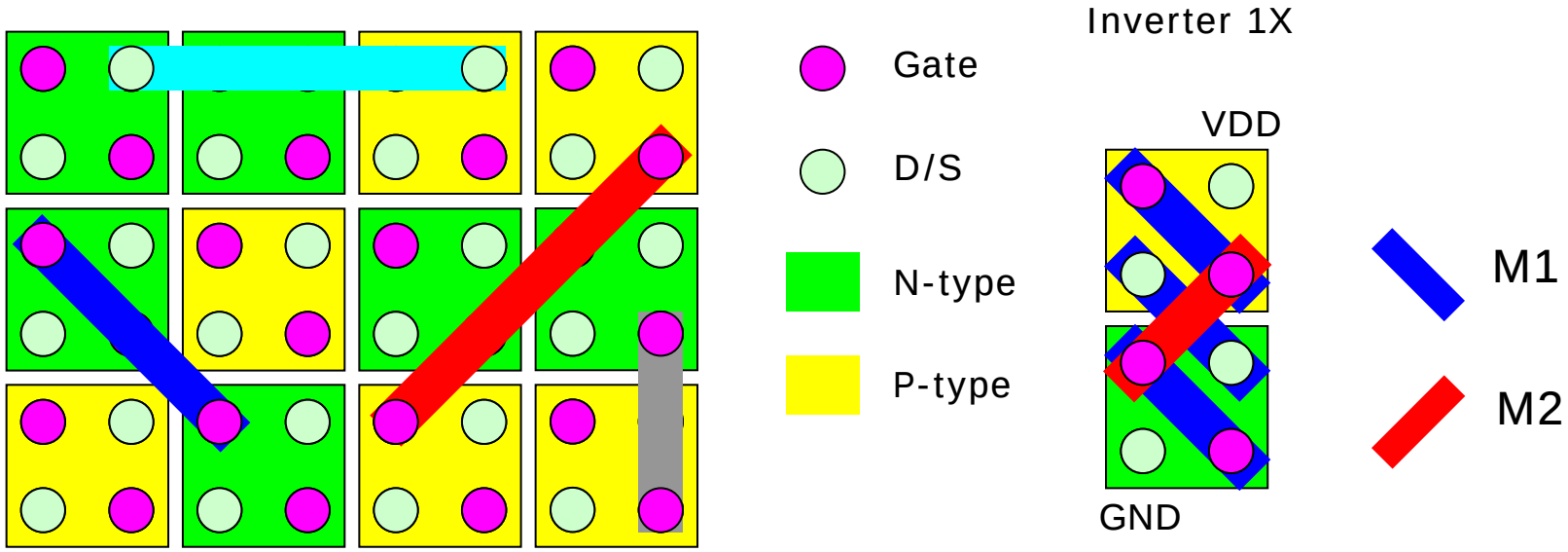
# Outline

- Introduction
- High Density Regular Layout Style
- Problem Formulation
- Analysis of Dense Layout Style
- Transistor-level Placement and Routing
- Experimental Results
- Conclusions

# Introduction

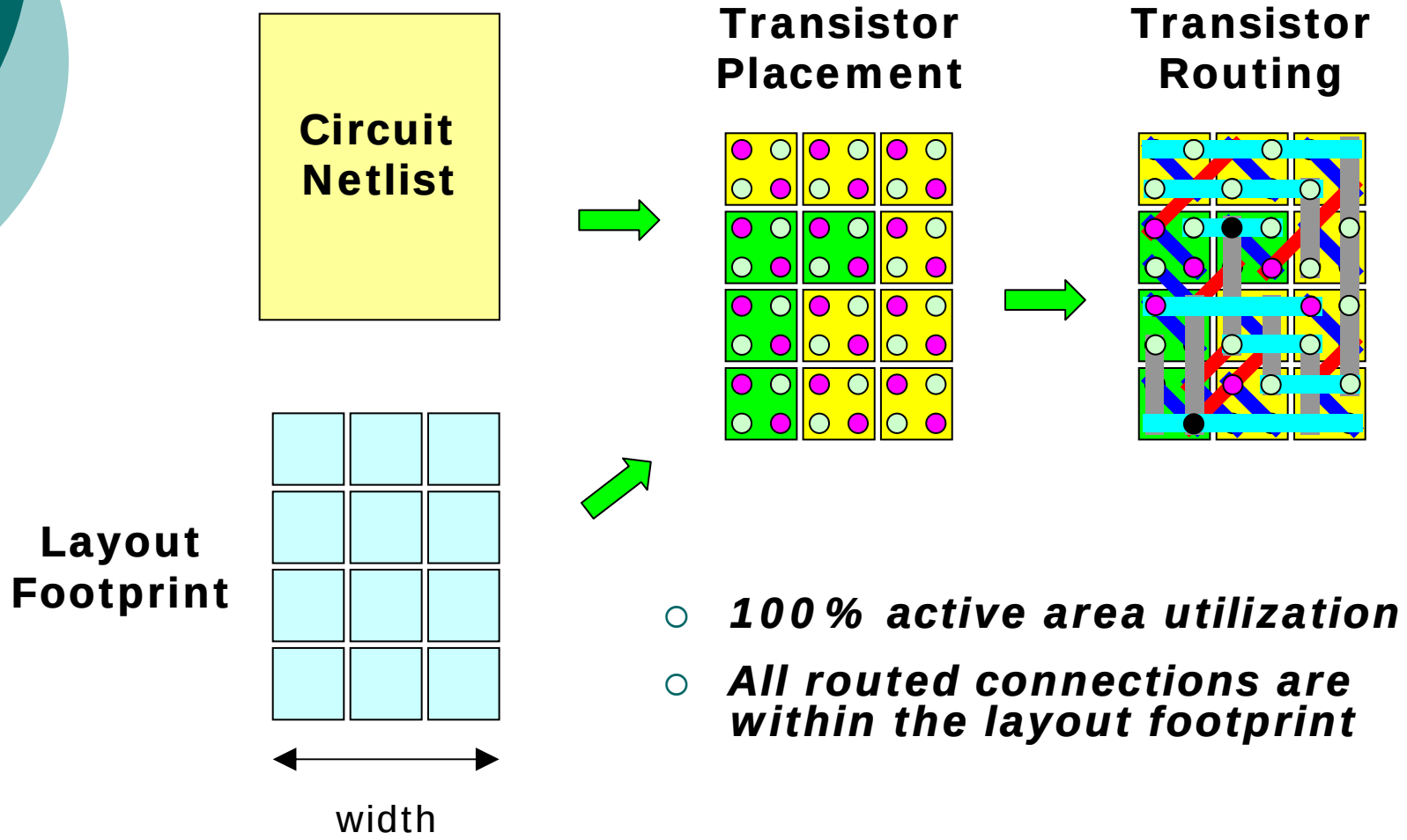
- Modern IC technology experiences manufacturing difficulties
  - Complex interactions between components
  - Difficult to abstract or model
  - Manufacture: lower yield & higher cost
- Regular Fabric
  - Pros:
    - Uniform patterns and similar neighborhoods
    - Interactions between components are easier to model and analyze
  - Cons:
    - Performance and area overhead
    - Layout restrictions

# High Density Transistor Array




- *Transistors are prefabricated in identical size*
  - Transistor sizing needs parallel connected multiple transistors
- *All wires on the same layer are parallel*
  - *Vias needed for turning connections*

# Problem Formulation

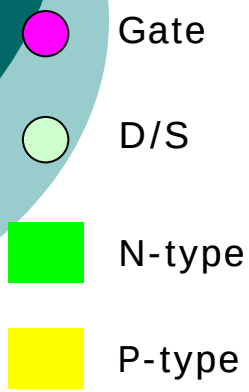


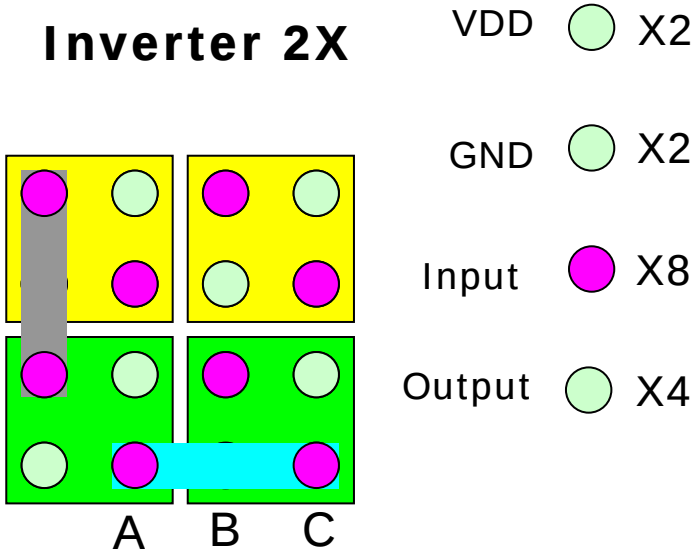

- ***100% active area utilization***
- ***All routed connections are within the layout footprint***

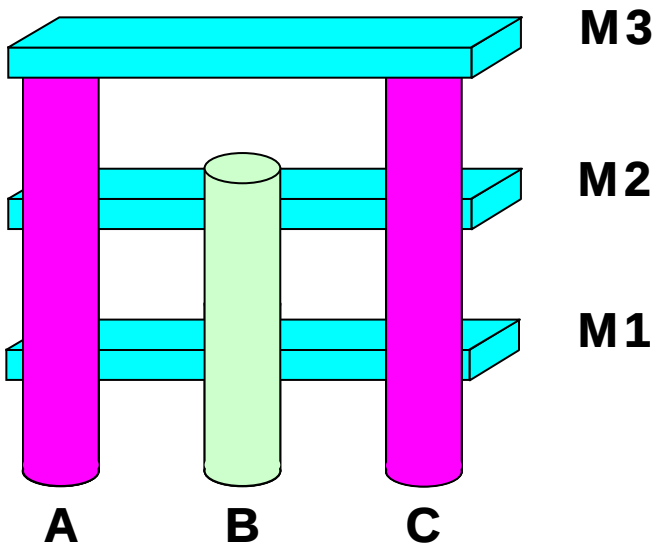
# Placement & Routing Characteristics

- High density layout style
  - No routing space between transistors
- All routing wires are on top of the active device area
  - Each connection affects routability of other nets
- Placement & routing
  - Transistor positions and drain-source assignment are critical
  - Routing criteria
    - Wire length
    - Routing resource congestion
    - ***Pin blocking***

# Pin Blocking




**A pin is covered by connections of other nets at Mk**

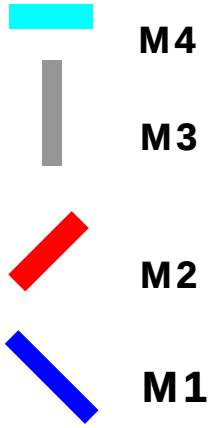
⇨ It has to be connected using M1~Mk-1

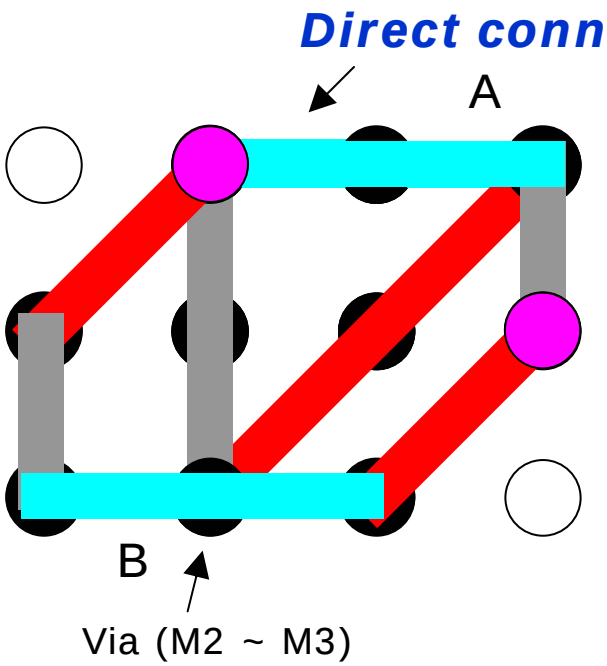
**Pin B cannot be connected to M1**
M1 or M2

***Pin Blocking!***

Metal ordering: M1 M2 M3 M4

# Connection Types




**Adjacent pins**

**Aligned non-adjacent pins**

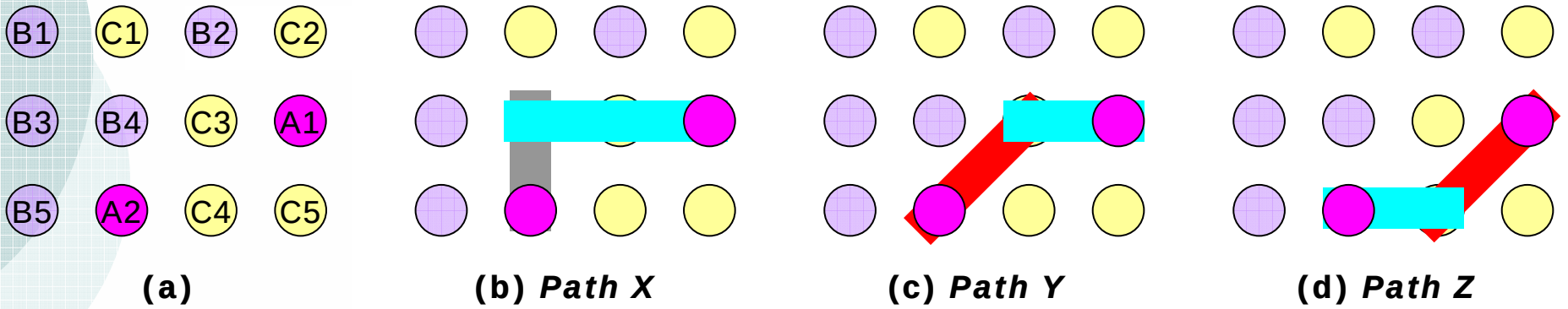
**Staggered non-adjacent pins**

- **To ensure covered (black) pins routable**
  - *Minimize # of covered (black) pins*
  - *Pass through wires on higher metal layers*

# Transistor Routing

- Input:
  *Transistor positions and pin assignment are fixed*
- Objective:
  *Route all nets within the given footprint*
- Greedy route selection
  - Path probability
    - Estimate how neighboring pins are affected by a routed connection
    - For multi-pin nets, use tree topology to adjust path probability
  - Resource congestion
    - Many nets compete for wire segments and vias
- SAT solver
  - Evoked when problem is sufficiently reduced by greedy routing

# Path Weight




**(a)** **(b) *Path X*** **(c) *Path Y*** **(d) *Path Z***

- ***Path:*** a connection between two pins
- ***Route:*** a set of all paths between two pins
- ***NW*** (Number of ways out of the pin):
  at most ***2*** for a layer, at most ***8*** for four layers
- ***Path impact*** (For the covered pins):
  $( NW_{after} / NW_{before} )$
- ***Path Weight:***
  product of path impacts of all covered pins

| **Path (A1→A2)** | *X* | *Y* | *Z* |
|---|---|---|---|
| **Covered pins** | C3, B4 | C3 | C4 |
| **Covered levels** | C3: M4<br>B4: M3 | C3: M2 | C4: M2 |
| **Path impact** | C3: 5/7<br>B4: 3/7 | C3: 2/7 | C4: 0/3 |
| **Path weight** | **0.306** | **0.286** | **0** |
| **Validity** | **O** | **O** | **X** |

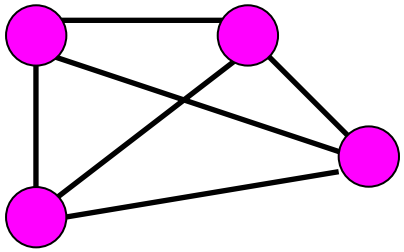
# Path Probability

- Find all paths of a route:
  - Enumerate all *feasible* paths
  - Feasible paths are limited by the via number on a path
  - This number depends on the length of a route
- Path weight:
  - Differentiates between paths of a single route

*2-pin Net*

***# of Route: 1***

*Multi-pin Net (k)*



***# of Route: k(k-1)/2***

Only (k-1) routes need to be connected

- Path probability adjustment:
  - Use a graph to model a multi-pin net
  - A tree represents a possible routing of the net

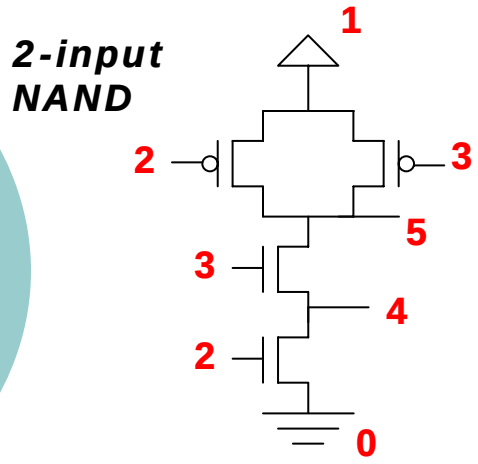
# Transistor Placement

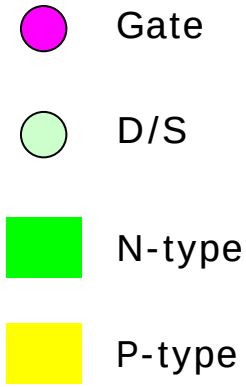
- Objective:

  *Assign transistors to the physical locations within the given footprint and decide their orientations to maximize routability*

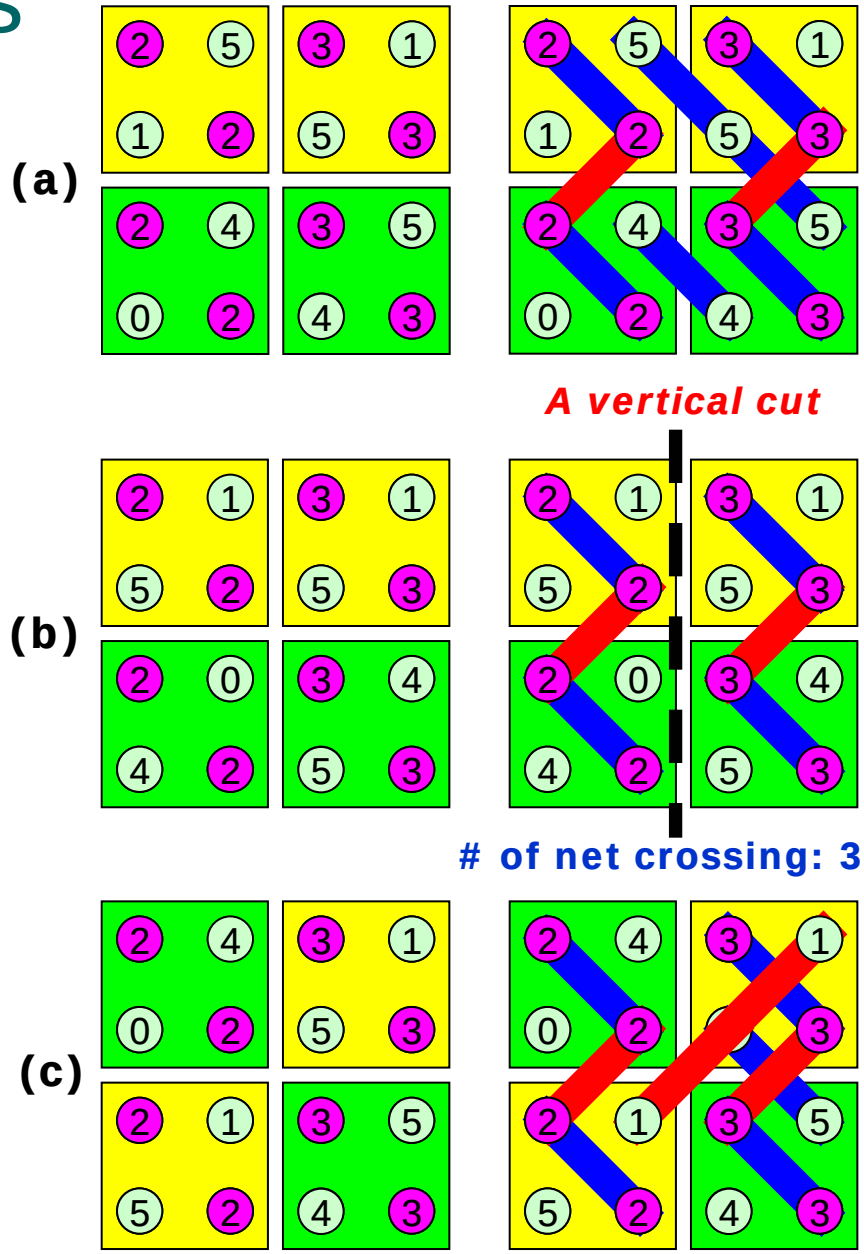
- Placement approach
  - Simulated annealing-based
  - Cost function attempts to
    - Maximize utilization of lower metal layers (diagonal wires)
    - Avoid non-direct connections on higher metal layers (M3 & M4)
    - Capture the expected routing congestion by examining the available wire tracks and the number of nets crossing footprint cuts

# Placement Examples




| Placement | (a) | (b) | (c) |
|---|---|---|---|
| *Total # of routes* | **10** | **10** | **10** |
| *Total # of routes after RDR@M1M2* | **1** | **4** | **2** |
| *Total wire length at M3 & M4* | **4** | **10** | **8** |




***RDR@M1M2*:** routable direct routes at M1 & M2

# Experiments

- Experimented with circuits containing 12~72 transistors
- Placement experiments
  - Compare the routability-driven SA placer (RD-SA) and a bounding-box based SA placer (BB-SA)
  - RD-SA can always produce a routable placement in 1 or 2 attempts
  - BB-SA was run 20 times on each example; success rate 10%~20%
- Routing experiments
  - The greedy algorithm termination criteria

    → The number of disconnected components of a net
  - K-greedy:

    The greedy algorithm stops when all nets consist of at most K disjoint sub-nets
  - 1-greedy:

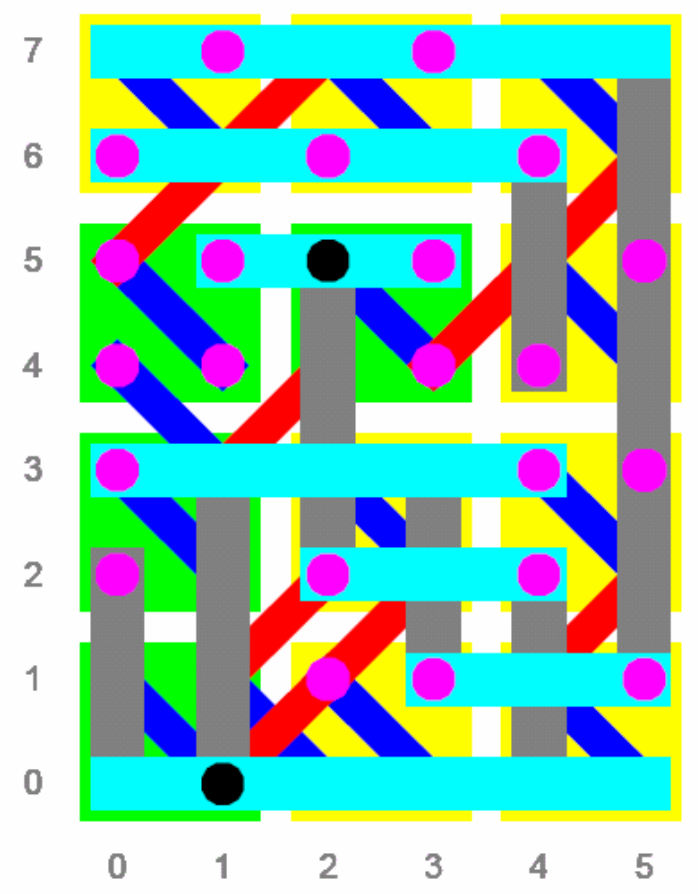
    The greedy algorithm attempts to complete the routing; SAT solver is not evoked.

# Placement Experimental Results

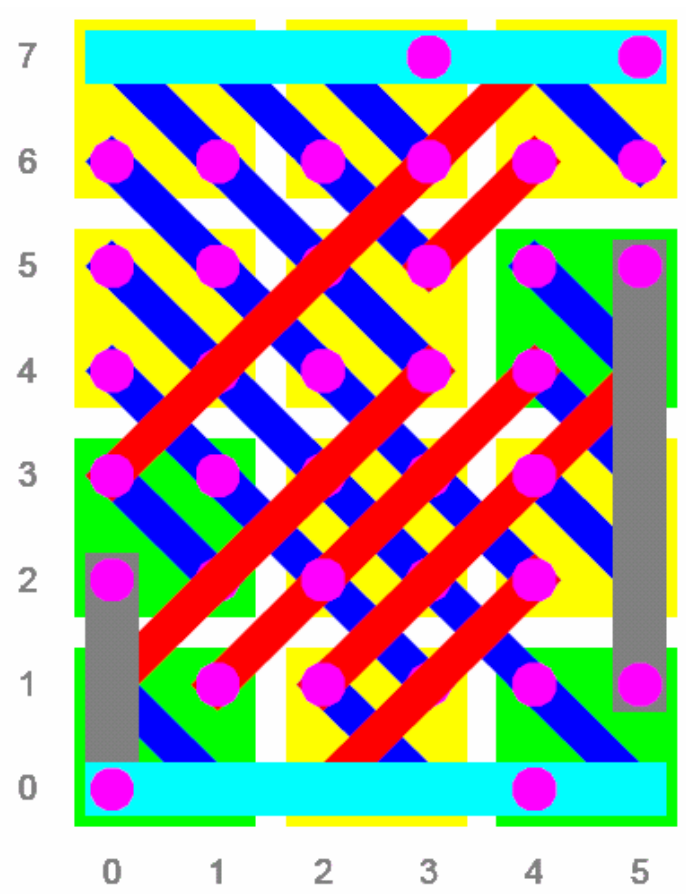
| Circuit | # of Tran. | Ft. Width | RD-SA | | | Ratio(BB-SA/RD-SA) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Length | M3M4 Depth | HL | Bound. Box | Length | M3M4 Depth | HL |
| **#1** | 12 | 4 | 90 | 72 | 16 | *0.59* | *1.54* | *0.33* | *2.75* |
| **#2** | 24 | 4 | 237 | 109 | 61 | *0.88* | *1.20* | *0.52* | *1.66* |
| **#3** | 36 | 4 | 342 | 134 | 90 | *0.94* | *1.10* | *0.83* | *1.14* |
| **#4** | 28 | 4 | 319 | 84 | 96 | - | - | - | - |
| **#5** | 42 | 6 | 455 | 109 | 135 | *0.91* | *1.08* | *0.79* | *1.13* |
| **Macro #1** | 54 | 6 | 502 | 216 | 105 | *0.90* | *1.16* | *0.71* | *1.25* |
| | | 7 | 515 | 204 | 124 | - | - | - | - |
| **Macro #2** | 60 | 6 | 667 | 152 | 212 | - | - | - | - |
| | | 8 | 708 | 151 | 225 | - | - | - | - |
| **Macro #3** | 72 | 6 | 847 | 128 | 278 | *0.82* | *1.13* | *0.77* | *1.17* |
| | | 8 | 770 | 197 | 213 | *0.86* | *1.22* | *0.69* | *1.36* |
| **AVG** | | | | | | ***0.84*** | ***1.20*** | ***0.66*** | ***1.49*** |

*Ckt #1: Sized AOI 211 cell; Ckt #2: Sized OAOI211 cell; Ckt #3: Sized OAI3111 cell; Ckt #4: Static FA; Ckt #5: Static 2-bit adder*

# AOI211 Placement Results

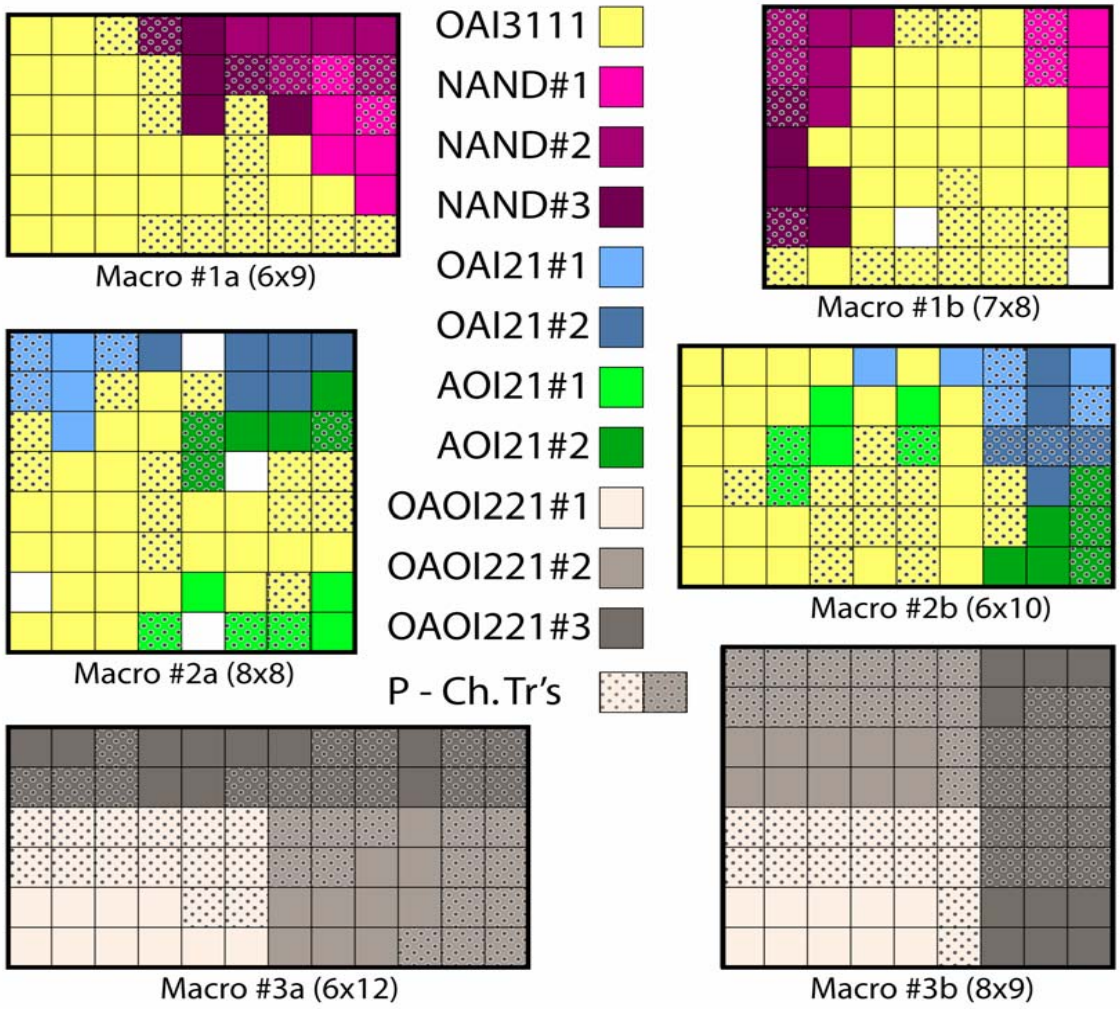



BB-SA




RD-SA

# Macro Placement Results



# Routing Experimental Results

| *Circuits* | *Stops* | *Length* | *M3 Av.* | *M4 Av.* | *HL* | *Time (Sec.)* |
|---|---|---|---|---|---|---|
| **Sized AOI211 (12 transistor)** | 3-greedy | 90 | 2 | 6 | 16 | 5.57 |
| | 2-greedy | 91 | 2 | 6 | 18 | 2.45 |
| | 1-greedy | 92 | 1 | 6 | 22 | <1 |
| **Sized OAOI221 (24 transistor)** | 3-greedy | 237 | 2 | 3 | 61 | 2335 |
| | 2-greedy | 238 | 2 | 3 | 63 | 12.54 |
| | 1-greedy | 241 | 0 | 3 | 67 | 6.61 |
| **Sized OAI3111 (36 transistor)** | 3-greedy | 342 | 1 | 2 | 90 | 7894 |
| | 2-greedy | 348 | 1 | 2 | 96 | 31.45 |
| | 1-greedy | - | - | - | - | - |
| **Macro #1 (54 transistor)** | 3-greedy | 502 | 2 | 3 | 105 | 12543 |
| | 2-greedy | 511 | 2 | 2 | 111 | 354.12 |
| | 1-greedy | - | - | - | - | - |

# Conclusions

- Analyzed characteristics of a high-density super-regular layout style.
- Proposed automatic transistor place and route algorithm.
  - Obtained high-quality P&R outcome for most test cases.
  - Explored new design flexibilities and opportunities offered by the regular fabrics.

# Q & A

**Thank you!**