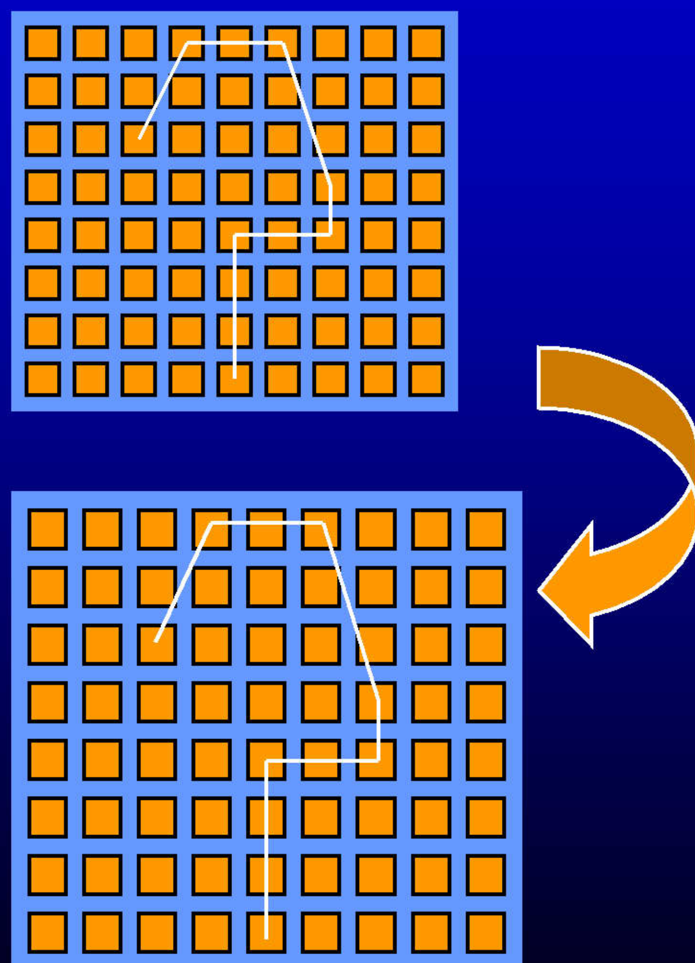
# Exhibit F
# (7 of 12)

Timing closure

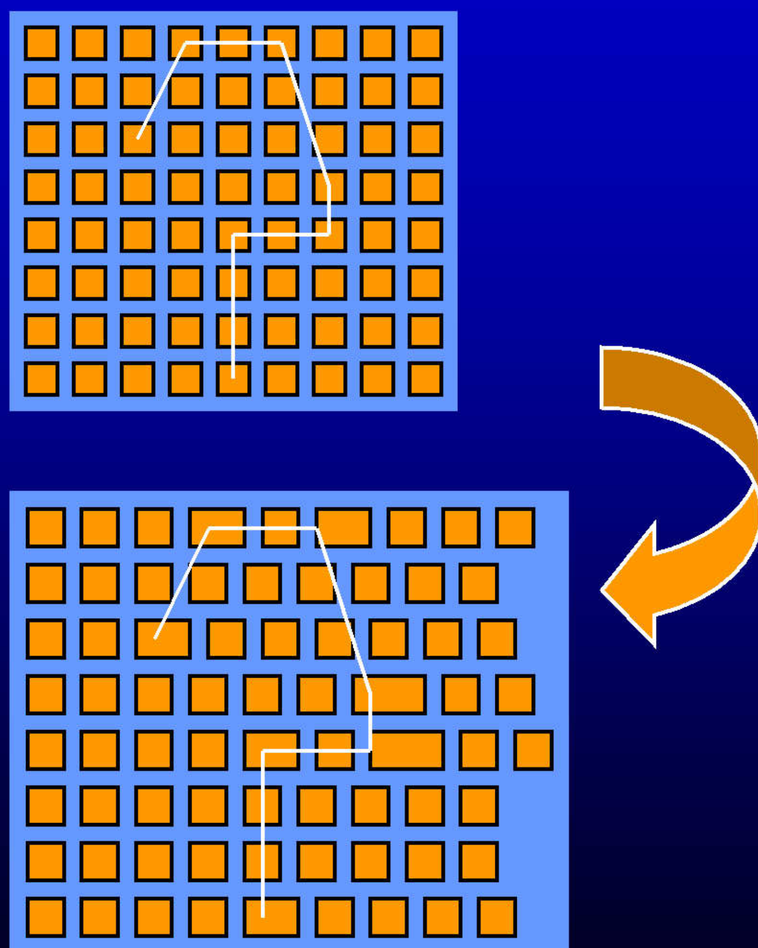
# Conventional way: Worst case delay sets timing



- 99% of paths meets timing, 1% does not
- Cell sizes do not change during Place and Route
- Design conservatively to avoid excessive iterations. Also WLM is tuned conservatively.
- This **oversizes** all cells
  - because also cells on non-critical paths are sized up.
- Chip significantly bigger than necessary (10-30%)

**size + parasitics = timing**

# What about In-place optimization?



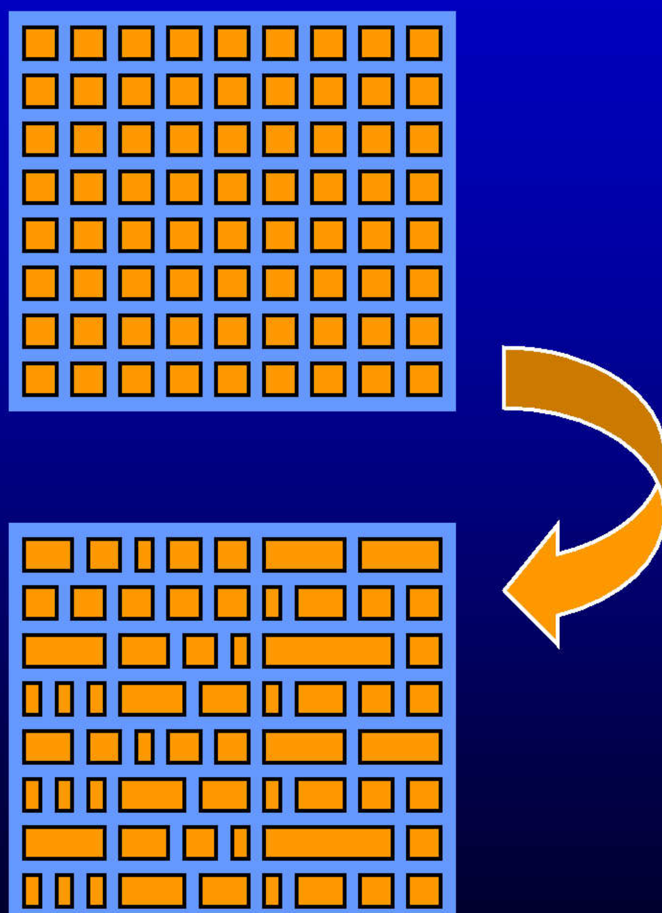
- Do a post-placement ECO,
- Change only the cells on the critical paths.
- Conservatism is still required because of limited ECO capacity.
- All non-critical cells are still oversized
- Chip still bigger than necessary.

size + parasitics = timing

Timing closure

Timing closure

# Gain based synthesis: area is additive



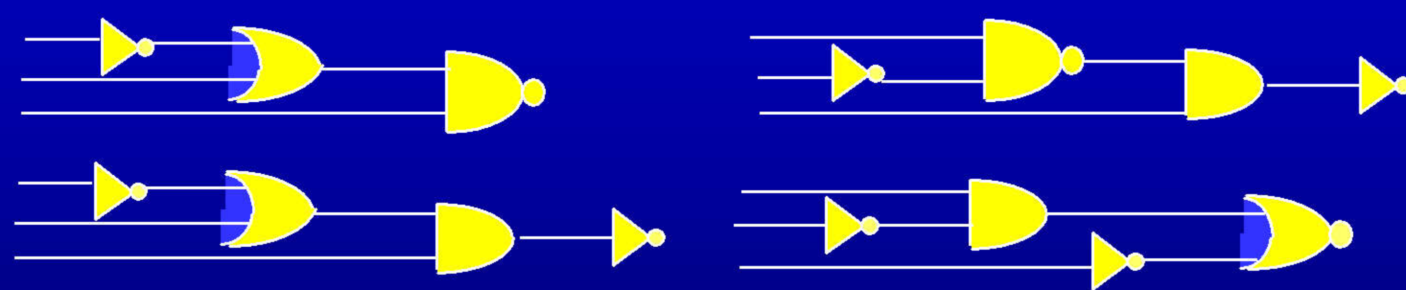
- Timing is fixed,
- As a result, cell sizes change.
- But large cells and small cells cancel out: some get bigger, others smaller
- All cells have exactly the right drive strength: many paths are almost critical.
- Chip size remains small (10-30% smaller than conventional way)

timing + parasitics = size

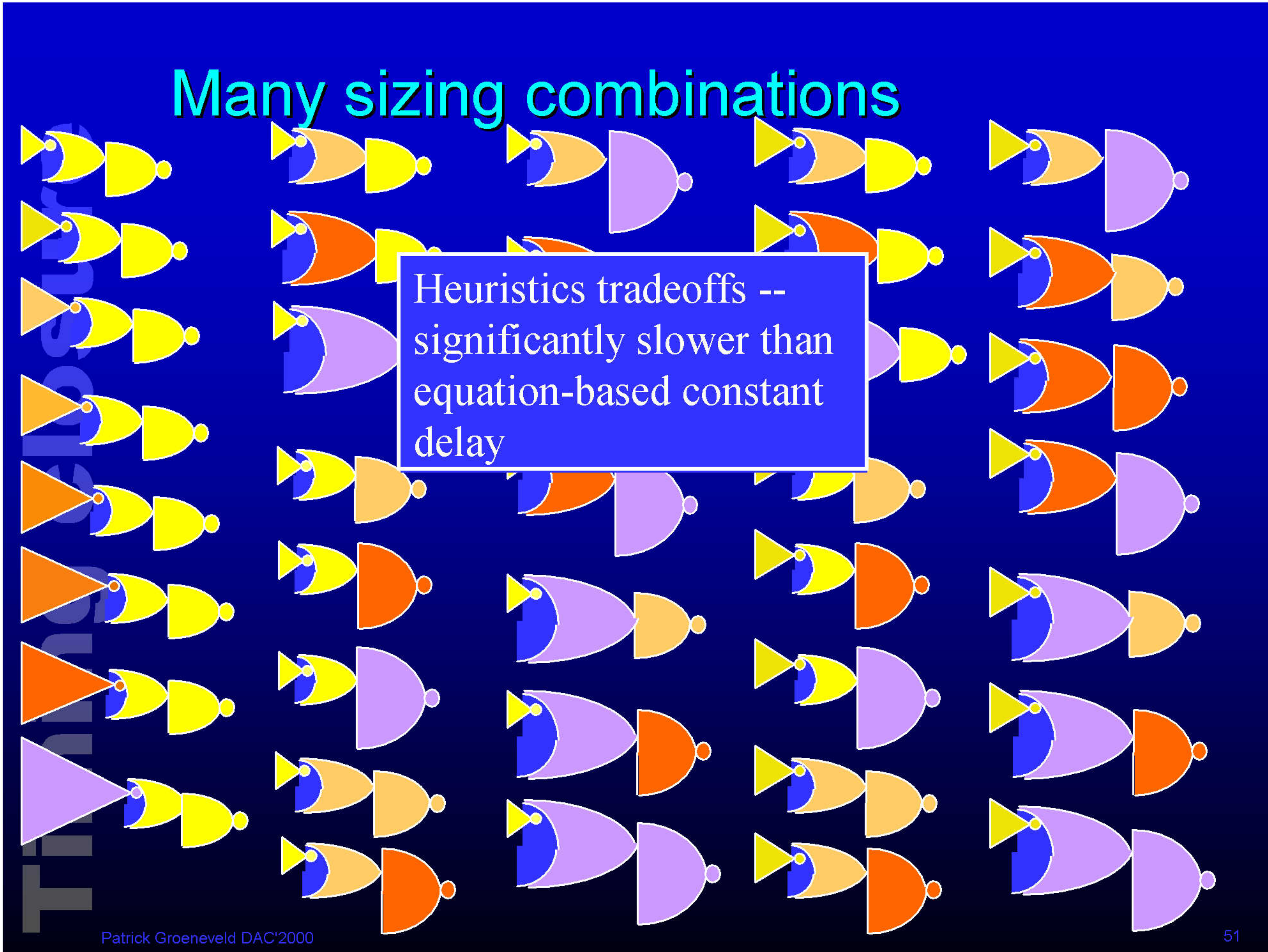
Timing closure

# Logic (wireload) Synthesis

- For a simple function ( (A’ + B) * C ) `
- Various logic structures are possible with one size



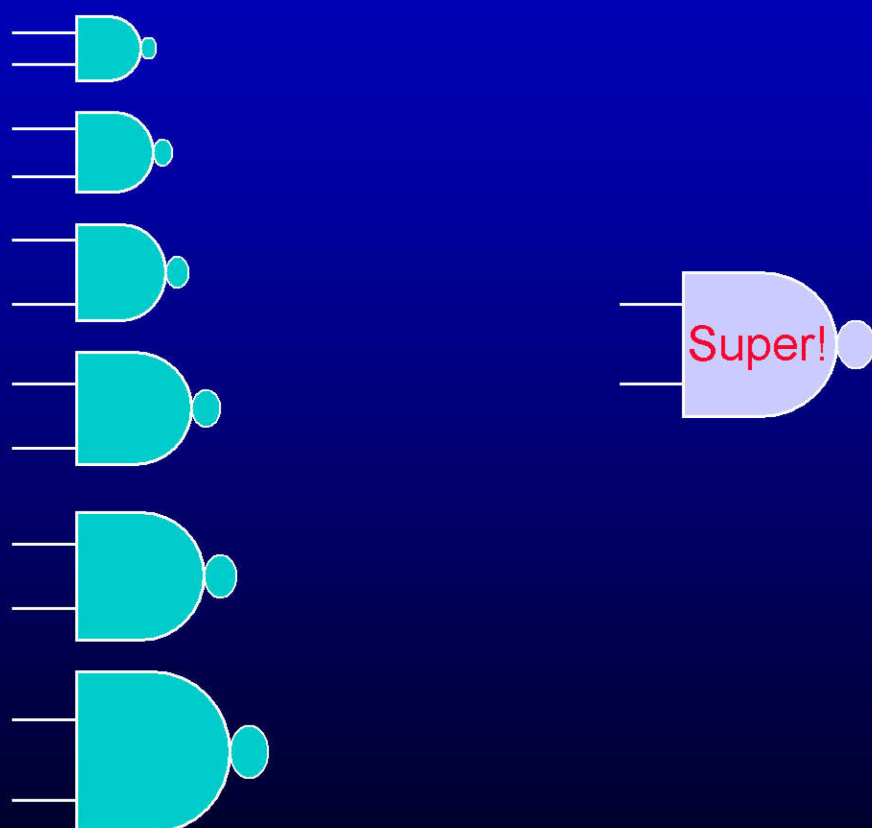
- Conventional logic synthesis tool attempts to optimize the delay by:
  - Logic restructuring
  - Picking the proper sizes
- This is driven by a vague idea of the wire load




MAGMA0791084

# Gain-based synthesis: supercells

- Need a single ‘super’ cell representing all sizes in a logic function.




- Contains:
  - g, h, p
  - size-range

# Gain-based mapping

- In timing-critical parts, the mapper picks super cells that have low parasitic delay and highest maximum drive strength.
- In non-critical parts, ‘weaker’ super cells can be used.
  - Pick cells that have potentially the smallest size.
- Insert buffers on high-fanout nets

# Putting it together

- Map onto generic ‘super cells’ with flexible area.
- Optimize gains for all super cells such that maximum speed is achieved. ***This fixes all delays in the circuit!***
- *Give up* if the (optimally conditioned) circuit does not meet the given timing criteria.
- Perform ‘sizing driven placement’: keep delay constant by adapting cell size to parasitic capacitance of the wires. Parasitic wire delay is based on coarse routing of the wires.
- Fix remaining timing problems through buffering, cloning, restructuring.
- Update floor plan if the timing is still not met.
- For each supercell, pick the one standard cell that matches the required drive strength.
- Legalize the placement (a.k.a detailed placement)
- Perform final routing under delay constraints.