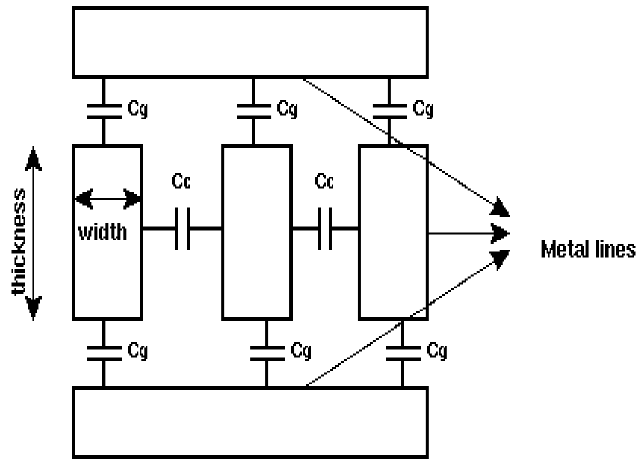
EARLY PROBABILISTIC NOISE ESTIMATION FOR CAPACITIVELY COUPLED INTERCONNECTS

Murat Becer, Rajendran Panda
Advanced Tools Group, Motorola, Austin, TX
David Blaauw
Univ. of Michigan, Ann Arbor, MI
Ibrahim Hajj
American University of Beirut, Lebanon

Introduction

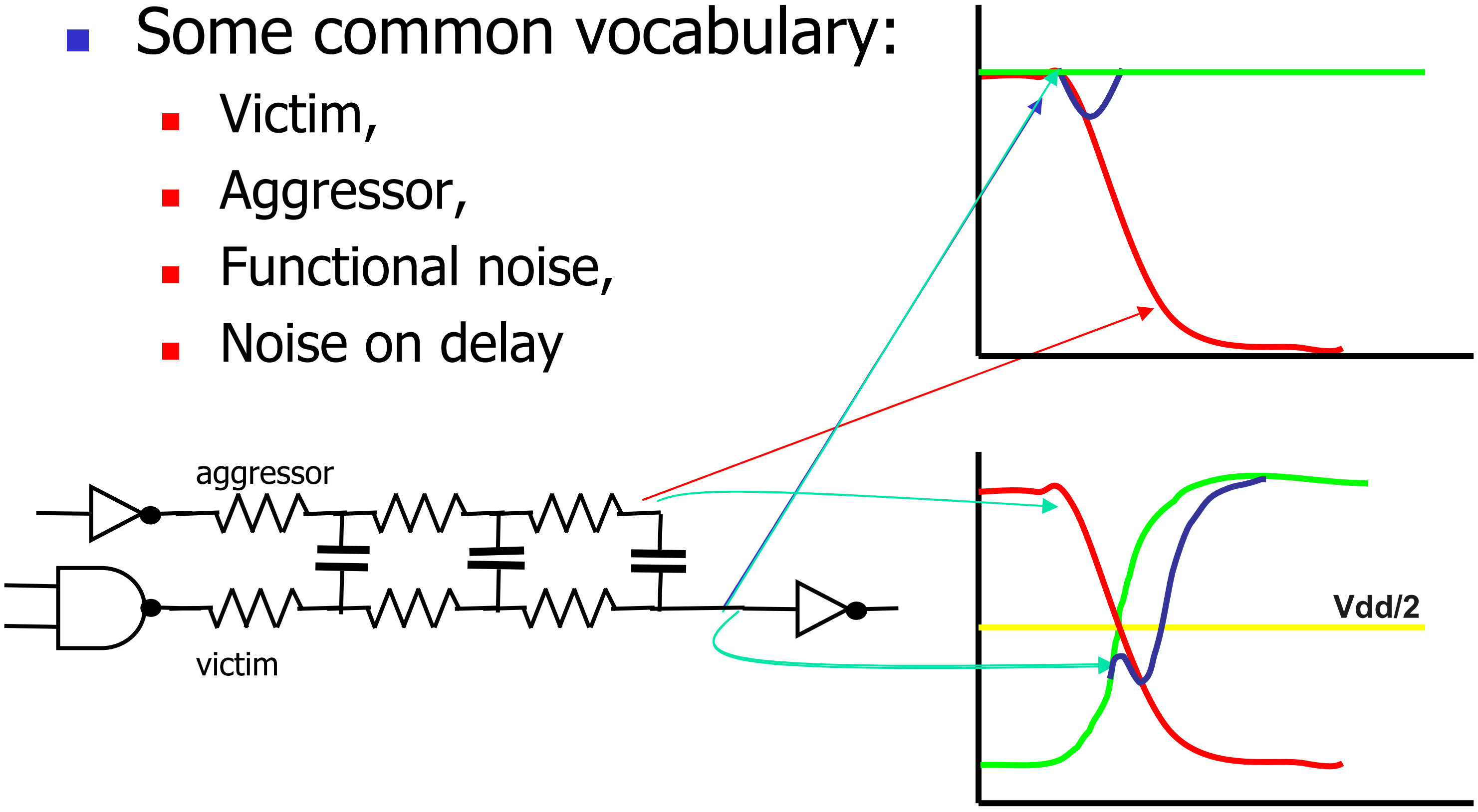
- Ratio of crosstalk capacitance to total capacitance is increasing.
- More performance aggressive circuit structures compromising noise immunity are being used.



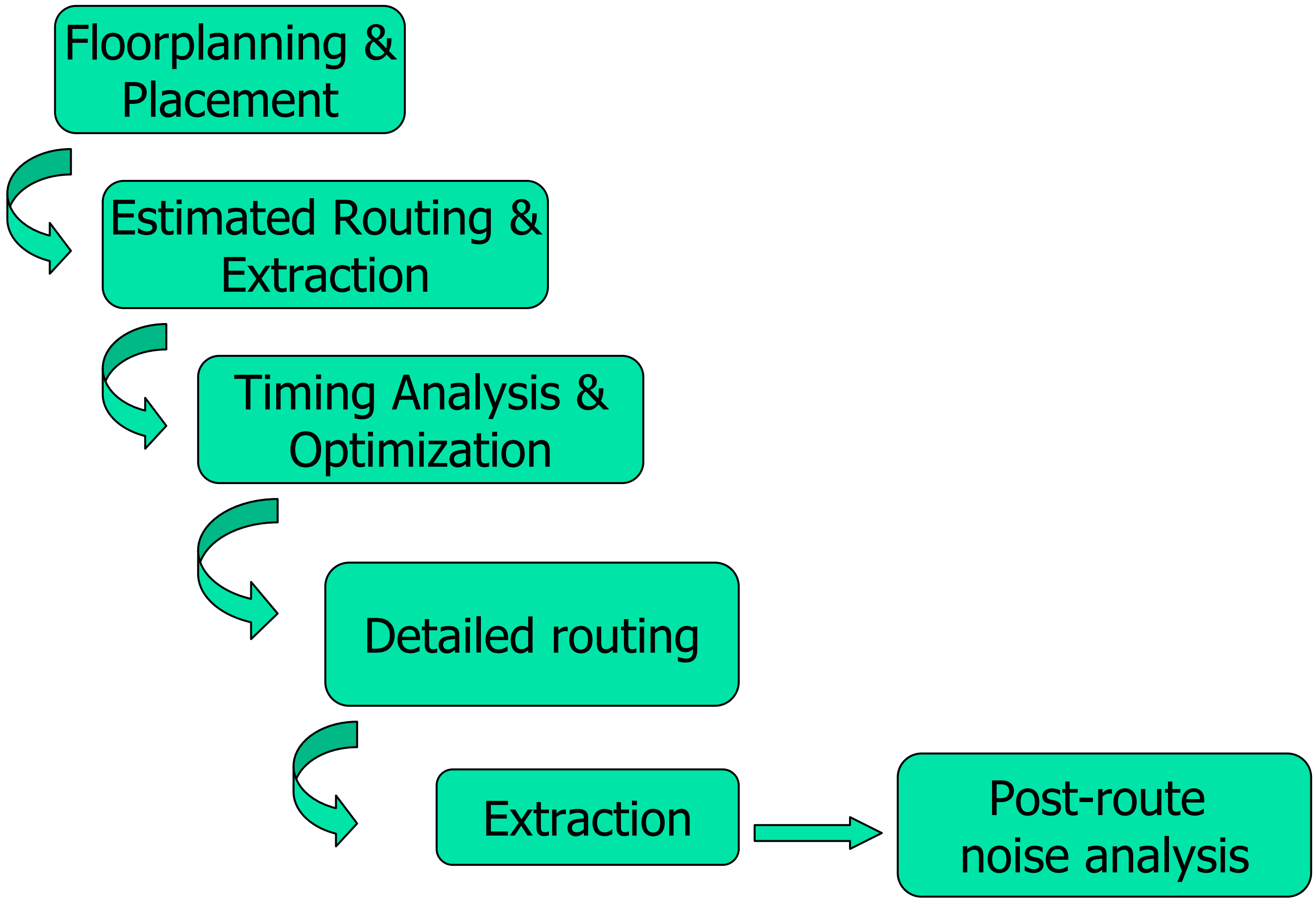
- Noise failures have become a significant design and verification issue for large and high performance designs.

Motivation

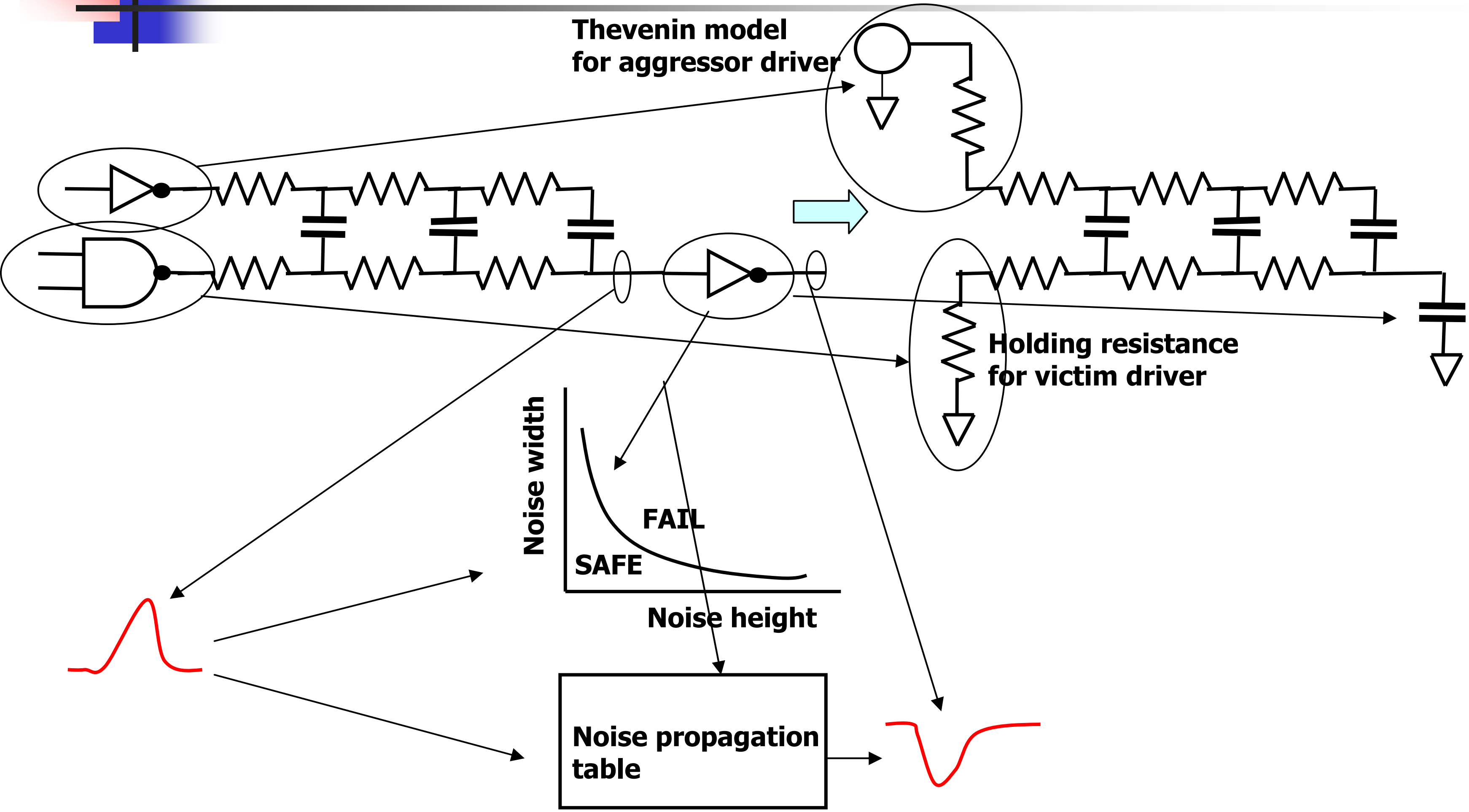
- Some common vocabulary:
  - Victim,
  - Aggressor,
  - Functional noise,
  - Noise on delay



Noise in Current Design Cycle



Post-route Noise Analysis



Post-route Noise Analysis: Too Late!!

- Number of failing nets reach several thousands!!
- Flexibility to change the design and fix these problems is greatly reduced.
- Driver sizing, wire spacing, buffer insertion etc. are difficult to apply at this stage and will require that the entire design be re-legalized and re-routed.
- This can give rise to new noise failures on previously stable nets.
- This can lead to convergence problems and lengthen the design cycle.

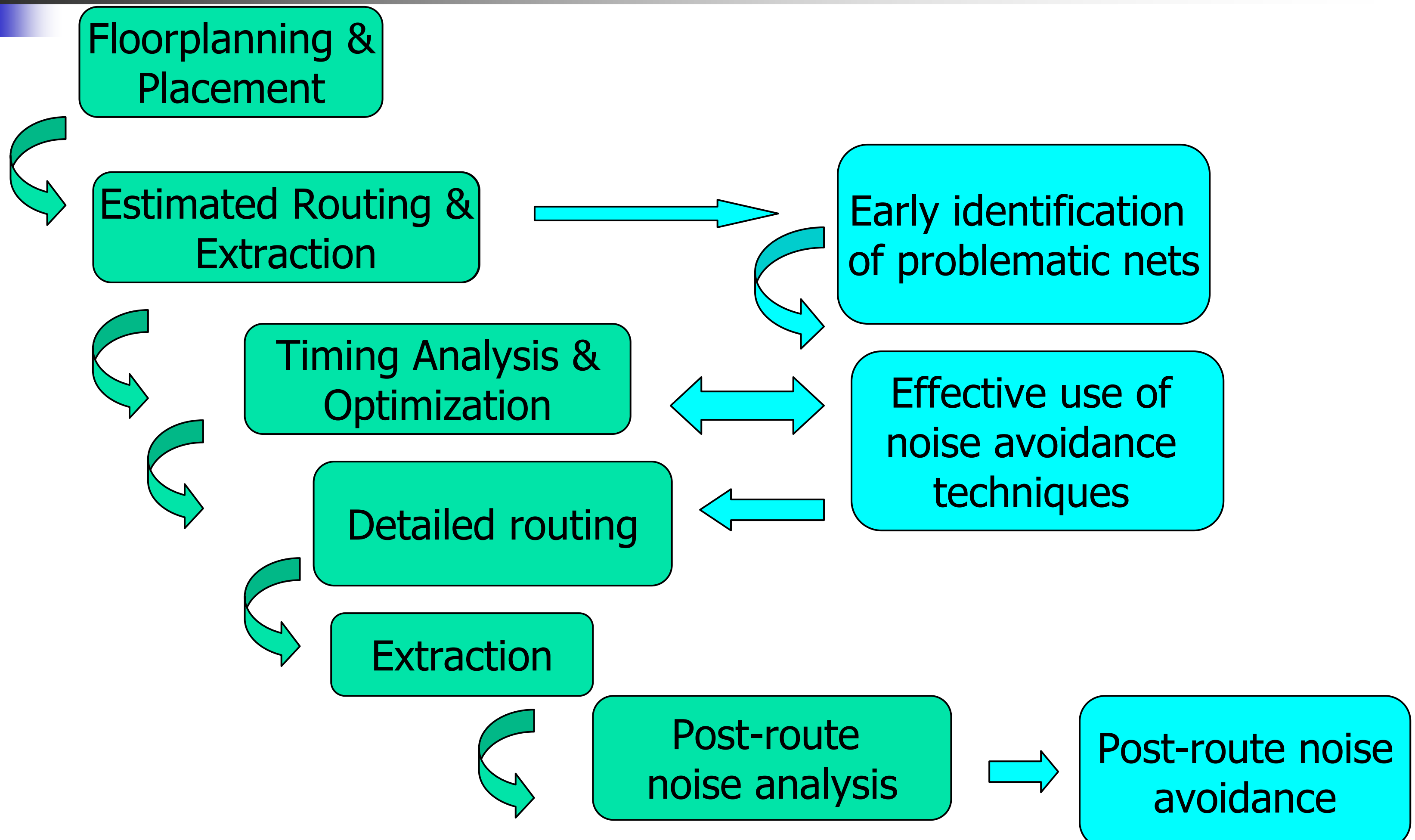
What needs to be done

- Noise awareness should be incorporated into the design cycle as early as possible.
- Methods have been proposed to solve noise problems during detailed routing.
- These methods utilize a limited set of noise avoidance methods: wire sizing and spacing.
- They use approximate noise models (e.g. only based on common parallel length of neighboring nets) for performance reasons.
- Detailed routing is already very complex. Such an approach will over-constrain an already hard problem.

What needs to be done

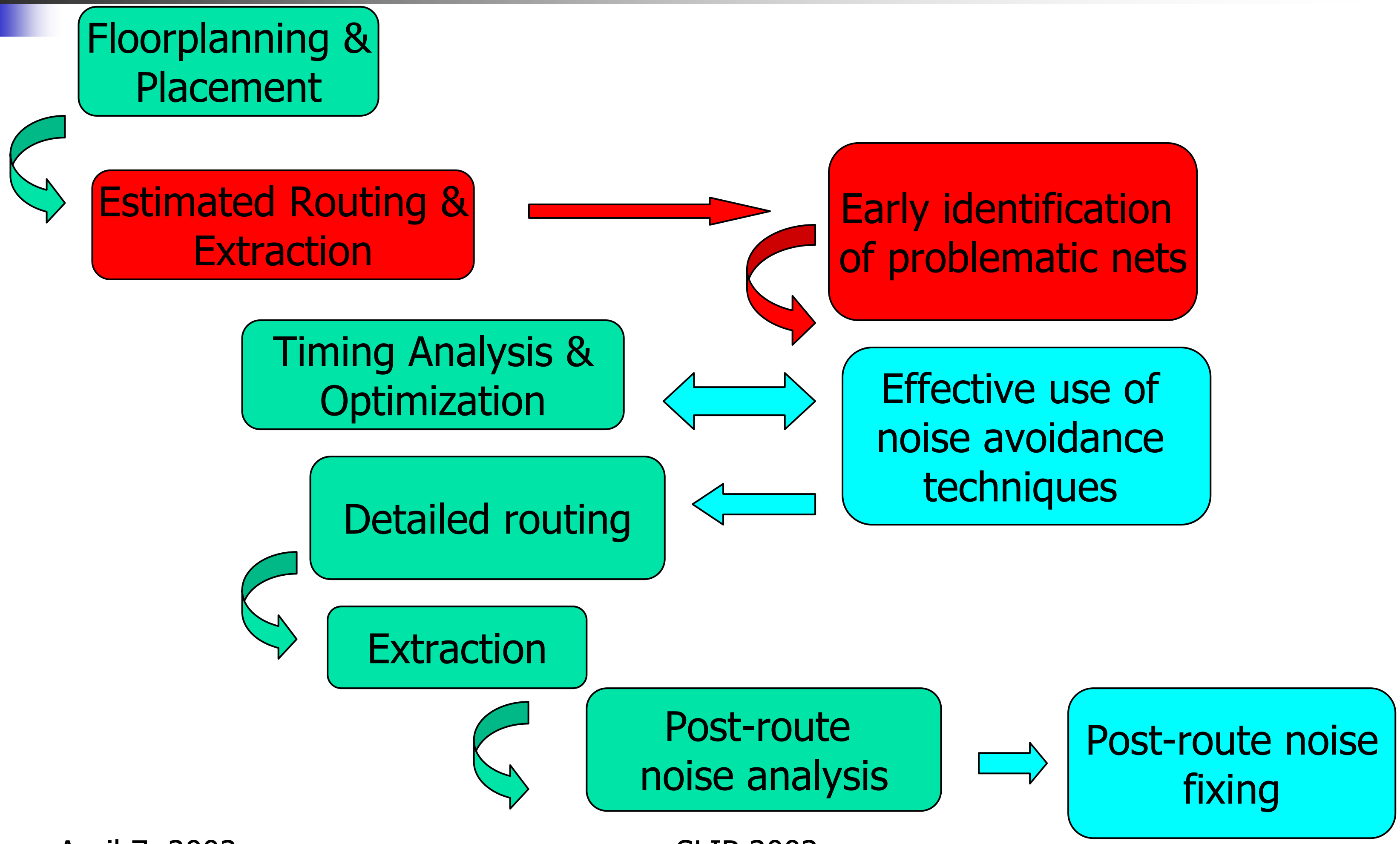
- Methods to identify problematic nets in an earlier design stage, before detailed routing, where flexibility is abundant.

Noise Aware Design Cycle



Blocks and connections addressed in this paper

Pre-route noise estimation



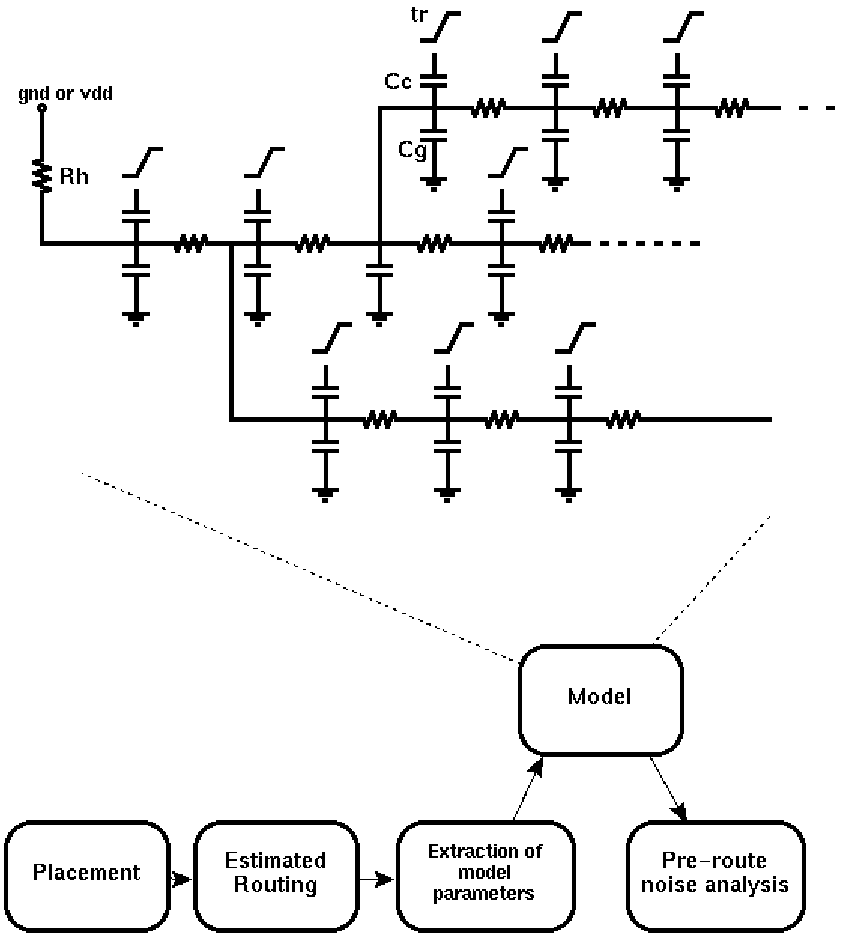
Pre-route noise estimation

- Goal: To identify problematic nets as early as possible

Pre-route Noise Estimation

- Although much flexibility exists to fix noise at this stage, little information is available on which nets are likely to fail.
- Exact wire length, wire topology, relative positioning of wires are not available.
- To be able to perform accurate pre-route noise estimation, need to estimate accurately:
  - distributed interconnect characteristics of a net,
  - its coupling capacitance to neighboring nets,
  - drive strength of its neighbors

Noise Estimation Methodology



Estimated routing

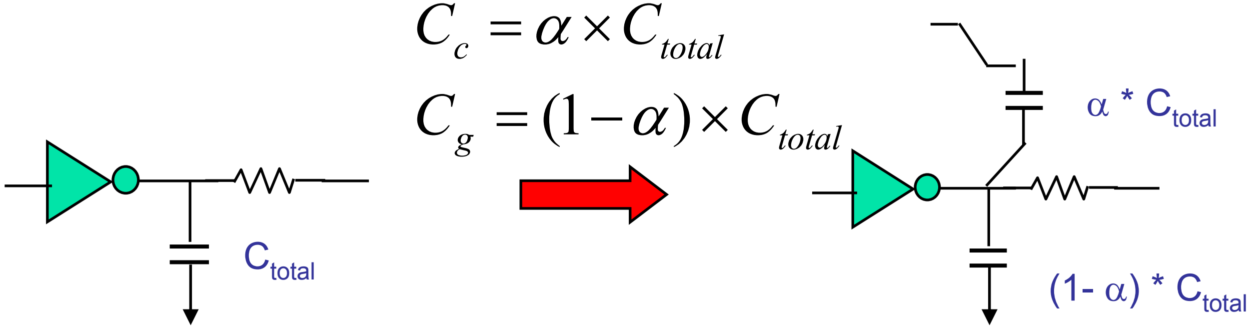
- An estimated routing is required to be able to extract the required parameters.
- Simplest form: Steiner tree routers.
  - Congestion is not taken into account,
  - Multiple nets can be assigned to a single track,
  - No information regarding proximity and identity of neighboring nets,
  - An estimate of length and topology of a net can be obtained.

Estimated Routing

- From this information, grounded RC tree representation of the net can be constructed.
- To estimate coupling capacitance:

$$C_c = \alpha \times C_{total}$$

$$C_g = (1-\alpha) \times C_{total}$$



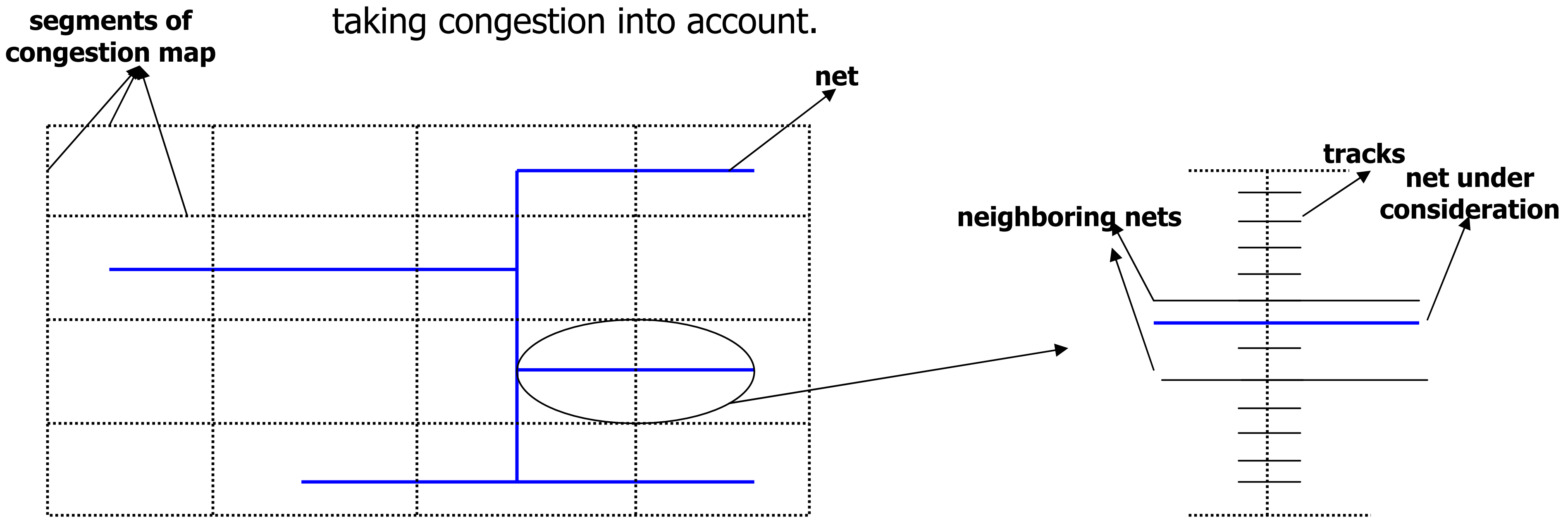
- Transition time t_r can be estimated conservatively based on the speed of the design.

Estimated Routing

- Due to the crude nature of these estimations, significant discrepancy between resulting estimated noise analysis and detailed noise analysis after routing:
  - False failures: Nets that are erroneously identified as failing in estimated noise analysis. They require unnecessary allocation of resources to fix them.
  - Missed failures: Nets that are erroneously identified as non-failing in estimated noise analysis. They will need to be in post-route stage.
- Goal: Minimize false failures and missed failures.

Estimated Global Routing

- Estimated global routers take congestion into account
  - Divides the design into cells,
  - Assigns the number of available tracks for each cell,
  - Connects the instance pins of a net utilizing available tracks of cells while taking congestion into account.

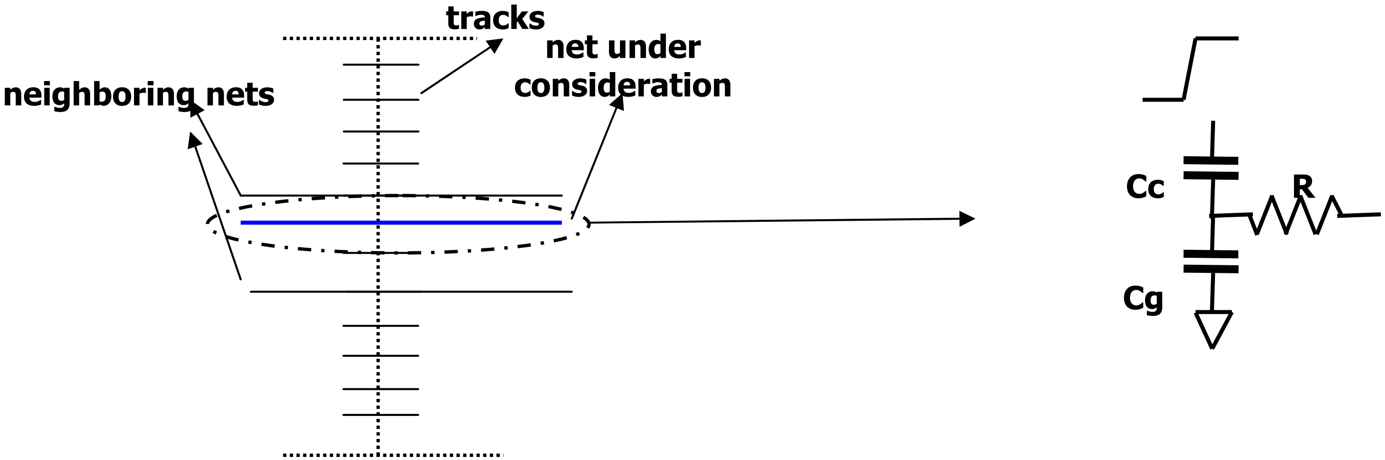


Estimated Global Routing

- Congestion information is given on a segment by segment basis. For each segment i:
  - total number of tracks: n_i
  - number of tracks used by global router: k_i
  - set of nets assigned to that segment are available.
- Note that there is no information on which particular track within the segment, a given net is using, i.e. nets are not ordered thus exact neighbors are still unknown.

Congestion Based Parameter Extraction

- We propose using the congestion map information to extract interconnect parameters such as resistance and ground capacitance as well as coupling capacitance and aggressor information for each net



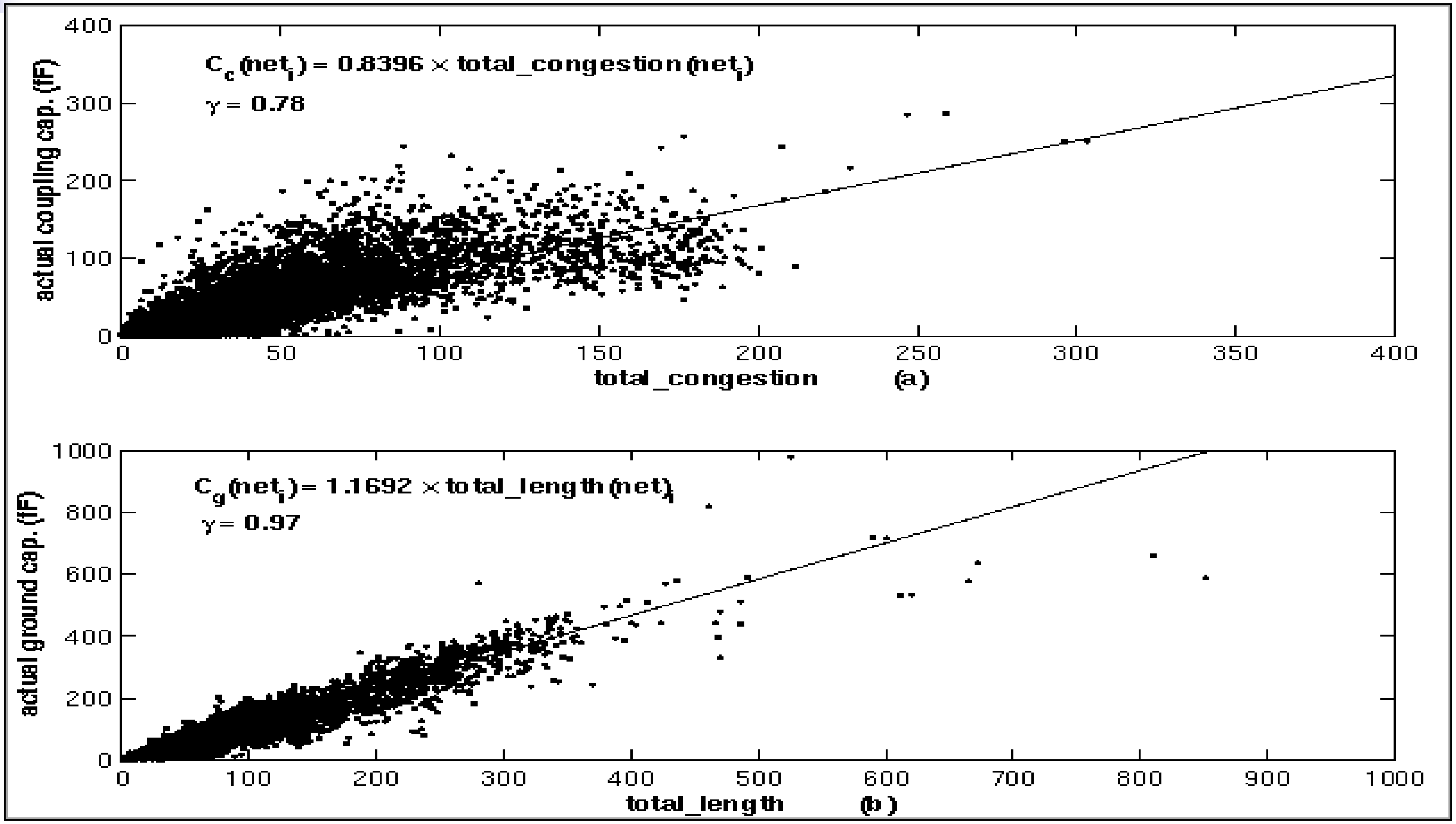
Correlations

- Length and congestion of nets resulting from estimated global routing are typically consistent with those after detailed routing.
- Look at some correlations to verify this on a 0.18μ high performance microprocessor (~58000 nets)
  - Estimated total congestion of a net vs. actual extracted coupling capacitance of a net,
  - Estimated total length of a net vs. actual extracted ground capacitance of a net

$$\text{total length} = \#\text{ of segments a net goes through}$$

$$\text{total congestion} = \sum_{i=1}^{\text{segments(net)}} \frac{k_i}{n_i}$$

Correlations



Calibration Method

- We propose to use correlation data for a previous completed design (possibly of an older technology) to estimate parameters of a new design.

Calculate coefficients for the older technology, t_{old}:

$$K_c(t_i) = C_c(t_{old}) / total\_congestion$$
$$K_g(t_i) = C_g(t_{old}) / total\_length$$

Map coefficients to the new technology, t_{new}:

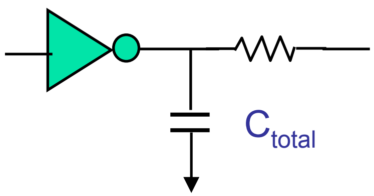
$$K_c(t_{new}) = \frac{C_{c\ unit}(t_{new})}{C_{c\ unit}(t_{old})} K_c(t_{old})$$
$$K_g(t_{new}) = \frac{C_{g\ unit}(t_{new})}{C_{g\ unit}(t_{old})} K_g(t_{old})$$

Estimate caps for each net:

$$C_c(t_{new}) = K_c(t_{new}) \times total\_congestion$$
$$C_g(t_{new}) = K_g(t_{new}) \times total\_length$$

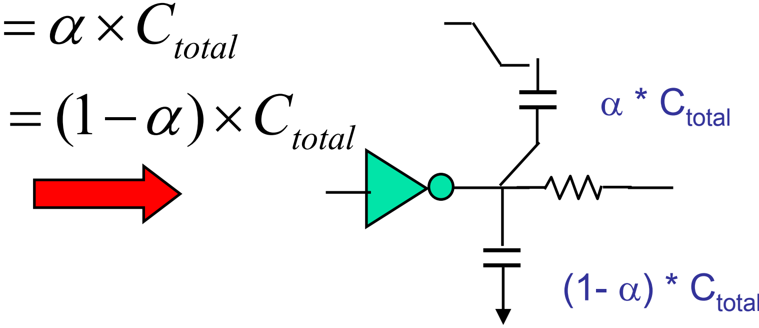
Calibration Method

- Can estimate total coupling and ground capacitance of a net using congestion map information.



$$C_c = \alpha \times C_{total}$$

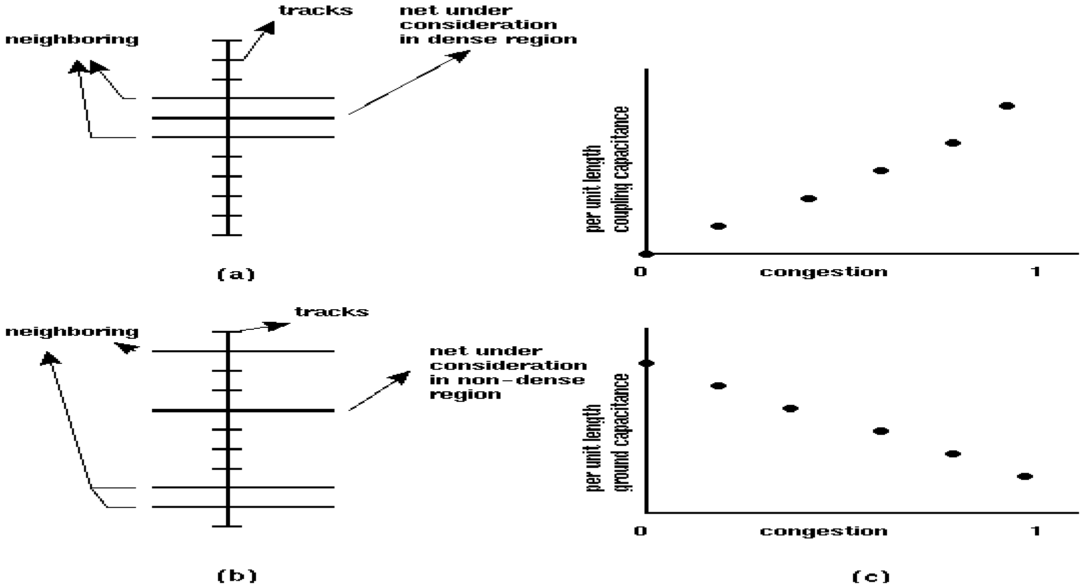
$$C_g = (1-\alpha) \times C_{total}$$



- A different α and total capacitance for <u>each net</u>.
- Total coupling and ground caps are distributed equally for each segment of the congestion map that the net traverses -> distributed RC netlist for each net.
- Drawback: Congestion is taken into account for a net as a whole. Resulting RC netlist will have same α ratio for all segments.

Probabilistic Extraction

- Probabilistic estimation of coupling and ground capacitances using congestion information for each segment that a net traverses.
- How?: Per unit coupling and ground caps for a particular interconnect technology are characterized for a number of density configurations:



Probabilistic Extraction

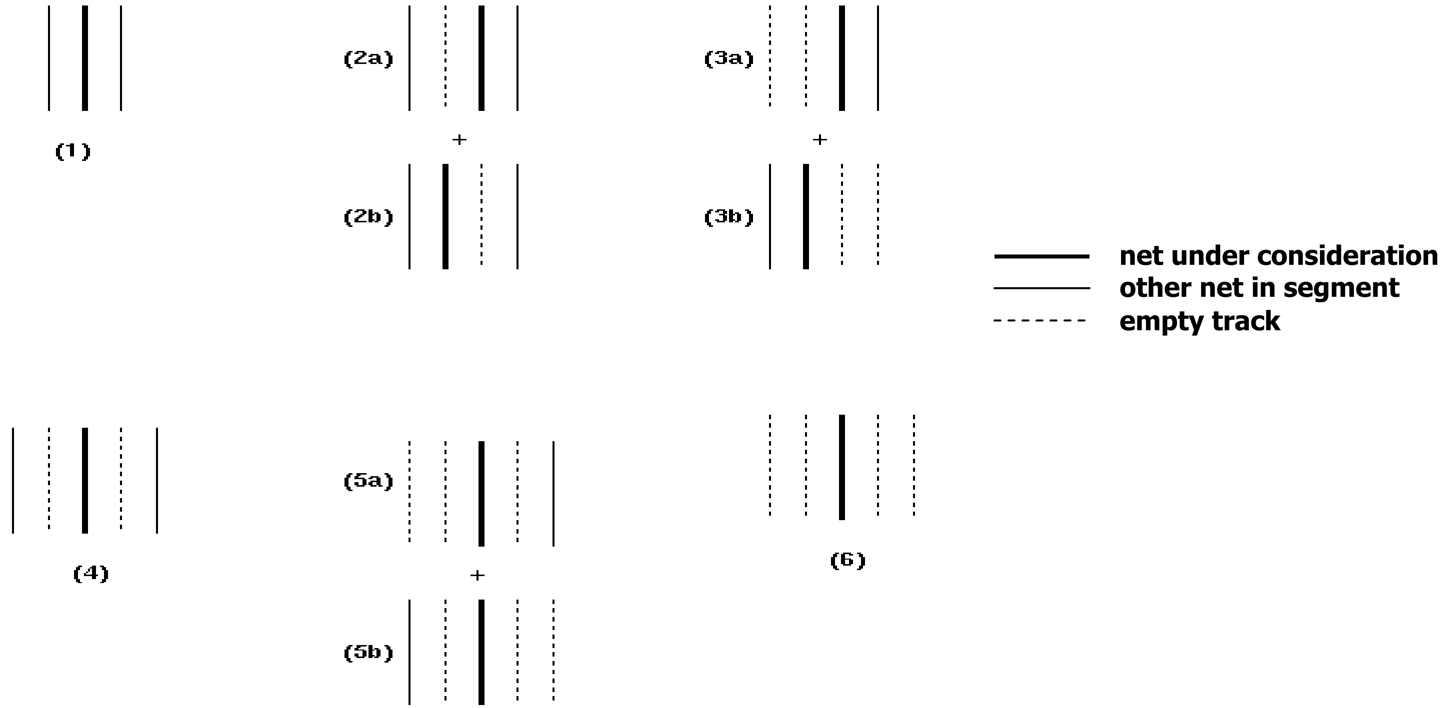
- Idea: Estimate per unit length capacitance for each segment by enumerating possible congestion configurations for that segment.
- Total number of possible configurations for a congestion map segment is

$$total\_configurations = \binom{n}{k} k!$$

- Infeasible to enumerate and characterize all possible congestion configurations
  - Capacitances of a net are effected only by the location of the nearest neighboring nets,
  - The effect of a neighboring net that is more than two tracks from the net is considered insignificant

Probabilistic Extraction

- Thus we consider 6 unique configurations:



- Capacitance to neighbors more than 2 tracks away is insignificant and ignored.

Probabilistic Extraction

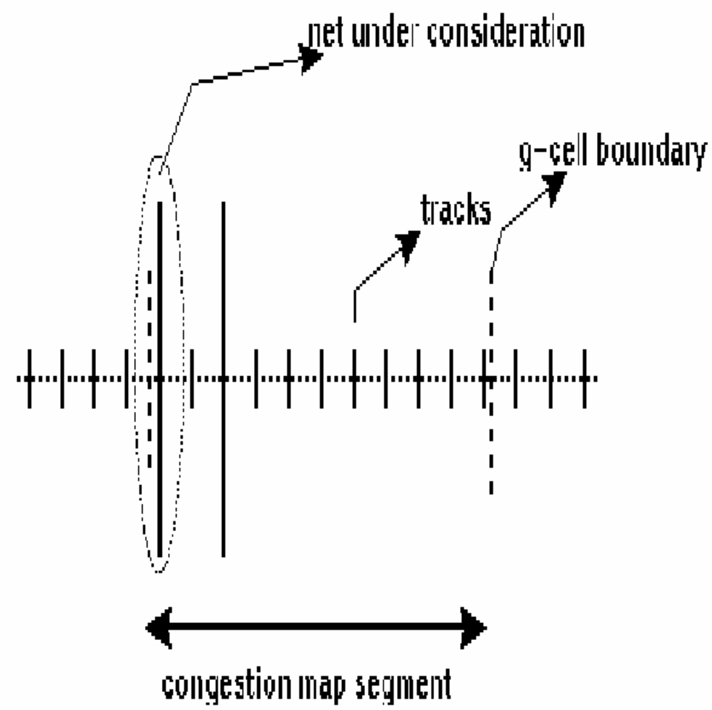
- After determining per unit length coupling and ground capacitance for each configuration, probability of each configuration is computed.

$$probability(i) = \frac{conf(i)}{total\_configurations} \qquad \text{for } i=1,\dots 6$$

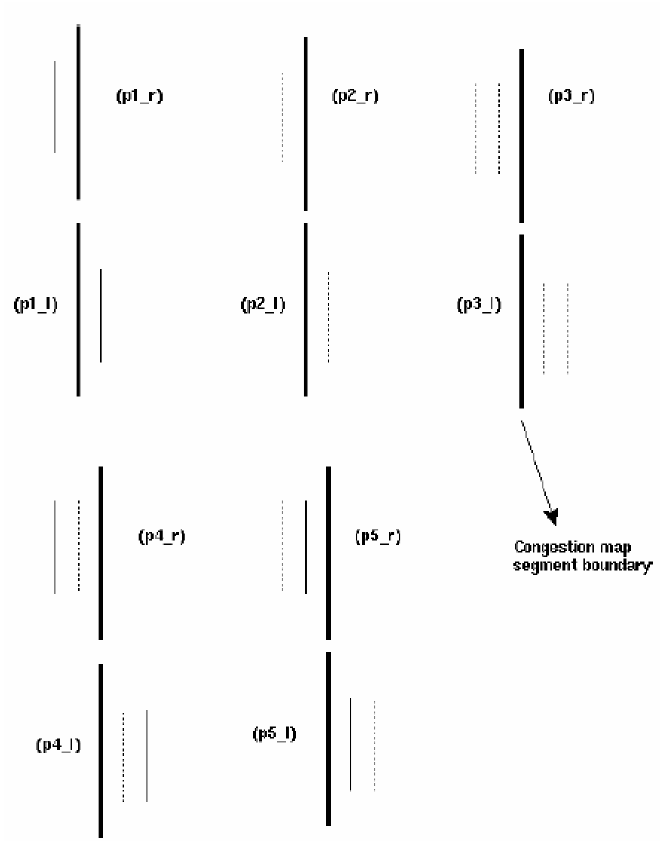
- The number of enumerations for each configuration depends on the number of tracks available (n) and number of tracks used (k) .
- For the cases where the net under consideration is close to the boundaries, the neighboring g-cells should also be taken into account

Neighboring g-cells

- How do we distribute these enumerations among the 6 defined density configurations ?
- Neighboring g-cell congestion should be taken into account



Neighboring g-cells



$$p1\_r = \frac{k\binom{n-1}{k-1}(k-1)!}{\binom{n}{k}k!}$$

Probability that the right most track of a g-cell is filled

Using these probabilities, we distribute the enumerations close to the boundaries among the defined configurations

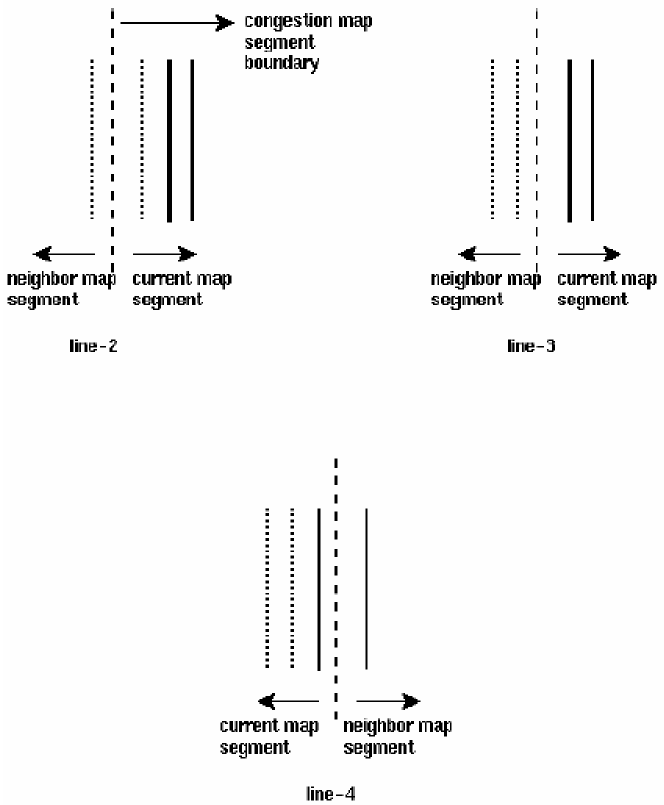
An Example

$$conf(3a) = (n-3)\times(k-1)\times\binom{n-4}{k-2}(k-2)!$$

$$+(k-1)\times\binom{n-3}{k-2}(k-2)!\times p2\_r$$

$$+(k-1)\times\binom{n-2}{k-2}(k-2)!\times p3\_r$$

$$+\binom{n-3}{k-1}(k-1)!\times p1\_l$$



Probabilistic Extraction

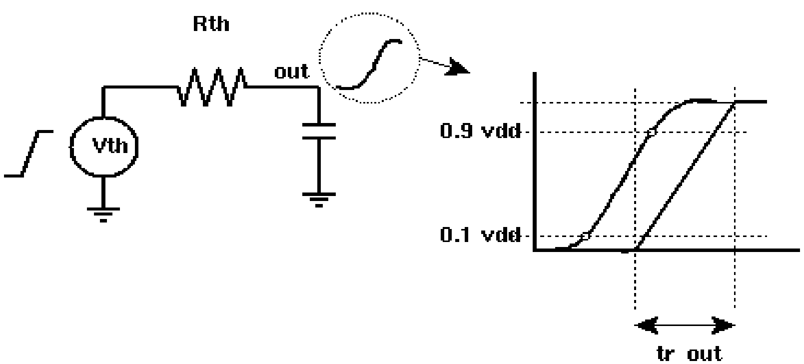
- For each segment that a net traverses, the pre-computed per-unit length coupling/ground capacitance $cc_{(i)} / cg_{(i)}$ of configuration i is weighted by the probability of configuration i, which is then scaled by the length of the segment. Estimated coupling/ground capacitance of a net segment is the summation of the weighted contributions for all configurations:

$$C_{c\,total(segment)} = \sum_{i=1}^{6} C_{c(i)} \times probability\left(conf_{(i)}\right) \times length(segment)$$

$$C_{g\,total(segment)} = \sum_{i=1}^{6} C_{g(i)} \times probability(config_{(i)}) \times length(segment)$$

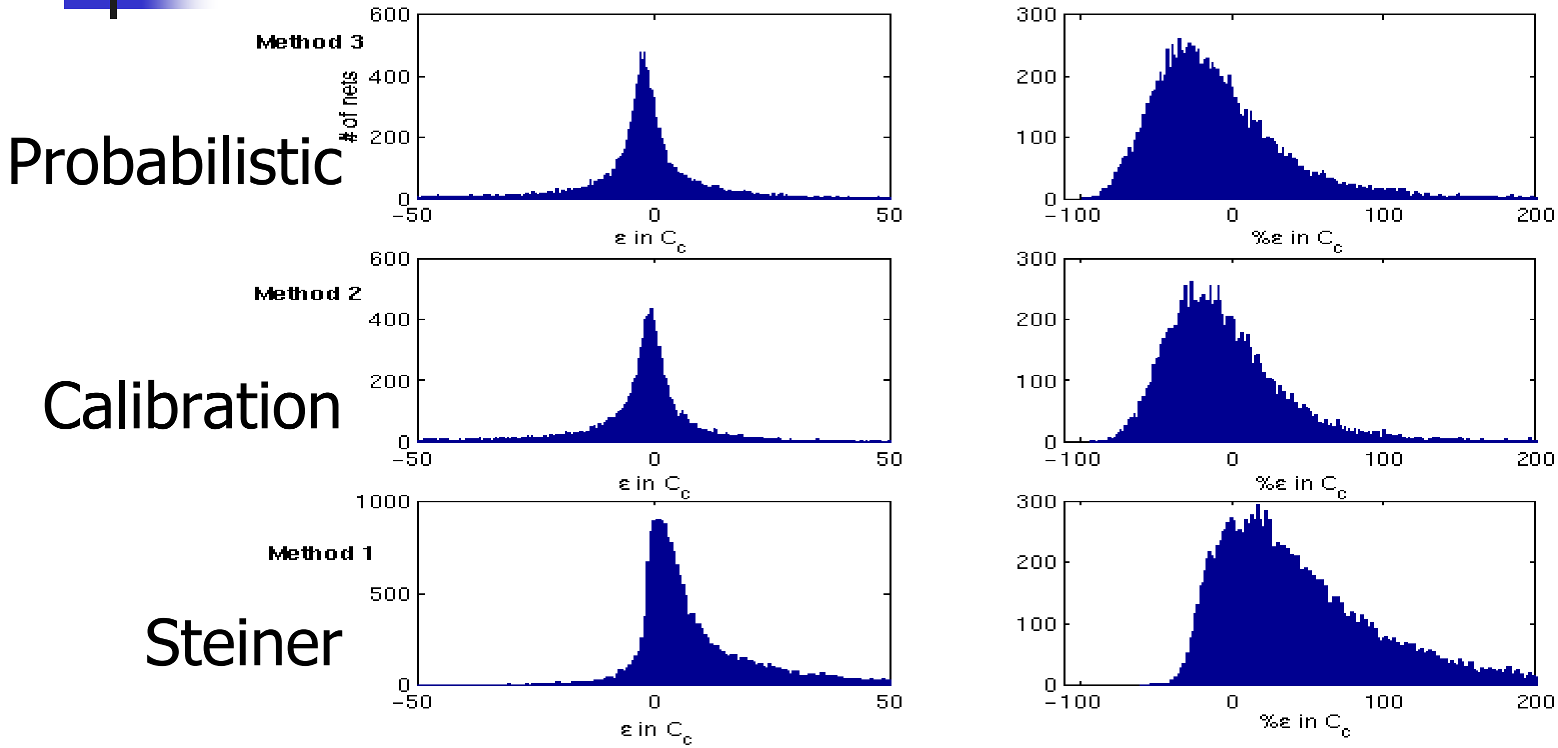
Aggressor Strength Estimation

- All nets that share a congestion map segment are possible neighbors.
- Likelihood of a net to be an aggressor to another net increases as the number of shared segments increases
- To estimate an average aggressor for a net:
  - Find, say,10 possible neighbors with the highest number of shared segments. For each of these aggressors,
    - Find total capacitance, as discussed before,
    - Obtain Thevenin model of the driver gate from cell library
    - Compute a weighted average transition time $t_{r_{out}}$ on the aggressor net



$$v_{out}(t)=\begin{cases}\dfrac{-RC+t+RCe^{-t/RC}}{t_r}, & 0<t\le t_r\\ \dfrac{-RC}{t_r}(e^{t_r/RC}-1)e^{-t/RC}+1, & t>t_r\end{cases}$$

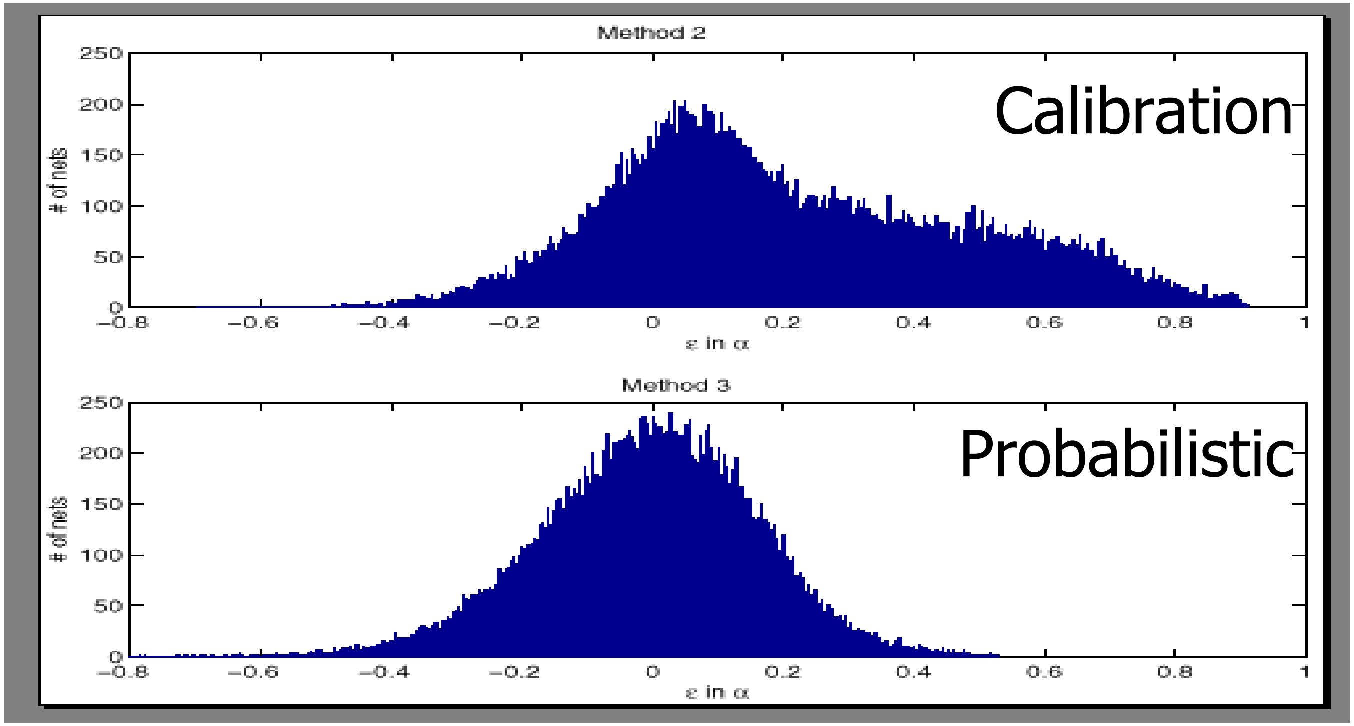
Estimation errors in 0.18μ processors (Two processors, 58K and 125K nets)



- Method 1: Steiner tree based approach: α=0.5 --> 52.74% and 85.14% error
- Method 2: Calibration method --> -2% and 16% error
- Method 3: Probabilistic extraction method --> -1.5% and 8% error

Calibration vs. Probabilistic

Absolute Error in α

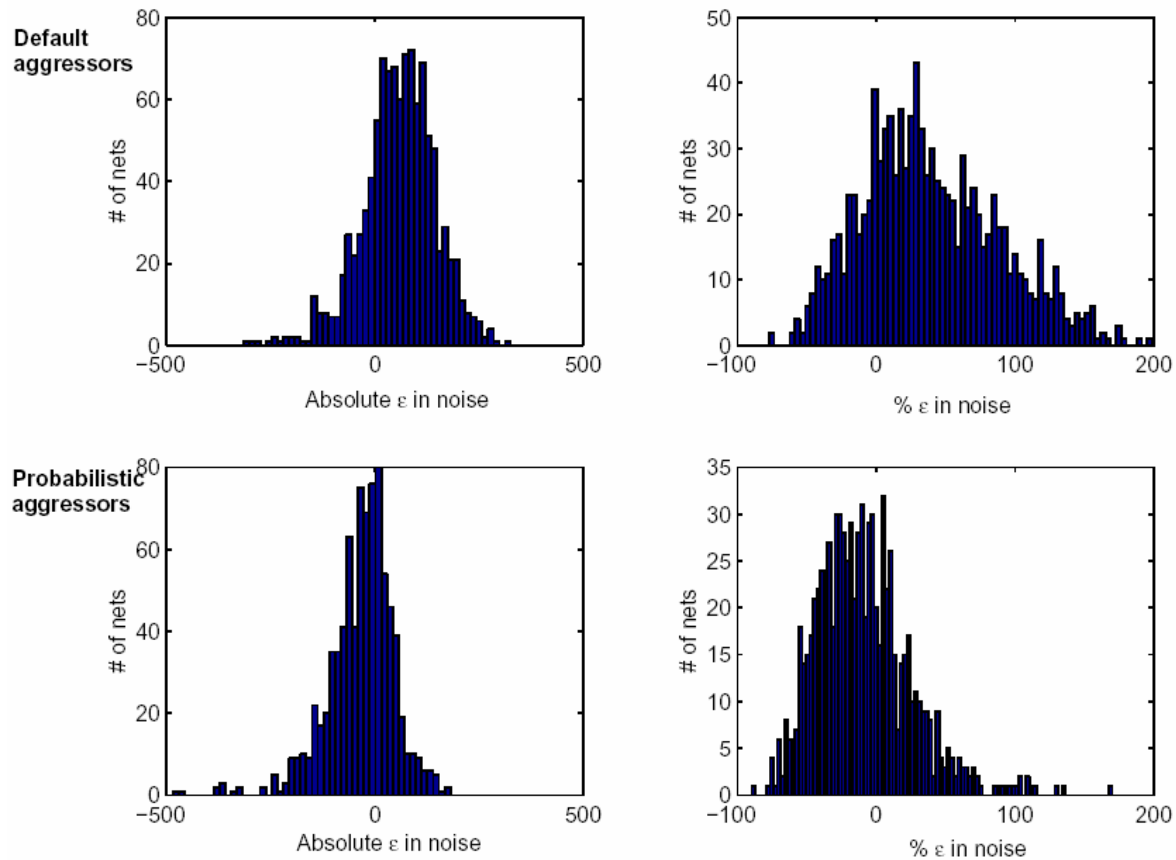


- Note, probabilistic method estimated localized α values, but calibration method used one α for all segments of the same net.

Probabilistic vs. Default Aggressors

Using probabilistic aggressors instead of strong default aggressors reduce average error in noise peak from 40% to 9%

A trade-off exists between estimated noise peak and early identified noisy nets



Failing nets in pre and post route noise analysis

Chip	Method	Missed	Common	False
chip-1	Steiner	148	694	531
	Calibration	207	635	241
	Probabilistic	161	681	205
chip-2	Steiner	134	487	3509
	Calibration	193	428	1591
	Probabilistic	90	531	1409

- Common: # of nets that fail in both pre- and post-route noise analysis
- Missed : # of nets not identified by pre-route noise analysis that subsequently failed in post-route noise analysis
- False : # of nets that failed in pre-route noise analysis but not in post-route noise analysis
- Probabilistic method reduced the number of false failures by as much as 60% while predicting about the same number of real failures as the Steiner based method

Conclusion

- Post-route stage is too late and too expensive for fixing noise violations
- Proposed 3 techniques for pre-route noise estimation based on Steiner/Global routing
- Congestion from global router combined with a probabilistic estimation of coupling and aggressor strength can predict noise violations quite reliably