Statistical Timing Based Optimization using Gate Sizing

Aseem Agarwal, Kaviraj Chopra, David Blaauw, *Vladimir Zolotov

University of Michigan, Ann Arbor
* IBM T. J. Watson Research Center, Yorktown heights

TAU
1st March, 2005
Overview – Circuit optimization

- Optimization formulation: Minimize the circuit Area/Power
  - Constraint on circuit delay (timing yield) or vice versa
  - Determine design variables (ex: gate size)
- Outcome
  - Improved power, area, delay, noise – well tuned circuit

- Deterministic optimization – delay is non-statistical
  - Compute the sensitivity of the obj. fn. to design variables using STA
  - Feed into Non-linear optimizer

- Process variation not properly accounted for
  - Yield loss
  - Need for a Statistically-aware optimizer (statistical delay models)
Outline of talk

- Need for Statistical Timing Based Optimization
- Our Contribution
- Results and Analysis
- Summary
Need for Statistical Timing Based Optimization

- Deterministic Optimization creates a so-called *timing-wall*
  - No advantage in improving non-critical paths
  - Degrade statistical performance

- Statistical Delay improvement (given a det. sized ckt.)
  - With an additional area penalty
    - Same nominal delay
    - Iso-area
      - Increased nominal delay
Critical and near-critical paths of a design
- Different standard deviations
- Size down low sigma paths / Size up high sigma paths (one trade-off)
  - Decision made by the optimization objective
Prior Work

- Deterministic delay model based
  - H. Hashimoto [ISCAS ‘01]
    - Deterministic optimization creates a *timing wall*
    - Height of the *wall* impacts the statistical delay by large amt.
  - X. Bai [DAC ‘02]
    - Provide incentive in the det. formulation to avoid a *wall*

- Statistical delay model based (Non-linear programming problem)
  - E. Jacobs [DATE ‘00]
    - Gaussian approximation for max (analytical formulation)
    - Sensitivity computation complexity is $O(n^2)$
  - S. Raj [DAC ‘04]
    - Path based approach – enum. all paths in worst case
    - Demonstrate large improvements on benchmark ckts.
Proposed work and assumptions

- Statistical delay model based (Co-ordinate descent)
  - Uses our bound based SSTA approach
  - Sensitivity computation
  - Exact approach for finding the high sensitivity gates
    - Fast due to our proposed pruning method
  - Combined approach using a slack based heuristic
    - Runtime improvement upto 2 orders of magnitude
  - Demonstrate significant improvements comp. to det.
  - Consider the ind. random component of intra-die
    - Extension to include spatial correlation
  - Consider pdfs to be bounded on max and min values
Outline of talk

- Need for Statistical Timing Based Optimization
- Our Contribution
  - Different optimization objectives
  - Brute force formulation
  - Theory of perturbation bounds
  - Exact sensitivity computation
  - Heuristic sensitivity computation
- Results and Analysis
- Summary
Optimization Objective

- Optimization changes both mean and shape of PDF
  - Need measure of quality

- Simple objective – 99% confidence
Optimization Objective

- Different cost functions for ASIC and Microprocessors
Brute-force formulation

- Statistical objective function
  - helps evaluate the change in the waveform
- Sensitivity Computation
  - Complexity \( O(V \times E) \)
  - Need better sensitivity computation
Outline of talk

- Need for Statistical Timing Based Optimization
- Our Contribution
  - Different optimization objectives
  - Brute force formulation
  - Theory of perturbation bounds
  - Exact sensitivity computation
  - Heuristic sensitivity computation
- Results and Analysis
- Summary
Perturbation bounds

Arrival time cdfs

\( \Delta_i \) diminishes

\( \Delta_i \) remains same

\( \Delta_i \) diminishes proof in the paper
Exact pruning of nodes

- $\Delta_i$ can only diminish
  - This property can be used for pruning
    - Propagate a perturbation to the sink node
    - Prune other gates in the circuit if they are lesser
      - Reduces complexity

- Need to determine gates to be propagated first
  - Greedy selection
  - Level by level propagation
Exact Sensitivity computation

- Perform a run of SSTA
- Propagate pert. arrival times CDFs till the level of the gate
- Sort the gates by perturbation
- Select the best gate greedily and prop. one level forward
- Update the sorted order
- If gate reaches the sink node update lower bound
- Prune gates below the lower bound
  - Exact pruning
- When pruning is over
  - Either size the best gate / provide sensitivities to a n.l.o
- Repeat for next iteration
Heuristic Sensitivity computation

- Compute slack distributions by backward propagation
- Convolve forward and backward pdfs
- Compute remainder cdf (remainder ckt)
- Convolve perturbed forward and backward pdfs
- Max with the remainder cdf
- Compare with circuit delay cdf to obtain sensitivity

- Extremely fast
- Combined appr.
Experimental setup

- Synthesized ISCAS benchmark circuits (180 nm library)
- The delay model used is \( D_e = D_{int} + K \times \frac{C_{load}}{C_{cell}} \)
- Standard deviation is 10% of mean delay
- Compare w/ det. optim. using MINOS
  - Statistical timing run on the obtained circuit
- Area delay curves plotted for 800 sizing iterations
Runtime Results

- 20 X by bound based prune, 89 X improvement by combined approach

<table>
<thead>
<tr>
<th>Circuit name</th>
<th>Average time per iteration (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>brute-f</td>
</tr>
<tr>
<td>c432</td>
<td>5</td>
</tr>
<tr>
<td>c499</td>
<td>90</td>
</tr>
<tr>
<td>c880</td>
<td>15</td>
</tr>
<tr>
<td>c1355</td>
<td>95</td>
</tr>
<tr>
<td>c1908</td>
<td>102</td>
</tr>
<tr>
<td>c2670</td>
<td>43</td>
</tr>
<tr>
<td>c3540</td>
<td>194</td>
</tr>
<tr>
<td>c5315</td>
<td>403</td>
</tr>
<tr>
<td>c6288</td>
<td>3600</td>
</tr>
<tr>
<td>c7552</td>
<td>1190</td>
</tr>
</tbody>
</table>
# Optimization results – 99% delay

- Av. delay imp – 7.6 %, Av. Sigma imp – 17%

<table>
<thead>
<tr>
<th>name</th>
<th>det. delay</th>
<th>slack-based delay</th>
<th>%impr</th>
<th>bound-based delay</th>
<th>%impr</th>
<th>combined approach delay</th>
<th>%imp</th>
<th>%sigma imp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>3.45</td>
<td>3.40</td>
<td>1.4</td>
<td>3.25</td>
<td>5.8</td>
<td>3.25</td>
<td>5.8</td>
<td>12.2</td>
</tr>
<tr>
<td>c499</td>
<td>4.05</td>
<td>3.48</td>
<td>14.0</td>
<td>3.38</td>
<td>16.5</td>
<td>3.38</td>
<td>16.5</td>
<td>31.4</td>
</tr>
<tr>
<td>c880</td>
<td>4.18</td>
<td>4.04</td>
<td>3.3</td>
<td>3.94</td>
<td>5.74</td>
<td>3.94</td>
<td>5.74</td>
<td>13.8</td>
</tr>
<tr>
<td>c1355</td>
<td>4.70</td>
<td>4.25</td>
<td>9.5</td>
<td>4.10</td>
<td>12.7</td>
<td>4.10</td>
<td>12.7</td>
<td>30.7</td>
</tr>
<tr>
<td>c1908</td>
<td>6.20</td>
<td>6.02</td>
<td>2.9</td>
<td>5.82</td>
<td>6.1</td>
<td>5.82</td>
<td>6.1</td>
<td>10.8</td>
</tr>
<tr>
<td>c2670</td>
<td>3.61</td>
<td>3.55</td>
<td>1.7</td>
<td>3.50</td>
<td>3.0</td>
<td>3.50</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>c3540</td>
<td>5.98</td>
<td>5.80</td>
<td>5.2</td>
<td>5.70</td>
<td>6.9</td>
<td>5.70</td>
<td>6.9</td>
<td>13.5</td>
</tr>
<tr>
<td>c5315</td>
<td>5.90</td>
<td>5.70</td>
<td>3.4</td>
<td>5.40</td>
<td>8.47</td>
<td>5.40</td>
<td>8.47</td>
<td>14.8</td>
</tr>
<tr>
<td>c6288</td>
<td>15.8</td>
<td>15.5</td>
<td>1.9</td>
<td>15.05</td>
<td>4.75</td>
<td>15.05</td>
<td>4.75</td>
<td>23.0</td>
</tr>
<tr>
<td>c7552</td>
<td>8.10</td>
<td>7.80</td>
<td>3.8</td>
<td>7.60</td>
<td>6.17</td>
<td>7.60</td>
<td>6.17</td>
<td>13.4</td>
</tr>
</tbody>
</table>
3 different optimizations – c6288

- Deterministic Optimization
- Statistical Slack-based Optimization
- Our Combined Optimization and bound-based pruning method
c880 – Optimized circuit delay PDF

The diagram illustrates the probability density function (PDF) of circuit delay. It shows two curves:

1. Statistical optimization PDF using our combined approach.
2. Deterministic optimization PDF using MINOS optimal solution.

The deterministic approach reduces variance compared to the statistical approach.
Accuracy of bounds – c3540

- Deterministic Optimization
  - 99% pt. using bounds
  - 99% pt. using Monte Carlo

- Statistical Optimization
  - 99% pt. bounds
  - 99% pt. Monte Carlo
Cost functions – c880

- Linear cost function
- Optimized using linear cost function
- Optimized using 99% pt.

(delay (ns) vs. p)
Summary

- Brute force formulation
- Novel theory of perturbation bounds
- Statistical optimization algorithm
- Exact analysis / reduced complexity - upto 20X run time reduction on benchmarks
- Combined method - upto 89X reduction
- Significant sigma and delay improvement
Thank You