Timing Slack Aware Incremental Register Placement with Non-uniform Grid
Generation for Clock Mesh Synthesis

Speaker: Jianchao Lu
Jianchao Lu, Xiaomi Mao, Baris Taskin
VLSI Lab
Electrical & Computer Engineering
Drexel University
Outline

- Preliminaries
- Previous Works
- Methodology
- Experimental Results
- Conclusions
Clock Mesh Network

- Consists of top level clock tree, mesh grids and stub wires.
Clock network is a global network of interconnect wires and buffers.

Clock signal switching introduces a lot of dynamic power dissipation.

Consumes more than 40% of the total power.

\[ P = \alpha C V^2 f_{clk} \]

- Switching factor
- VDD
- Frequency
- Capacitance
Power Dissipation on Clock Network

- Clock network is a global network of interconnect wires and buffers.
- Clock signal switching introduces a lot of dynamic power dissipation.
- Consumes more than 40% of the total power.

\[ P = \alpha C V^2 f_{clk} \]

Switching capacitance = \( \alpha \times C_{\text{total}} = (C_{\text{grid}} + C_{\text{stub}} + C_{\text{tree}}) \)
Outline

- Preliminaries
- Previous Works
- Methodology
- Experimental Results
- Conclusions
Most Relevant Previous Works


Meshworks [1]

- Identifies relationship between grid size and total mesh wire.
- Optimal grid size based on skew.
- Mesh reduction.
- Modified buffer driver insertion.

ILP Based Mesh Synthesis [3]

- Mesh generation and sink assignment algorithms.

Outline

- Preliminaries
- Previous Works
- Methodology
- Experimental Results
- Conclusions
Proposed Method

- Optimizing the placement during the clock mesh synthesis.

![Diagram showing steps of the proposed method]

1. Initial placement
2. Construct feasible moving regions
3. Mesh wire generation and placement
4. Incremental register placement
5. Buffer insertion and top level tree generation
Step 1: Creating Feasible Moving Region of Each Register
Creating Feasible Moving Regions
Creating Feasible Moving Regions
Creating Feasible Moving Regions
Creating Feasible Moving Regions
Creating Feasible Moving Regions
Step 2: Mesh Generations

- Registers can be moved in feasible moving regions without negative timing slack.
- Choose the minimum amount of mesh tracks that all the registers can be moved on as the mesh network.
Step 2: Mesh Generations

- Registers can be moved in feasible moving regions without negative timing slack.
- Choose the minimum amount of mesh tracks that all the registers can be moved on as the mesh network.
Mesh Generation Problem

- Problem: Assume each mesh track is a set and each register is an element. Finding the minimum amount of sets that includes all the elements is equivalent to finding the minimum amount of mesh tracks that can connect to the mesh wires.
- Greedy algorithm: Greedily add the candidate mesh track with the minimum cost.
- Cost of each grid wire = total distance of the registers from the grid/number of new elements added in the solution set.
Step 3: Incremental Register Placement

Objective: minimizing total stub wire.

Subject to:
- The timing constraints.
- The registers should be non-overlapped.

Variables:
- Registers locations.

Minimize the total stub wirelength.

\[
\begin{align*}
\text{min} & \quad \sum_{k} w_{stub}^k \\
\text{s.t.} & \quad D_{fo_k}^f + D_{CQ}^f + D_{f_i}^f \
& \quad \leq T - S_f - L_i f - D_{mk_i}^f, \quad \forall (R_i \rightarrow R_f) \\
& \quad D_{fo_k}^f = K_w C_0 w_{fo_k}^f, \quad \forall R_i \\
& \quad D_{CQ}^f = D_{R0} + K_w C_0 w_{fo_k}^f, \quad \forall R_i \\
& \quad D_{f_i}^f = K_w C_0 w_{f_i}^f + D_{G0} + K_w C_0 w_{f_i}^f, \quad \forall R_f \\
& \quad w_{stub}^k = x_{dist}(R_i, M_{vi}) \ (\text{or } y_{dist}(R_i, M_{hi})) , \quad \forall R_i \\
& \quad x_{dist}(R_i, M_{vi}) \geq x_{R_i} - x_{M_{vi}}, \quad \forall R_i \\
& \quad x_{dist}(R_i, M_{vi}) \geq x_{M_{vi}} - x_{R_i}, \quad \forall R_i \\
& \quad y_{dist}(R_i, M_{hi}) \geq y_{R_i} - y_{M_{hi}}, \quad \forall R_i \\
& \quad y_{dist}(R_i, M_{hi}) \geq y_{M_{hi}} - y_{R_i}, \quad \forall R_i \\
& \quad w_{f_o_k} = x_{dist}(R_i, f_{o_k}) + y_{dist}(R_i, f_{o_k}), \quad \forall R_i \\
& \quad x_{dist}(R_i, f_{o_k}) \geq x_{R_i} - x_{f_{o_k}}, \quad \forall R_i \\
& \quad x_{dist}(R_i, f_{o_k}) \geq x_{f_{o_k}} - x_{R_i}, \quad \forall R_i \\
& \quad y_{dist}(R_i, f_{o_k}) \geq y_{R_i} - y_{f_{o_k}}, \quad \forall R_i \\
& \quad y_{dist}(R_i, f_{o_k}) \geq y_{f_{o_k}} - y_{R_i}, \quad \forall R_i \\
& \quad w_{f_i} = x_{dist}(R_f, f_i) + y_{dist}(R_f, f_i), \quad \forall R_f \\
& \quad x_{dist}(R_f, f_i) \geq x_{R_f} - x_{f_i}, \quad \forall R_f \\
& \quad x_{dist}(R_f, f_i) \geq x_{f_i} - x_{R_f}, \quad \forall R_f \\
& \quad y_{dist}(R_f, f_i) \geq y_{R_f} - y_{f_i}, \quad \forall R_f \\
& \quad y_{dist}(R_f, f_i) \geq y_{f_i} - y_{R_f}, \quad \forall R_f \\
& \quad x_{R_i} - x_{R_f} \geq W_f, \\
& \quad \text{or } x_{R_f} - x_{R_i} \geq W_f, \\
& \quad \text{or } y_{R_f} - y_{R_i} \geq L_f, \\
& \quad \text{or } y_{R_i} - y_{R_f} \geq L_f.
\end{align*}
\]
The Incremental Placement Results
(s35932 in ISCAS89)

Before placement

After placement
Top Level Clock Tree Generation

- Insert buffer drivers on the intersection of the mesh grid wires[1][2].
- Generate top level clock tree where the sinks are buffer drivers of the mesh grid wires. (Buffered DME)
Outline

- Preliminaries
- Previous Works
- Methodology
- Experimental Results
- Conclusions
Experimental Results

Set 1: Compare the proposed method with [2] using different grid sizes.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Proposed</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>s13207</td>
<td>6*7</td>
<td>8*8</td>
</tr>
<tr>
<td>s15850</td>
<td>5*4</td>
<td>8*8</td>
</tr>
<tr>
<td>s35932</td>
<td>11*7</td>
<td>12*12</td>
</tr>
<tr>
<td>s38417*</td>
<td>10*9</td>
<td>12*12</td>
</tr>
<tr>
<td>s38584</td>
<td>12*7</td>
<td>11*11</td>
</tr>
</tbody>
</table>

Set 2: Compare the proposed method with [2] using the same grid sizes.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Proposed</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>s13207</td>
<td>6*7</td>
<td>6*7</td>
</tr>
<tr>
<td>s15850</td>
<td>5*4</td>
<td>5*4</td>
</tr>
<tr>
<td>s35932</td>
<td>11*7</td>
<td>11*7</td>
</tr>
<tr>
<td>s38417*</td>
<td>10*9</td>
<td>10*9</td>
</tr>
<tr>
<td>s38584</td>
<td>12*7</td>
<td>12*7</td>
</tr>
</tbody>
</table>

Mesh Wire Reduction

Set 1 (Different grid size)

Average improvement of 51.9%.

Set 2 (Same grid size)

Average improvement of 50.8%.
Clock Power Reduction

Set 1 (Different grid size)

- Average improvement of 48.3%.

Set 2 (Same grid size)

- Average improvement of 28.1%.
Skew Results (45nm PTM)

Set 1 (Different grid size)

Set 2 (Same grid size)

Average skew is in the same range.

Skew is improved by 0.8ps.
Trade-off

- The trade-off is the logic wirelength change due to the register placement.
Implications of Placement

Congestion

Before Register Placement

After Register Placement
Routing Congestion

- The timing slack is decreased by an average of 22ps, which is very limited compared to the 2ns clock period.
Outline

- Preliminaries
- Previous Works
- Methodology
- Experimental Results
- Conclusions
Conclusions

- **Advantages**
  - Significantly reduced power dissipation.
  - Guaranteed timing slack (pre-routing).

- **Disadvantages**
  - Power density increase.
  - Timing slack decrease.
Thank You!