A Fast Power Network Optimization Algorithm for Improving Dynamic IR-drop

Speaker: Yang-Tai Kung
Dep. of EE, National Cheng Kung University
Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion
Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion
Introduction

Powerplanning becomes a more critical step.

- A modern chip consumes more power when its design complexity continually increases.
- A design has a small margin for voltage drop as technology process advances.

Most of previous works only consider static power consumption without handling dynamic power consumption during powerplanning or power network optimization.

For simplicity, existing studies either adopt the worst or average power consumption when dynamic IR-drop is considered.

- However, it may waste unnecessary routing resource if they adopt the worst power consumption to perform voltage analysis.
- It still has chance to violate the IR-drop or EM constraints if they adopt the average power consumption.
Introduction

Power network optimization considering dynamic IR-drop is very difficult and is very time consuming without a good strategy.

- Power consumption of a device may be various at different time and it is composed of a large number of Power consumption Files (PFs for short) during a period of time.
- It is impossible to optimize power network according to all PFs.

There exist limited researches about power network optimization for dynamic IR-drop.

An efficient and effective power optimization approach to consider dynamic IR-drop is required by industry.
Optimize power network according to critical PFs selected from a large set of PFs.
- Apply K-clustering algorithm to classify all PFs into several groups according to power distribution maps (PDMs for short) induced by different PFs.
- Generate a representative power consumption file (RPF for short) for each group to represent the PFs in the group.

Propose an efficient and effective approach to repair voltage violations in the hotspot region of power network which are induced by many PFs.

The experimental results have shown that the ratio of IR-drop violations can be significantly reduced by our methodology compared to the classic approach.
Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion
Input:
- Locations and shapes of standard cells and macros from DEF and LEF files.
- The average power consumption of standard cells and macros from PTPX (primetime).
- The capacitances of pins and wires and the power network of the chip extracted from IC Compiler.
- Switching activity information from RTL Value Change Dump (VCD for short).

Output:
- The widths of vertical power stripes (VPSs for short).

Constraints:
- The IR-drop constraint
- The maximum current density (EM) constraint
- The minimum width constraint
- The maximum width constraint
Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion
Generate PFs according to PrimeTime and VCD file, where the switching power of cells are estimated from signal switching information in a VCD file for all time slots.

\[ P_{\text{switching}} = \frac{1}{2} \times C \times V^2 \]

Total switching power consumption at each time slot.
Overview of Our Methodology

- **DEF/LEF/VCD**
- Generate all power consumption files (PFs) from VCD
- Eliminate non-critical power consumption files
- Construct power distribution maps
- Discover representative power consumption files (RPFs)
- Optimize power network according to RPFs

- **Eliminate non-critical PFs.**
  - A PF is not critical when its total power consumption is smaller than average power consumption.

![Power Consumption Chart]

- **DEF/LEF/VCD**
- Total Power Consumption (W)
- Time Scale (ps)
- Non-critical power consumption files.
Overview of Our Methodology

- Build PDMs for the critical PFs.
  - A map is composed of $h$ grids.
  - A PDM for a PF $f_i$ is represented by a vector $X_i = <x_1^i, x_2^i, \ldots x_h^i>$, where $x_j^i$ denotes power consumption in a grid $j$.

1. DEF/LEF/VCD
2. Generate all power consumption files (PFs) from VCD
3. Eliminate non-critical power consumption files
4. Construct power distribution maps
5. Discover representative power consumption files (RPFs)
6. Optimize power network according to RPFs

Each PF

<table>
<thead>
<tr>
<th>PF $f_1$</th>
<th>PF $f_2$</th>
<th>PF $f_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.003</td>
<td>0.023</td>
<td>0.002</td>
</tr>
<tr>
<td>0.013</td>
<td>0.017</td>
<td>0.037</td>
</tr>
<tr>
<td>0.017</td>
<td>0.12</td>
<td>0.019</td>
</tr>
<tr>
<td>0.12</td>
<td>0.031</td>
<td>0.034</td>
</tr>
</tbody>
</table>
Overview of Our Methodology

- Classify PFs into different clusters according to their power distributions, and then select a RPF for each cluster.

1. **Generate all power consumption files (PFs) from VCD**
2. **Eliminate non-critical power consumption files**
3. **Construct power distribution maps**
4. **Discover representative power consumption files (RPFs)**
5. **Optimize power network according to RPFs**
Overview of Our Methodology

- **DEF/LEF/VCD**
- Generate all power consumption files from a VCD file
- Eliminate non-critical power consumption files
- Construct power distribution maps
- Discover representative power consumption files (RPFs)
- Optimize power network according to RPFs

- Construct a voltage violation map (VVM for short) for a PF, where each grid $g$ corresponds to a node in a power network.
  - Positive value in a grid denotes the IR-drop violation value of a node; otherwise, the value is zero.

<table>
<thead>
<tr>
<th></th>
<th>0.2</th>
<th>0.3</th>
<th>0.1</th>
<th>0.3</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Construct a VVM for a PF
Overview of Our Methodology

1. Generate all power consumption files (PFS) from VCD.
2. Eliminate non-critical power consumption files.
3. Construct power distribution maps.
4. Discover representative power consumption files (RPFs).
5. Optimize power network according to RPFs.

Repeatedly select a critical RPF to optimize power network until the power network has no IR-drop violation for all RPFs.

- Construct VVMs for RPFs
- Any IR-Drop Violation?
  - Yes: Optimize Power Network by Selection of a Critical RPF
  - No: Update VVMs for RPFs

Optimized P/G Network
Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
  - Correlation Metric
  - K-Clustering Algorithm
  - Cost Function for a RPF in Each Group
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion
**Discovery of Representative Power Consumption Files**

- Discover RPFs and optimize power network according to these files.
  - Runtime can be saved significantly without sacrificing design quality.

**Method:**
- **Step1:** Cluster PFs according to their power distributions.
- **Step2:** Find a RPF for each cluster.
We use distance $d_{i,j}$ to represent the similarity between PDMs $X_i$ and $X_j$ of two PFs $f_i$ and $f_j$, which is calculated by the following function:

$$d_{i,j} = 1 - \rho_{i,j}, \ 0 \leq d_{i,j} \leq 2$$

where $\rho_{i,j}$ is called as statistical correlation, which is defined as follows:

$$\rho_{i,j} = \frac{\text{cov}(i,j)}{\sigma_i \cdot \sigma_j}, \ -1 \leq \rho \leq 1$$

where $\sigma_i$ ($\sigma_j$) is standard deviation.

function $\text{cov}(i, j)$ represents the covariance of PDMs $X_i$ and $X_j$ as follows:

$$\text{cov}(i, j) = \frac{1}{h} \cdot \sum_{b=1}^{h} [(x^i_b - \bar{x}^i) \cdot (x^j_b - \bar{x}^j)]$$

where $\bar{x}^i$ ($\bar{x}^j$) denotes the average power consumption in the vector $X_i (X_j)$.

$h$ denotes the number of bins in a PDM.
K-Clustering Algorithm

- Divide all PFs into K clusters according to their power distributions.
  - PFs in each cluster have similar power distribution.
- Construction of a complete graph $G(N, E)$ before applying K cluster algorithm.
  - Initialize a node $n_i$ for each PF $f_i$.
  - Connect an edge $e_{i,j}$ for each pair of $n_i$ and $n_j$, where $d_{i,j}$ denote the weight of the edge.
K-Clustering Algorithm

- In the beginning, we consider each node $n_i$ as a cluster $c_i$.
- The algorithm will repeatedly combine two clusters until the number of clusters is $K$.
- After a node $n_i$ is merged into a cluster, we use $c_i$ to represent the associated cluster of $n_i$ for each edge $e_{i,j}$ for simplicity.
Algorithm: K-Clustering Algorithm

Input: a weighted complete graph $G(N, E)$; $K$ is number of clusters;
Output: $K$ clusters, each cluster has the similar power consumption files;

1. Sort all edges in $E$ in the non-decreasing order and store in $L$.
2. $S \leftarrow \{\emptyset\}$; \hspace{1cm} ▶ Let $S$ denote a set of edges connecting to two clusters.
3. $\kappa \leftarrow |N|;$
4. for (each edge $e_{i,j}$ in $L$ ) do
5. \hspace{1cm} for (each edge $e_{p,q}$ connecting to $c_i$ and $c_j$ ) do
6. \hspace{2cm} $S \leftarrow S \cup \{d_{p,q}\}$;
7. \hspace{1cm} end for
8. \hspace{1cm} if ($d_{i,j}$ is the smallest value in $S$ ) then
9. \hspace{2cm} $c_i \leftarrow c_i \cup c_j$;
10. \hspace{2cm} Remove all edges $S$ from $E$;
11. \hspace{2cm} $\kappa \leftarrow \kappa - 1$;
12. \hspace{1cm} end if
13. \hspace{1cm} if ($\kappa = K$) then
14. \hspace{2cm} break;
15. \hspace{1cm} end if
16. $S \leftarrow \{\emptyset\}$;
17. end for
Select a RPF from each cluster $|C|$ if it has the maximum value according to the following function:

$$\Psi(f_i) = \sum_{b=1}^{h} (\alpha \cdot x_b^i + r_b \cdot \delta_b)$$

- $x_b^i$: the power consumption of bin $b$ in PF $f_i$.
- $\alpha$: the power consumption weight of user defined.
- $\delta_b$: the distance of bin $b$ to the closest power pad.
- $r_b = \begin{cases} 1, & \text{if } x_b^i \text{ is top } \eta\% \text{ power consumption in } X_i \\ 0, & \text{else} \end{cases}$
Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion
Optimization of a Power Network According to RPFs

1. Initialization of VVMs for RPFs
2. Identification of a Hotspot Region for Each RPF
3. Selection of the RPF with the Most Severe Hotspot Region
4. Optimization of a Power Network According to the Selected RPF
5. Fast Update of VVMs for all RPFs

Any IR-Drop Violation?

YES

NO

Power Network Output
Initialization of Voltage Violation Maps for RPFs

Optimization of a Power Network According to RPFs

- Initialization of VVMs for RPFs
- Identification of a Hotspot Region for Each RPF
- Selection of the RPF with the Most Severe Hotspot Region
- Optimization of a Power Network According to the Selected RPF
- Fast Update of VVMs for all RPFs
- Any IR-Drop Violation?
- Power Network Output
Identification of a Hotspot Region for Each RPP

Optimization of a Power Network According to RPFs

- Initialization of VVMs for RPFs
- Identification of a Hotspot Region for Each RPF
- Selection of the RPF with the Most Severe Hotspot Region
- Optimization of a Power Network According to the Selected RPF
- Fast Update of VVMs for all RPFs

Any IR-Drop Violation?

YES

NO

Power Network Output
identification of a Hotspot Region for Each RPF

- Discover the grid with the most serious IR-drop violation for each RPF.
- Generate a window with a proper size to cover the grid.
Selection of RPF with the Most Serious IR-drop Region

Optimization of a Power Network According to RPFs

- Initialization of VVMs for RPFs
- Identification of a Hotspot Region for Each RPF
- Selection of the RPF with the Most Severe Hotspot Region
- Optimization of a Power Network According to the Selected RPF
- Fast Update of VVMs for all RPFs
- Any IR-Drop Violation?
  - YES
  - NO

Power Network Output
Select a RPF to optimize power network according to the hotspot regions in different RPFs $f_i$’s.

Use the following function to determine the priority of a $f_i$:

$$
\Psi(f_i) = \sum_{g=1}^{h} \varphi_g \ast v^i_g
$$

- $v^i_g$ denotes IR-drop violation value of grid $g$ in a RPF $f_i$.
- $\varphi_g = \begin{cases} 
1, & \sigma \leq 1 \\
10, & 1 < \sigma \leq 3, \text{ violation grid } g \text{ repeatedly appears in the window of } \sigma \text{ RPFs.} \\
100, & \sigma > 3 
\end{cases}$
- The red region means violation repeatedly in the different RPFs.

(a) IR-drop Violation Region of RPF1

(b) IR-drop Violation Region of RPF2

(c) IR-drop Violation Region of RPF3
Optimization of a Power Network According to RPFs

- Initialization of VVMs for RPFs
- Identification of a Hotspot Region for Each RPF
- Selection of the RPF with the Most Severe Hotspot Region
- Optimization of a Power Network According to the Selected RPF
- Fast Update of VVMs for all RPFs
- Any IR-Drop Violation?
  - YES
  - NO
  - Power Network Output
Propose an iterative method to adjust the widths and lengths of VPSs in a region as follows:

- Compute the equivalent VPS resistance of the region with the worst voltage violation, and compute the IR-drop violation ratio for each power stripe in the region.
- Fix the width of each VPS to a larger ESW (effective stripe width) with respect to the current width ESW in the table constructed in advance.
- Adjust length of VPSs according to its equivalent resistance and IR-drop ratio.

![VPSs Sizing Table](image)
Optimization of a Power Network According to RPFs

- Initialization of VVMs for RPFs
- Identification of a Hotspot Region for Each RPF
- Selection of the RPF with the Worst Hotspot Region
- Optimization of Power Network According to the Selected RPF
- Fast Update of VVMs for all RPFs
- Any IR-Drop Violation?
  - YES
  - NO
- Power Network Output
Apply an efficient node based approach to update voltages of those nodes which are close to \( node_i \) if we increase the VPS area at \( node_i \).

- It is quite time consuming to update the voltage of the entire power network.
- The computation can be limited to a small region because the change at \( node_i \) will propagate out and vanish after some distance.
Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion
Experimental Results

Environments:

<table>
<thead>
<tr>
<th>Programming Language</th>
<th>C++</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux Workstation</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Intel® Xeon® E5500 2.27GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>90GB</td>
</tr>
<tr>
<td>System</td>
<td>Cent OS 5.1</td>
</tr>
</tbody>
</table>

Our benchmarks are based on real designs from Himax.

<table>
<thead>
<tr>
<th>Cir.</th>
<th># of Cells</th>
<th># of Mac.</th>
<th>Supply Voltage (V)</th>
<th>Static Power (W)</th>
<th>Max. Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cir1</td>
<td>672952</td>
<td>222</td>
<td>1.10</td>
<td>0.033</td>
<td>0.0789</td>
</tr>
<tr>
<td>Cir2</td>
<td>193688</td>
<td>277</td>
<td>0.81</td>
<td>0.503</td>
<td>0.7802</td>
</tr>
<tr>
<td>Cir3</td>
<td>5909306</td>
<td>874</td>
<td>1.10</td>
<td>0.903</td>
<td>1.3270</td>
</tr>
</tbody>
</table>
Compare with other approaches including average PF and top five worst PFs, and our K-clustering algorithm obtains the smallest “Total V.”.

“Total V.” denotes the ratios of violated PFs to the total PFs.

“Total V.” by our approach are about 1/11 and 1/6 with respect to the average PF and the top five worst PF, respectively.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Average PF Optimization</th>
<th>Top Five Peak PF Optimization</th>
<th>Selected RPF Optimization (K = 10)</th>
<th>Selected RPF Optimization (K = 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Area (10^6 ( \mu \text{m}^2 ))</td>
<td>Max IR drop (mV)</td>
<td>Total V. (%)</td>
<td>Op. (s)</td>
</tr>
<tr>
<td>Cir1</td>
<td>4.85</td>
<td>68.4</td>
<td>45.7</td>
<td>29</td>
</tr>
<tr>
<td>Cir2</td>
<td>25.9</td>
<td>49.7</td>
<td>78.0</td>
<td>48</td>
</tr>
<tr>
<td>Cir3</td>
<td>89.2</td>
<td>65.1</td>
<td>68.2</td>
<td>753</td>
</tr>
<tr>
<td>Nor.</td>
<td>0.97</td>
<td>1.07</td>
<td>11.4</td>
<td>0.54</td>
</tr>
</tbody>
</table>
The numbers of PFs which violate IR-drop constraint in different approaches for each circuit.

- Total PFs
- Optimize Average PF
- Optimize top 5 Peak PFs

(K=10) Op. RPFs
(K=5) Op. RPFs

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Total PFs</th>
<th>Optimize Average PF</th>
<th>Optimize top 5 Peak PFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cir1</td>
<td>410</td>
<td>187</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>Cir2</td>
<td>141</td>
<td>110</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>Cir3</td>
<td>173</td>
<td>118</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>14</td>
</tr>
</tbody>
</table>
Compare K-clustering with K-means, which is the most classic clustering algorithm.

Modify K-clustering which does not need to specify number of clusters K in advance.

Compare two algorithms using different cost functions.

\[ \text{diff}(i, j) = \frac{1}{h} \sum_{b=0}^{h} |x^i_b - x^j_b| \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cost</th>
<th>K-means</th>
<th>K-clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cir1</td>
<td>Area (10^6 \text{ um}^2)</td>
<td>Total Vio.(%)</td>
<td>Total Time(s)</td>
</tr>
<tr>
<td>Cir2</td>
<td>4.92</td>
<td>4.7</td>
<td>583</td>
</tr>
<tr>
<td>Cir3</td>
<td>26.2</td>
<td>5.3</td>
<td>527</td>
</tr>
<tr>
<td>Nor.</td>
<td>92.0</td>
<td>7.6</td>
<td>3568</td>
</tr>
<tr>
<td></td>
<td>0.98</td>
<td>16.8</td>
<td>1.43</td>
</tr>
</tbody>
</table>
Compare our window based sizing algorithm with other approach including iterative methodology and sequential linear programming methodology (SLP for short) [1].

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Original Status</th>
<th>Iterative method</th>
<th>SLP method</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Area (10^6 \text{um}^2)</td>
<td>Total OV.</td>
<td>Max. IR drop (mV)</td>
<td>Area (10^6 \text{um}^2)</td>
</tr>
<tr>
<td>Cir1</td>
<td>4.59</td>
<td>398</td>
<td>67.1</td>
<td>5.53</td>
</tr>
<tr>
<td>Cir2</td>
<td>24.9</td>
<td>8536</td>
<td>52.6</td>
<td>35.8</td>
</tr>
<tr>
<td>Cir3</td>
<td>86.5</td>
<td>10267</td>
<td>66.9</td>
<td>137.2</td>
</tr>
<tr>
<td>Nor.</td>
<td>0.92</td>
<td>0.96</td>
<td>1.11</td>
<td>1.30</td>
</tr>
</tbody>
</table>

Experimental Results

Cluster 1
Cluster 2
Cluster 3
RPF of cluster 1
IR-drop of RPF 1
RPF of cluster 2
IR-drop of RPF 2
RPF of cluster 3
IR-drop of RPF 3

Cir 2

Legend:
- 7
- 6
- 5
- 4
- 3
- 2
- 1

Color Scale:
0% - 10%
Experimental Results

Cir 1

RPF1 before
RPF1 after
RPF2 before
RPF2 after
RPF3 before
RPF3 after

Cir 2

RPF1 before
RPF1 after
RPF2 before
RPF2 after
RPF3 before
RPF3 after
Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion
Propose a power network optimization based on clustering algorithm for dynamic IR-drop.

Propose an efficient method to resolve dynamic IR-drop violations for industrial cases.
- Construct PDMs from a VCD file and classify them into several categories through a clustering algorithm.
- Find a representative power consumption file in each cluster.
- Optimize power network by resizing the VPSs in the most severe voltage violation region in each RPF.

The experimental results have demonstrated that our approach can greatly reduce the ratio of violated power profiles to total critical power profiles compared to intuitive approaches.
Thank You For Your Attention