Transition-Aware Decoupling-Capacitor Allocation in Power Noise Reduction

Po-Yuan Chen  
Department of Computer Science  
National Tsing Hua University  
HsinChu, Taiwan 300  
Email: pychen@cs.nthu.edu.tw

Che-Yu Liu  
Department of Computer Science  
National Tsing Hua University  
HsinChu, Taiwan 300  
Email: g9562529@oz.nthu.edu.tw

TingTing Hwang  
Department of Computer Science  
National Tsing Hua University  
HsinChu, Taiwan 300  
Email: tingting@cs.nthu.edu.tw

Abstract—Dynamic power noises may not only degrade the circuit performance but also reduce the noise margin which may result in the functional errors in integrated circuit. Decoupling capacitor (decap) allocation is one of the most effective way in reducing serious dynamic power noises (hotspots). To allocate decap before placement, we observed that not only locations but also rising time of functional cells are required to accurately predict power noises. Compared to a previous work which only takes neighborhood relation into consideration, our method is more efficient in reducing hotspots. Furthermore, to reduce the hotspots after placement, instead of only using the empty space as proposed in the previous work, we move out cells in the area with serious power noise area (hot area). The obtained empty space can be used to accommodate decaps to further reduce the hotspots. The experimental result shows, compared to the previous work [1], our estimation function to allocate decap before placement is 23% better in reducing power noises. Moreover, compared to a method which fills decaps to all remaining empty space, our cell move algorithm can almost eliminate all the remaining hot grid nodes and hot cells. In summary, compared to the original circuits (without decap), about 60% of hotspots can be removed using our prediction function before placement, and most of the remaining hotspots are removed by our cell moving step after placement.

I. INTRODUCTION

In modern VLSI design, power supply networks are used to provide reliable operating of voltage. As fabrication technology scale progresses and nominal supply voltage decreases, dynamic power noises become an important issue in power supply network design. The dynamic power noises may not only degrade the circuit performance [1], [2], [5] but also reduce the noise margin which may result in functional errors in integrated circuit [4], [5]. Therefore, to keep the reliability of power supply network, reducing dynamic power noises is important.

Decoupling capacitor (decap) allocation is widely used to reduce dynamic power noises [1], [2], [6]. Decaps can reserve electronic charge and release it while cells make switches. Therefore, the drops on power supply networks can be reduced. However, the allocation effort is effective only when the decaps are placed near the hotspot cells (cells that have the most serious dynamic power noises). Previous work on placing decap cells can be categorized into two types. The first one is performed after placement [2], [4]. Since IR drop on the supply is analyzed after cells are placed, more accurate calculation of power noises can be obtained. However, the drawback of this approach is that large decap is hard to be reallocated after placement. The most suitable location for decap cells may not be empty, and thus cells (including decap and functional cells) may be moved far away from their optimal positions. The second one is to determine which cells decoupling cells are bound to before placement [1]. Then, as the second step, cells are only fine tuned to compensate the inaccuracy of prediction after placement. The challenge of this before placement method is to predict hotspot cells accurately so that the number of cells moved is as few as possible. In this paper, the second approach will be taken.

A previous work [1] taking the before placement approach was proposed. It has shown 19% more efficient on reducing dynamic power noises as compared to the method which distributes decaps evenly to cells. To determine the quantity of bound decap for each cell, the authors predict the neighborhood current consumption (NCC). To compute the NCC of each cell, they first compute the mutual contraction value for each connection before placement. Then, the connections whose mutual contraction value is among top 30% are classified as strong connections. At last, the NCC value for each cell is computed taking into consideration neighboring cells with strong connections and neighboring cells’ switching current consumption. The cell with large NCC value will be bound with large decap. In essence, presumed locations of cells are used to allocate decap. We observe NCC value is not adequate to model the phenomenon of dynamic power noises. The worst case of dynamic power noises occur at the beginning of a clock cycle. For instance, we conduct an experiment of the number-of-hotspots distribution on benchmark circuit s9234 (in ISCAS89 benchmark set) by the vectorless approach suggested in [7], [8] which has been proven to be efficient in evaluating leakage current and dynamic power noises in the worse case. In the vectorless approach, a clock cycle is divided into several time intervals. For each time interval, the modified nodal analysis (MNA) [9] is applied to evaluate the dynamic power noises. Figure 1 shows the result. The horizontal axis shows the time intervals in a clock cycle and the vertical axis shows the distribution of the number-of-hotspots in a clock cycle. Here, a hotspot is defined as a placed cell whose power noise is larger than 5% of Vdd. From this figure, we observe that a large number of hotspots occur at the beginning of a clock cycle. Therefore, timing information should be taken into consideration in predicting power noises.

In this paper, we will propose a more accurate prediction function to bind decap to functional cells before placement. Then, as the second step, we propose a cell moving method to move cells from hot area which suffers serious dynamic power noises. Next, the empty space obtained by moving cells can accommodate decaps to reduce the dynamic power noises.

The organization of this paper is as follows. In Section II, the modeling and analysis of power supply network are presented. In Section III and Section IV, design flow and algorithms are discussed. Experimental result is shown in Section V. Finally, conclusion is made in Section VI.

II. MODELING AND ANALYSIS OF POWER SUPPLY NETWORK

In our experiment, we divide the chip core into several blocks and each block corresponds to a grid node [1]. The discharge current of each gate is modeled as a triangular waveform and the power grid network is as a RC network where net resistors and net capacitance...
are included. To calculate the dynamic power noises, in each block, we lump the decoupling capacitors as single capacitor and connect it to the grid node. In addition to the decoupling capacitors, we also model the discharge current of each gate as a current source and lump them as a single current source connecting to the grid node similar to [1].

The modified nodal analysis (MNA) [9] is adopted to calculate the dynamic power noises. After applying the Backward Euler technique, the relationship of components are modeled as follows:

\[ G + \frac{C}{h}v[k] = I[k] + \frac{C}{h}v[k - 1] \]

where \( G \) is the conductance matrix, \( C \) capacitance, \( v \) a vector of nodal voltages, \( k \) the \( k \)th time interval, \( I \) the vector of current sources, and \( h \) the time for transient analysis.

III. DESIGN FLOW

Figure 2 presents our design flow. The input is a synthesized circuit. We propose two algorithms in the design flow. The first one, Decap Padding, is performed before placement, and the second, Cell Moving, is after placement. In the first step, based on our observation, we will develop a new decap padding function to determine the quantity of decap bound to each cell. Since decaps are bound with cells, they are placed near the corresponding cells after placement. By this method, dynamic power noises can be effectively reduced. Although the area of decaps is reserved before placement, hotspots may still exist after placement. Therefore, we propose in the third step to further eliminate hot area by utilizing more accurate after-placement information. The main idea of the third step is to move cells from hot area. Then, the empty space obtained by moving cells can be further allocated to decaps, and hence the dynamic power noises in the hot area are reduced.

IV. ALGORITHMS

In this section, Decap Padding and Cell Moving algorithms will be presented in following subsections.

A. Decap Padding

By the experimental result in Figure 1, we observe a large number of hotspots occurs at the beginning of a clock cycle. To accurately pad Decaps to cells, we apply another experiment on circuit s9234 (in ISCAS89 benchmark set). Similar to the experiment in Figure 1, this experiment divides a clock cycle into several time intervals. Then, the number of rising cells in each time interval in the worst case is calculated by the vectorless approach in [7]. The experimental result is shown in Figure 3. The horizontal axis shows the time intervals in a clock cycle and the vertical axis shows the number of rising cells in each time interval. By Figures 1 and 3, we observe the hotspots happen while a large number of cells rise simultaneously.

In the previous work [1], neighboring cells with strong connections and neighboring cells’ switching current consumption (computed as switching probability × output loading) are used to determine the amount of decoupling capacitor bound to a cell. Since cells are assumed to make switching at the same time in one clock cycle, only location of cells are taken into consideration to reduce dynamic power noises in [1]. However, if a clock cycle is carefully examined, we found that cells make transitions at different intervals. Therefore, except NCC values, rising interval of cells in a clock cycle should be taken into consideration to allocate decaps. Furthermore, by these two figures, we also observe that the relationship between the number of hotspots and the number of rising cells is not linear. As the number of rising cells decreases, the number of the hotspots decreases dramatically as shown in these two figures. Hence, the methodology to distribute decaps to cells should be in an exponential manner. For a more accurate padding, we first define the weight for each time interval as follows:

\[ \text{interval}_\text{weight}_i = (\text{rising}_\text{cells}_\text{number})^{exp} \]  

where \( \text{interval}_\text{weight}_i \) is the weight of the \( i \)th time interval, \( \text{rising}_\text{cells}_\text{number} \) is the number of rising cells in the \( i \)th time interval, and \( \text{exp} \) is a user-specified weight. Next, for each cell, the \( \text{tran}_\text{weight} \) of cell \( c \) is defined by:

\[ \text{tran}_\text{weight}_j = \sum_{i=1}^{\text{time}_\text{interval}} W(i, j) \]  

and

\[ W(i, j) = \begin{cases} \text{interval}_\text{weight}_i \text{ if cell } j \text{ rises in time}_\text{interval}, & \text{otherwise} \\ 0 & \end{cases} \]

where \( \text{time}_\text{interval} \) is the number of total time intervals. For instance, Figure 4(a) shows an example circuit composed of seven cells. The primary inputs in the example circuit are \( P_0 \) to \( P_4 \), and the primary outputs are \( P_5 \) to \( P_6 \). For simplicity, we assume, in this example, the rising and falling time of each pin in a cell are the same, and the delay of each pin in a cell is also the same. The number inside the cell represents the cell delay, and the cell name is above the cell. Figure 4(b) shows the possible rising cells in each time interval after applying vectorless approach where the length of a time interval is set to be one time unit. Note that a cell may rise in several time intervals. For example, for path \( P_3-C_6 \), \( c_6 \) rises in the 2th time interval, and for path \( P_1-C_3, c_3 \) rises in the 3th one. From this timing bar shown in Figure 4(b) and \( \text{exp} \) in Equation (1) being set to be 2, the \( \text{interval}_\text{weight}_i \) for \( i = 1 \) to 5 are \( 9(3^2), 4(2^2), 4(2^2), 1(1^2), \) and \( 1(1^2) \), respectively. Then, the \( \text{tran}_\text{weight}_j \) for a cell \( c \) can be calculated. For instance, for cell \( c_6 \), because \( c_6 \) rises in time intervals 2 and 3, \( \text{tran}_\text{weight}_6 = \text{interval}_\text{weight}_2 + \text{interval}_\text{weight}_3 = 8 \). Similarly, we can compute \( \text{tran}_\text{weight} \) for all cells and \( \text{tran}_\text{weight}_j \)s for \( j = 1 \) to 7 are \( 9, 9, 9, 4, 4, 8, \) and 2, respectively.

After defining the cost function taking into consideration the transition time of a cell, we now define a function to predict the amount of power noises of a cell. Since both \( \text{trans} \) and \( \text{location} \) are two important factors to induce power noises, we should not ignore the effect of location. Hence, the concept of strong connections borrowed from [10] is utilized. The strong connections are links whose two ends (cells) are predicted to be close after placement. To collect the strong connections, the connecting weight of each link is defined by following steps. First, if a net \( k \) connects \( d(k) \) nodes,
Fig. 4. (a) the example circuit (b) the rising gates in time intervals

each link \((u, v)\) in the net is assigned a weight by:

\[
link_{\text{weight}}(u, v) = \frac{2}{d(k) \times (d(k) - 1)}
\]

Then, the normalized link weight of link \((u, v)\), \(nlink_{\text{weight}}(u, v)\), is defined as:

\[
nlink_{\text{weight}}(u, v) = \frac{link_{\text{weight}}(u, v)}{\sum_x link_{\text{weight}}(u, x)}
\]

where \(\sum_x (u, x)\) is the sum of all \(link_{\text{weights}}\) of links incident to \(u\). Finally, the mutual contraction \(MC\) for link \((u, v)\) is computed by

\[
MC(u, v) = nlink_{\text{weight}}(u, v) \times nlink_{\text{weight}}(v, u)
\]

In [10], mutual contraction has been proven to be able to predict the wire length before placement. The larger the value of mutual contraction between two nodes is, the shorter the wire length is. Next, strong connections are links whose mutual contraction values are top 30\% among all links as defined in [1]. The strong connections are predicted to have short lengths. Then, we define the neighboring transition weight, \(ntw\), for \(cell_i\) as:

\[
ntw_j = \sum_{s \in nset(j)} \text{tran}_{\text{weight} s} \left( \frac{\text{trans}_{\text{weight} s}}{|nset(j)|} \right)
\]

where \(nset(j)\) is a collected set of neighbors linked by strong connections to \(cell_i\) and \(|nset(j)|\) is the size of this collected set. \(\text{trans}_{\text{weight} s}\) is the transition weight of \(cell_s\) in \(nset(j)\).

Then, our \(\text{decap}_{\text{weight}}\) for any \(cell_i\) is:

\[
de\text{cap}_{\text{weight}}(j) = \alpha \times \text{trans}_{\text{weight} j} + \beta \times ntw_j
\]

where \(\alpha\) and \(\beta\) are specified by designer. \(\text{trans}_{\text{weight}}\) is calculated by Equation (2) and \(ntw\) is calculated by Equation (3). Finally, the quantity of bound \(\text{decap}\) for \(cell_i\) is:

\[
\text{decap}(j) = \text{total}_{\text{decap area}} \times \frac{\text{decap}_{\text{weight}}(j)}{\sum_{z=1}^{\text{cells}} \text{decap}_{\text{weight}}(z)}
\]

where \(\text{total}_{\text{decap area}}\) is the area of bound decaps which is set to 20\% of total cell area in this paper as [1].

B. Cell Moving

In subsection IV-A, we propose a method to pad \(\text{decaps}\) to the cells before placement. Although the area of \(\text{decaps}\) is reserved before placement, \(\text{hotspots}\) may still exist after placement. Therefore, in this subsection, we propose a method to further eliminate hot area after placement.

At first, since cells are already placed, more accurate timing model such as half-perimeter wirelength [11] on connecting wires can be used. With more accurate timing information, dynamic power noises are analyzed. Then, \(\text{hot block list}\) is constructed. Each element in the list is modeled as \(\text{hot block}\), where \(i\) represents the block number in the circuit. \(\text{hot block}\) gives the value of maximum dynamic power noises of the \(i\)th block among all time intervals. The \(\text{hot block list}\) is sorted in decreasing order. Next, our algorithm will process each \(\text{hot block}\) sequentially. If the dynamic power noises of the current \(\text{hot block}\) is above a user-specified threshold, \(\text{Cell Moving}\) step will start moving out cells in the block to other appropriate block until the ratio of \(\text{current sum to decap area}\) is smaller than the threshold where \(\text{current sum}\) is the lumped value of total current sources in the block at the time interval in which maximum power noises occur and \(\text{decap area}\) is the area of total \(\text{decaps}\) in the block.

The reason behind this threshold is the assumption that the ratio of maximum current to \(\text{decap}\) is an effective index to power noises. The cell moving step is described as follows.

First, a \(\text{hot cell list}\) is constructed for current \(\text{hot block}\). The cells in \(\text{hot cell list}\) are those cell located in \(\text{block}\), and rising in time intervals with serious power noises. The cells in \(\text{hot cell list}\) are sorted by slack for current \(\text{hot block}\), in decreasing order. Then, the algorithm will select the first cell in \(\text{hot cell list}\) for process. The reason of this selection is that a cell with large slack is more flexible to be moved. After the selecting, the selected cell is removed from \(\text{hot cell list}\). The blocks that surrounds the current \(\text{hot block}\) will be the candidate-destination blocks. The destination block must satisfy two constraints. The first one is the empty area in the this block must larger than the area required by the moved cell. The second one is that the block cannot be a \(\text{hot block}\) at any time interval that the selected cell rises. The block which violates either one of the two constraints will be pruned. The rest of the blocks are collected in \(\text{candidate list}\) and sorted by \(\text{hot values}\).

The \(\text{hot value}\) of each block is computed by two numbers. The first one is the number of time intervals in which the block is a \(\text{hot block}\) and the second one is the number of total time intervals. A small \(\text{hot value}\) means that the block is not hot in most of time intervals. For example, if a block become \(\text{hot block}\) in 3 time intervals and the total number of intervals is 10. The \(\text{hot value}\) of this block is \(\frac{3}{10}\). The \(\text{hot values}\) in \(\text{candidate list}\) is sorted in increasing order. Then, \(\text{Cell Moving}\) step will consider to move the hot cell to the block with the least \(\text{hot value}\). If the timing of the circuit is kept after moving cell, the cell is moved to the first block and the timing information is updated. Otherwise, the next block in the \(\text{candidate list}\) is selected.

V. EXPERIMENTAL RESULT

The experiments are conducted using ISCAS89 benchmark circuits. The switching current of each cell is obtained by Hspice and modeled as a triangular waveform. First, all benchmark circuits are synthesized with TSMC .0.13µm cell library by Design Compiler. After synthesis, the Decap Padding algorithm is applied to bind decaps to functional cells before placement. Then, benchmark netlist is placed by SOC Encounter and power noises analysis is performed after placement. Finally, the Cell Moving algorithm with timing and power noise information is applied to further reduce hotspots.

Table I shows the characteristics of the benchmark circuits. The names of the benchmark circuits are listed in the first column. The second and third columns report the number of cells and cell area in \(\mu m^2\) after synthesis in each benchmark circuit.

<table>
<thead>
<tr>
<th>Circuits</th>
<th># of cells</th>
<th>cell area ((\mu m^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>79214</td>
<td>2022</td>
<td>26035</td>
</tr>
<tr>
<td>535032</td>
<td>12032</td>
<td>164254</td>
</tr>
<tr>
<td>38417</td>
<td>11635</td>
<td>162838</td>
</tr>
<tr>
<td>383848</td>
<td>13956</td>
<td>182786</td>
</tr>
</tbody>
</table>

TABLE I

BENCHMARK CIRCUITS
whose area is less than one Decap, will not be allocated as Decaps. In this experiment, exp in Equation (1) is set to 1.8, and and in Equation (4) are 0.7 and 0.3, respectively. The experimental results of original circuits without Decap allocation, NoDecap, are also reported as baseline for comparison. Table II shows the experimental result. The third, forth, and fifth columns report the value of maximum dynamic power noises, the number of hot grid nodes, and the number of hot cells. Here, hot grid nodes and hot cells are defined as those grid nodes and cells suffering the power noises larger than 5% of Vdd. The last row reports the normalized average value to the one of NoDecap. By this experiment, compared to WGT, our method is 7% more efficient in reducing maximum power noises, 24% and 23% better in reducing the number of hot grid nodes and the number of hot cells, respectively. Furthermore, compared to NoDecap in average, the resultant maximum power noises of our method is only 57% and the ratios of hot grid nodes and of hot cells are only 42% and 40%, respectively. That is, about 60% of hotspots of original circuit is removed before placement.

In the second experiment, we compare our result after performing Decap_Padding and Cell_Moving to the ones of Baseline and AllDecap. Baseline is the method only performing Decap_Padding algorithm, and AllDecap is that fills Decaps to all remaining empty space after performing Decap_Padding algorithm. In order to prove the efficiency of our Cell_Moving algorithm, after performing Cell_Moving, we also fill Decaps to all remaining empty space. The threshold of the area of Decaps in Cell_Moving algorithm is set to the average ratio of Decap_Padding for the non-hot blocks whose maximum power noises are within 1% to 3%. Table III shows the experimental result. By this experiment, compared to AllDecap, Cell_Moving is 12% more efficient in reducing maximum power noises, and it almost eliminates all hot grid nodes and hot cells. Our Cell_Moving is specially important when there is not enough empty space in a hot block. For example, in benchmark s38417, after AllDecap method is performed, there are still 68 hot grid nodes. However, by our Cell_Moving step, there is no hot grid node.

To analyze the efficiency of Cell_Moving algorithm, the third experiment is conducted to observe average power noises in hot blocks. In this experiment, all hot blocks in Baseline are marked, and average values of power noises in these blocks are reported. Then, the average of power noises in these blocks after performing AllDecap and Cell_Moving are also reported. The result is shown in Figure 5. The horizontal axis shows benchmark circuits, and the vertical axis reports the average power noises values among hot blocks. By this experiment, we understand that the average amount of power noises reduction by our method is higher than the one by AllDecap methods because we judiciously move active cells out from and allocate decaps to hot blocks in Cell_Moving step for each hot block. The power distribution networks become even. That is why we can reduce more power noises. By these experiments, we have shown that our algorithms is very efficient in reducing dynamic power noises.

VI. CONCLUSIONS

In this paper, we proposed two algorithms, Decap_Padding and Cell_Moving. The first algorithm, Decap_Padding, predict the hotspot cells and bind Decaps to those cells. The second algorithm, Cell_Moving, move cells out from hot blocks to further reduce hotspots. The experimental result shows, compared to the previous work [1], our estimation function to allocate decap before placement is 23% better in reducing power noises. Moreover, compared to a method which fills decaps to all remaining empty space, our Cell_Moving algorithm can almost eliminate all hot grid nodes and hot cells.

REFERENCES