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 ($hotspot$s). 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 $hotspot$s. Furthermore, to reduce the $hotspot$s $after$ $placement$, 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 $decap$s to further reduce the $hotspot$s. 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 $decap$s 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 $hotspot$s can be removed using our prediction function before placement, and most of the remaining $hotspot$s 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], [3] 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]. $Decap$s 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 $decap$s 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 $decap$s 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-$hotspot$ 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-$hotspot$s 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 $hotspot$s occur at the beginning of a clock cycle. Therefore, timing information should be taken into consideration in predicting power noises.

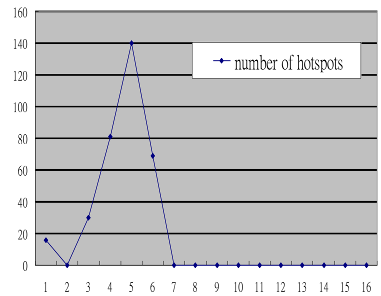

Fig. 1. The number-of-$hotspot$s distribution in one clock cycle

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 $decap$s 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 resisters and net capacitance

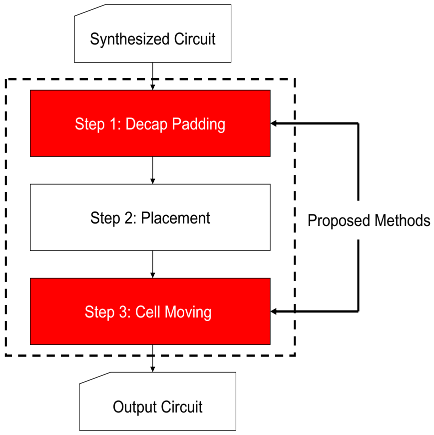

Fig. 2. Design Flow

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 kth 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 $decap$s 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 $decap$s is reserved before placement, $hotspot$s may still exist after placement. Therefore, we propose in the third step to further eliminate hot area by utilizing more accurate $after\text{-}placement$ 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 $decap$s, 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 $hotspot$s occurs at the beginning of a clock cycle. To accurately pad $Decap$s 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 $hotspot$s 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

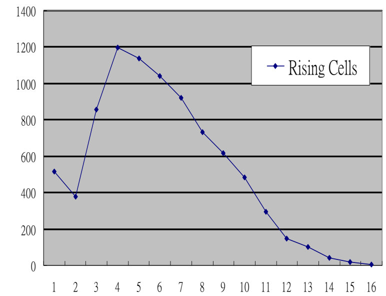

Fig. 3. The number of rising cells in time intervals

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 $decap$s. Furthermore, by these two figures, we also observe that the relationship between the number of $hotspot$s and the number of rising cells is not linear. As the number of rising cells decreases, the number of the $hotspot$s decreases dramatically as shown in these two figures. Hence, the methodology to distribute $decap$s to cells should be in an exponential manner. For a more accurate padding, we first define the weight for each time interval as follows:

$$interval\_weight_i = (rising\_cells\_number_i)^{exp} \tag{1}$$

where $interval\_weight_i$ is the weight of the ith time interval, $rising\_cells\_number_i$ is the number of rising cells in the ith time interval, and exp is a user-specified weight. Next, for each $cell_j$, the $tran\_weight_j$ of $cell_j$ is defined by:

$$tran\_weight_j = \sum_{i=1}^{|time\_interval|} W(i,j) \tag{2}$$

and

$$W(i,j) = \begin{cases} interval\_weight_i & if\ cell_j\ rises\ in\ time\_interval_i \\ 0 & otherwise \end{cases}$$

where $|time\_interval|$ 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_3, and the primary outputs are P_4 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_2-c_3-c_6, c_6 rises in the 3th one. From this timing bar shown in Figure 4(b) and exp in Equation (1) being set to be 2, the $interval\_weight_i$ 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 $tran\_weight_j$ for a $cell_j$ can be calculated. For instance, for cell c_6, because c_6 rises in time intervals 2 and 3, $tran\_weight_6 = interval\_weight_2 + interval\_weight_3 = 8$. Similarly, we can compute $tran\_weight$ for all cells and $tran_weight_j$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 $timing$ and $location$ 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,

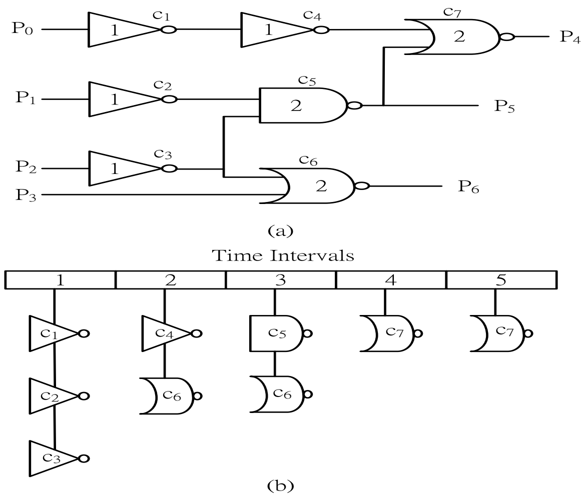

Fig. 4. (a) the example circuit (b) the rising gates in time intervals

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

$$link\_weight(u,v) = \frac{2}{d(k) \times (d(k) - 1)}$$

Then, the normalized link weight of link (u, v), $nlink\_weight(u, v)$, is defined as:

$$nlink\_weight(u,v) = \frac{link\_weight(u,v)}{\sum_x link\_weight(u,x)}$$

where $\sum_x (u, x)$ is the sum of all $link_weight$s of links incident to u. Finally, the mutual contraction MC for link (u, v) is computed by

$$MC(u,v) = nlink\_weight(u,v) \times nlink\_weight(v,u)$$

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_j, for $cell_j$ as:

$$ntw_j = \frac{\sum_{z \in n\_set(j)} tran\_weight_z}{|n\_set(j)|} \quad (3)$$

where $n\_set(j)$ is a collected set of neighbors linked by strong connections to $cell_j$ and $|n\_set(j)|$ is the size of this collected set. $tran\_weight_z$ is the transition weight of a $cell_z$ in $n\_set(j)$.

Then, our $decap$ weight for any $cell_j$ is:

$$decap\_weight(j) = \alpha \times tran\_weight_j + \beta \times ntw_j \quad (4)$$

where α and β are specified by designer. $tran\_weight_j$ is calculated by Equation (2) and ntw_j is calculated by Equation (3). Finally, the quantity of bound $decap$ for $cell_j$ is:

$$decap(j) = total\_decap\_area \times \frac{decap\_weight(j)}{\sum_{z=1}^{|cells|} decap\_weight(z)}$$

where $total\_decap\_area$ is the area of bound $decap$s which is set to be 20% of total cell area in this paper as [1].

B. Cell Moving

In subsection IV-A, we propose a method to pad $decap$s to the cells before placement. Although the area of $decap$s is reserved before placement, $hotspot$s 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 informatin, dynamic power noises are analyzed. Then, hot_block_list is constructed. Each element in the list is modeled as hot_block_i, where i represents the ith block in the circuit. hot_block_i gives the value of maximum dynamic power noises of the ith block among all time intervals. The hot_block_list is sorted in decreasing order. Next, our algorithm will process each hot_block sequentially. If the dynamic power noises of the current hot_block is above a user-specified threshold, $Cell\_Moving$ step will start moving out cells in the block to other appropriate block until the ratio of $current\_sum$ to $decap\_area$ is smaller than the threshold where $current\_sum$ is the lumped value of total current sources in the block at the time interval in which maximum power noises occur and $decap\_area$ is the area of total $decap$s in the block. The reason behind this threshold is the assumption that the ratio of maximum current to $decap$ is an effective index to power noises. The cell moving step is described as follows.

First, a hot_cell_list is constructed for current hot_block_i. The cells in hot_cell_list are those cell located in $block_i$ and rising in time intervals with serious power noises. The cells in hot_cell_list are sorted by slack for current hot_block_i, in decreasing order. Then, the algorithm will select the first cell in 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 hot_cell_list. The blocks that surrounds the current 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 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 $candidate\_list$ and sorted by hot_values.

The 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 hot_block and the second one is the number of total time intervals. A small hot_value means that the block is not hot in most of time intervals. For example, if a block become hot_block in 3 time intervals and the total number of intervals is 10. The hot_value of this block is $\frac{3}{10}$. The hot_values in $candidate\_list$ is sorted in increasing order. Then, $Cell\_Moving$ step will consider to move the hot cell to the block with the least 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 $candidate\_list$ 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 $decap$s 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 $hotspot$s.

TABLE I
Benchmark circuits

Circuits	# of cells	cell area (μm^2)
s9234	2022	26033
s13207	3378	49952
s35932	12032	164254
s38417	11633	162838
s38584	13956	182786

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 μm^2 after synthesis in each benchmark circuit.

In the first experiment, $Decap\_Padding$ algorithm is compared to the method, WGT, which also binding $decap$s to cells $before\ placement$ taking neighborhood current consumption (NCC) into consideration in [1]. As in [1], the area of bound $Decap$s is set to be 20% of total cell area. A $Decap$ area is equal to the area of a $INVX1$ cell in TSMC .0.13μm library. Any reserved area

TABLE II

THE COMPARISON OF *before-placement* METHODS

Circuits	Methods	Max. Noise (V)	# of hot grid nodes	# of hot cells
s9234	$NoDecap$	0.172	159	1201
	WGT	0.112	152	1202
	$Ours$	0.131	128	1046
s13207	$NoDecap$	0.088	98	1044
	WGT	0.054	9	96
	$Ours$	0.054	16	161
s35932	$NoDecap$	0.114	305	2744
	WGT	0.056	63	538
	$Ours$	0.064	57	535
s38417	$NoDecap$	0.132	527	5426
	WGT	0.08	313	2991
	$Ours$	0.072	210	1888
s38584	$NoDecap$	0.178	866	7290
	WGT	0.135	757	6259
	$Ours$	0.072	410	3540
AVG	$NoDecap$	1	1	1
	WGT	0.64	0.66	0.63
	$Ours$	0.57	0.42	0.4

TABLE III

THE COMPARISON OF *after-placement* METHODS

Circuits	Methods	Max. Noise	# of hot grid nodes	# of hot cells
s9234	$Baseline$	0.131	128	1046
	$AllDecap$	0.06	19	229
	$Cell\_Moving$	0.055	2	23
s13207	$Baseline$	0.054	16	161
	$AllDecap$	0.051	2	22
	$Cell\_Moving$	0.043	0	0
s35932	$Baseline$	0.064	57	535
	$AllDecap$	0.056	2	17
	$Cell\_Moving$	0.066	3	30
s38417	$Baseline$	0.072	210	1888
	$AllDecap$	0.069	68	720
	$Cell\_Moving$	0.047	0	0
s38584	$Baseline$	0.072	410	3540
	$AllDecap$	0.054	5	46
	$Cell\_Moving$	0.043	0	0
AVG	$Baseline$	1.36	8.55	6.93
	$AllDecap$	1	1	1
	$Cell\_Moving$	0.88	0.05	0.05

whose area is less than one $Decap$, will not be allocated as $Decap$s. In this experiment, exp in Equation (1) is set to 1.8, and α and β in Equation (4) are 0.7 and 0.3, respectively. The experimental results of original circuits without $decap$s 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 noise, 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 $hotspot$s 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 $Decap$s 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 $Decap$s to all remaining empty space. The threshold of $\frac{current\_sum}{decap\_area}$ in $Cell\_Moving$ algorithm is set to the average of $\frac{current\_sum}{decap\_area}$ 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 nodes.

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 repots 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 $decap$s to hot_block 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 $Decap$s to those cells. The second algorithm, $Cell\_Moving$, move cells out from hot_blocks to further reduce $hotspot$s. The experimental result shows, compared to the previous work [1], our estimation function to allocate $decap$ before placement

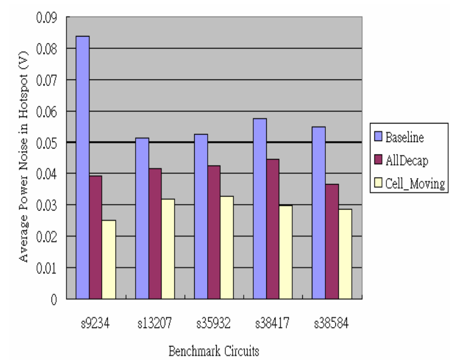

Fig. 5. The average power noises of hot_blocks in each benchmark

is 23% better in reducing power noises. Moreover, compared to a method which fills $decap$s to all remaining empty space, our $Cell\_Moving$ algorithm can almost eliminate all hot grid nodes and hot cells.

REFERENCES

[1] Chao-Yang Yeh and Malgorzata Marek-Sadowska, "Timing-aware Power Noise Reduction in Layout," *International Conference on Computer-Aided Design (ICCAD)*, pp. 627-634, Nov., 2005.

[2] Sanjay Pant and David Blaauw, "Timing-aware Decoupling Capacitance Allocation in Power Distribution Networks," *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 757-762, Jan., 2007.

[3] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda, "Vectorless Analysis of Supply Noise Induced Delay Variation," *International Conference on Computer-Aided Design (ICCAD)*, pp. 184-191 , Nov., 2003.

[4] H. Su, S. S. Sapatnekar, and S. R. Nassif, "Optimal Decoupling Capacitor Sizing and Placement for Standard-cell Layout Designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, issue 4, pp. 428-436, Apr., 2003.

[5] Y. Zhong, and Martin D. F. Wong, "Fast Algorithms for IR Drop Analysis in Large Power Grid," *International Conference on Computer-Aided Design (ICCAD)*, pp. 351-357 , Nov., 2005.

[6] M. Popovich, E. G. Friedman, R. M. Secareanu, and O. L. Hartin, "Efficient Placement of Distributed On-chip Decoupling Capacitors in Nanoscale ICs," *International Conference on Computer-Aided Design (ICCAD)*, pp. 811-816 , Nov., 2007.

[7] A.-C. Hsieh, T.-T. Lin, T.-W. Chang and T. Hwang, "A Functionality Directed Clustering Technique for Low Power MTCMOS Design - Computation of Simultaneously Discharging Current," *ACM Transactions on Design Automation of Electronic Systems*, vol. 12, issue 3, Aug., 2007.

[8] H. Kriplani, F. N. Najm, and I. N. Hajj, "Pattern Independent Maximum Current Estimation in Power and Ground Buses of CMOS VLSI Circuits: Algorithms, Signal Correlations, and Their Resolution," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 998-1012, 1995.

[9] L. T. Pillage, R. A. Rohrer, and C. Visweswariah, "Electronic Circuit and System Simulaton Methods," *McGraw-Hill*, 1995.

[10] Bo Hu and Malgorzata Marek-Sadowska, "Wire Length Prediction Based Clustering and Its Application in Placement," *Design Automatin Conference (DAC)*, pp. 800-805, 2003.

[11] M. Pan, N. Viswanathan, and C. Chu, "An Efficient and Effective Detailed Placement algorithm," *International Conference on Computer-Aided Design (ICCAD)*, pp. 48-55, 2005.