

Module Assignment for Pin-Limited Designs under the Stacked-Vdd Paradigm

Yong Zhan, Tianpei Zhang, and Sachin S. Sapatnekar
 Department of Electrical and Computer Engineering
 University of Minnesota

Abstract—This paper addresses the module assignment problem in pin-limited designs under the stacked-Vdd circuit paradigm. A partition-based algorithm is presented for efficiently assigning modules at the floorplanning level so as to reuse currents between the Vdd domains, and minimize the power wasted during the operation of the circuit. Experimental results on a DLX architecture show that compared with assigning modules to different Vdd rails using a bin-packing technique, the circuit generated by our algorithm has 32% lower wasted power, on average. In addition, experiments on a 3D IC example show that our module assignment approach is equally effective in reducing the power waste in 3D ICs.

I. MOTIVATIONS AND DESIGN CONSIDERATIONS

Due to the increased complexity and elevated power consumption of high performance VLSI circuits, the I/O pin limitation problem has become an important issue in chip design. According to the trend predicted by ITRS [1], one-half to two-thirds of all I/O pins will be dedicated to power delivery in contemporary and future technologies, while at the same time, the total number of package pins does not increase significantly. This situation has led to concerns about the signal integrity of power grids because each power pin must deliver a larger amount of current to the circuit as the chip performance improves, while simultaneously, the current flowing through the power grid will also increase, which worsens the IR and $L\frac{di}{dt}$ noise.

The pin limitation problem is especially pronounced in the emerging 3D IC technology which has appeared recently as a promising alternative to the prevailing 2D IC technology. By stacking multiple dies vertically while reducing the footprint area of each die, the 3D IC technology can significantly reduce the interconnect delays of long wires. However, the pin limitation problem is exacerbated because the reduced footprint area can only accommodate a smaller number of I/O pins.

The stacked-Vdd circuit paradigm proposed in [2] can be used to reduce the number of power pins required by a chip, and therefore, significantly relieve the pin-limitation bottleneck. In this new circuit paradigm, logic blocks are stacked several levels high and power is delivered to the circuit as multiples of the regular supply voltage V_{dd} . Next, the delivered supply voltage is divided into several Vdd domains each of which has a range of V_{dd} , and circuit blocks are distributed to different Vdd domains. Voltage regulators are used to control the voltage levels of internal supply rails.

An example of a two level stacked-Vdd circuit is shown in Fig. 1. The advantage of this new circuit structure is that when logic blocks are stacked n levels high and the current requirements between logic blocks operating in different Vdd domains are balanced, the current flowing through each external power grid would be reduced to $\frac{1}{n}$ of the original value, where the words external power grid refer to a power grid that is connected to power pins, i.e., nV_{dd} and GND rails. Therefore, the number of power pins required to supply current to the chip will be significantly reduced.

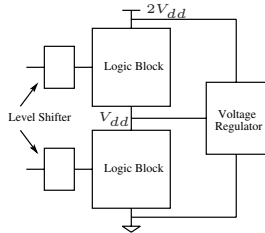


Fig. 1. A schematic of a 2-level stacked-Vdd circuit structure.

An important consideration in the design of a stacked-Vdd circuit is to locally maintain the current balance between logic blocks operating in different Vdd domains because otherwise, the imbalance current will flow through voltage regulators and be wasted¹. Another important issue that has to be considered is the design stage at which the circuit should be partitioned into different Vdd domains. Note that a level shifter is required at the output of a logic block if it is used to drive another logic block operating in a different Vdd domain. Level shifters occupy silicon area and cause extra delays in the circuit. In this paper, we address the module assignment problem

This work was supported in part by DARPA under the 3DIC program.

¹Because the power wasted in a voltage regulator is proportional to the current flowing through it, we will use the words wasted power and wasted current interchangeably in this paper.

at the floorplanning level where the number of modules is usually not very large. Therefore, the performance degradation and the area waste caused by level shifters can be largely ignored.

II. MODULE ASSIGNMENT ALGORITHM

Because the two level stacked-Vdd circuit provides a good tradeoff between chip performance and engineering complexity, it will be the focus of this work. In what follows, we will first present the module assignment algorithm assuming that the floorplan containing both regular modules and voltage regulators is given. The floorplanning process is then briefly described in Section III.

A. Problem formulation

The module assignment problem for a two level stacked-Vdd circuit is formulated as follows:

Given a floorplan including the location and size of each module and voltage regulator, the structure of the power grids, a set of current consumption traces of modules obtained through simulations over a set of benchmark programs, find the assignment of modules to the two different Vdd domains so that the total power wasted by voltage regulators is minimized.

Note that the tapping points of voltage regulators to the internal power grid, i.e., the connecting points between voltage regulators and the V_{dd} rail, have properties that are similar to those of power pins. For well designed regulators, they provide rather stable voltage levels at V_{dd} . In [3], it was demonstrated that each module primarily draws currents from nearby power pins, and the same observation can be applied to the tapping points of voltage regulators. Assume we have K voltage regulators distributed across the chip. Each regulator is represented by the point it taps into the V_{dd} grid. As shown in Fig. 2(a), we can divide the chip into K regions accordingly such that there is one regulator in each region and the i^{th} region contains all the points on chip that primarily draw(sink) currents from(to) the i^{th} regulator. The division of the chip into non-overlapping regions can be achieved through meshing the entire die area using a fine grid and calculating the value of certain metric associated with each grid cell to determine which region it belongs to. In this work, we use the Euclidean distance as the metric, i.e., each cell belongs to the region that is controlled by the nearest voltage regulator.

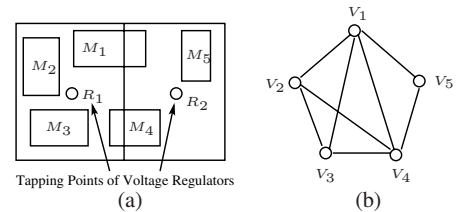


Fig. 2. Graph construction (a) partitioning the chip into disjoint regions each of which is controlled by a voltage regulator, and (b) constructing a graph where node V_i corresponds to module M_i .

After the chip is partitioned into disjoint regions, we can assume that the imbalance current caused by the modules located in a particular region only goes through the regulator in the same region. If a module is located at the boundary between multiple regions, it will be decomposed into several submodules with one submodule in each region it overlaps and with the constraint that all submodules must be assigned to the same Vdd domain.

Let us focus on a particular region corresponding to a particular voltage regulator. Assume the modules located in this region are M_1, M_2, \dots, M_n , where the current flowing through module M_i as a function of time t is given by $I_i(t)$. Because voltage regulators can only respond to the low to mid frequency components of the imbalance currents while the high frequency components are usually handled by on-chip decaps, we pre-process the input current traces obtained through cycle-accurate power simulations to smooth out the high frequency components in the current signals. Therefore, $I_i(t)$ should be understood as containing only the low to mid frequency components of the current flowing through module M_i .

If we associate a 0/1 integer variable x_i with module M_i defined as

$$x_i = \begin{cases} 0 & \text{if } M_i \text{ operates between the } 2V_{dd} \text{ and } V_{dd} \text{ rails} \\ 1 & \text{if } M_i \text{ operates between the } V_{dd} \text{ and } GND \text{ rails} \end{cases} \quad (1)$$

the total current flowing through the voltage regulator at time t will be approximated by

$$I_R(t) = \left| \sum_{i=1}^n I_i(t) \cdot (1 - x_i) - \sum_{i=1}^n I_i(t) \cdot x_i \right| = \left| \sum_{i=1}^n I_i(t) \cdot (1 - 2x_i) \right| \quad (2)$$

The objective is to minimize the average wasted current $\overline{I_R(t)}$. It can be shown that this is a NP-hard problem that can only be solved using heuristics. In our work, instead of minimizing $\overline{I_R(t)}$, we minimize $\overline{I_R(t)^2}$, which can be written as

$$\overline{I_R(t)^2} = \left[\sum_{i=1}^n I_i(t) \right]^2 - 4 \sum_{i < j} \overline{I_i(t)I_j(t)}(x_i + x_j - 2x_i x_j) \quad (3)$$

It is easy to see from (3) that to minimize $\overline{I_R(t)^2}$ is equivalent to maximize

$$S = \sum_{i < j} \overline{I_i(t)I_j(t)}(x_i + x_j - 2x_i x_j) \quad (4)$$

$$\text{and} \quad x_i + x_j - 2x_i x_j = \begin{cases} 0 & \text{if } x_i = x_j \\ 1 & \text{if } x_i \neq x_j \end{cases} \quad (5)$$

The intuition behind (4) and (5) is that if modules M_i and M_j are in different Vdd domains, a positive term $\overline{I_i(t)I_j(t)}$ will appear in summation (4), but not otherwise. Based on this observation, the problem of maximizing S in (4) can be cast into the following equivalent graph partitioning problem.

Given a weighted graph $G = (V, E, W)$ where $V = \{V_1, V_2, \dots, V_n\}$, $E = \{(V_i, V_j) | V_i, V_j \in V\}$, and the weight set $W = \{w(V_i, V_j) = \overline{I_i(t)I_j(t)} | V_i, V_j \in V\}$, find a two way partitioning of the graph so that the total cut of the edges crossing the partition is maximized.

Up to this point, we have been considering one of the K disjoint regions over the chip and the modules that are completely located within it. For the entire chip, a similar graph partitioning problem can be formulated where node V_i in the graph corresponds to module M_i in the layout. The only difference from the problem formulation shown above is in calculating the weight of each edge in the graph. Let S_i represent the area of the i^{th} module, and denote the overlap area between the i^{th} modules and the k^{th} region over the chip by S_{ik} . The weight of edge (V_i, V_j) is calculated by

$$w(V_i, V_j) = \left[\sum_{k=1}^K \frac{S_{ik}S_{jk}}{S_i S_j} \right] \overline{I_i(t)I_j(t)} \quad (6)$$

The intuition behind (6) is that for any pair of modules, only the portions that are located in the same region over the chip count toward the calculation of the correlation between them. A further implication of (6) is that if modules M_i and M_j are completely separated into two disjoint regions, the weight $w(V_i, V_j)$ will be zero, and therefore, the corresponding edge can be removed from the graph. An example of graph construction is shown in Fig. 2(b).

B. Graph partitioning heuristic

A two step heuristic is used to partition the node set of the graph into two subsets so that the total cut is maximized. In the first step, we greedily assign nodes to the subsets so as to obtain a reasonably good initial partition. The primary operations involved in this step include sorting the weighted edges in decreasing weight order and examining them consecutively. For each edge under examination, if none of the two nodes associated with it has been assigned to a partition, we assign them to two different partitions. If one of the nodes has been assigned but not the other, we assign the other node to the opposite partition. Finally, if both nodes have been assigned, we skip the current edge and proceed to the next edge in the sorted edge list.

After the initial node assignment, we use a F-M like algorithm to iteratively improve the partition and increase the cut size. The difference between our algorithm and the conventional F-M algorithm as shown in [4] is that our algorithm maximizes the cut size of a graph where the nodes carry no weight, while the conventional F-M algorithm minimizes the cut size of a graph and balances the node weights between the two partitions.

Of key importance in an F-M like algorithm is the calculation of gains when a node is moved from one partition to the other. In our algorithm, the initial gain of moving a node V_i from its current partition to the opposite partition is given by

$$g(V_i) = \sum_{V_j \in FP(V_i)} w(V_i, V_j) - \sum_{V_j \in TP(V_i)} w(V_i, V_j) \quad (7)$$

where $FP(V_i)$ contains all the nodes that are in the same partition as node V_i and are connected to V_i , and $TP(V_i)$ contains all the nodes that are in

the opposite partition to node V_i and are connected to V_i . For example, in the partition shown in Fig. 3, the initial gain of moving node V_2 from its current partition to the opposite partition is given by $g(V_2) = w(V_1, V_2) - w(V_2, V_4) - w(V_2, V_5)$.

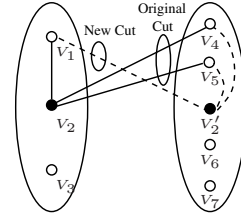


Fig. 3. An example for gain calculation in the F-M like algorithm showing the cut set before and after the node V_2 is moved to the opposite partition.

When a node V_i is moved from one partition to the other, the gains associated with the nodes connected to V_i must be updated. For node V_j that is connected to V_i , the gain associated with V_j should be updated as follows:

- If V_j and V_i were in the same partition before the movement of V_i

$$g(V_j)^{new} = g(V_j)^{old} - 2w(V_i, V_j) \quad (8)$$

- if V_j and V_i were in different partitions before the movement of V_i

$$g(V_j)^{new} = g(V_j)^{old} + 2w(V_i, V_j) \quad (9)$$

The complete F-M like algorithm that is used to iteratively improve the cut size of the partition is listed in Fig. 4.

```

1) do
2)   node_moved ← false;
3)   Calculate the initial gain of each node;
4)   for i ← 1 to number_of_nodes
5)     Select the free node that has the maximum gain
       and call it  $V_{s(i)}$ ;
6)     Lock node  $V_{s(i)}$ ;
7)     Update the gains of the nodes connected to  $V_{s(i)}$ ;
8)   end for;
9)   Find the number  $K$  such that  $G = \sum_{i=1}^K g(V_{s(i)})$ 
       is maximized;
10)  if  $G > 0$ 
11)    Make the  $K$  moves permanent;
12)    node_moved ← true;
13)    Free all locked nodes;
14)  end if;
15) while( node_moved == true );

```

Fig. 4. Iterative improvement of the partition through a F-M like algorithm.

III. FLOORPLANNING AND THE COMPLETE ALGORITHM FLOW

We have used Parquet [5] with a modified cost function to perform the floorplanning. Besides the conventional optimization objectives such as wirelength, an additional objective that is unique to our problem is that the voltage regulators, which are also considered as modules, should be distributed uniformly across the die. This will help reducing the power grid noise in the V_{dd} rail. We have achieved this new objective through modifying the cost function in Parquet. Specifically, a penalty term

$$\text{penalty} = \sum_{i=1}^{K-1} \sum_{j=i+1}^K \frac{1}{d(R_i, R_j)} \quad (10)$$

is added to the cost function used in simulated annealing, where R_i with $i = 1, 2, \dots, K$ represent voltage regulators, and $d(R_i, R_j)$ is the distance between the points where regulators R_i and R_j tap into the V_{dd} grid. The intuition behind (10) is that when regulators R_i and R_j come closer to each other, the penalty term will become larger, and therefore, the corresponding floorplan will have a higher probability of being rejected.

The overall flow of the complete algorithm is given in Fig. 5.

IV. EXPERIMENTAL RESULTS

A. Floorplanning using modified Parquet

Fig. 6(a) shows the floorplan of a microarchitecture used in [6] with ten voltage regulators inserted. The microarchitecture is based on the DLX architecture [7] and the dark regions represent voltage regulators. All modules are assumed to be hard and the floorplanning is performed using Parquet with the modified cost function as described in the previous section. We can see that with the simple modifications we have made to Parquet, the voltage regulators can be reasonably uniformly distributed across the die.

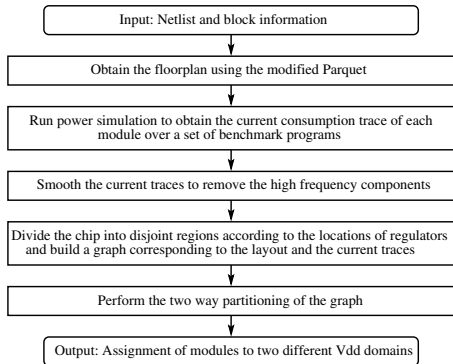


Fig. 5. Complete algorithm for assigning modules to two different Vdd domains.

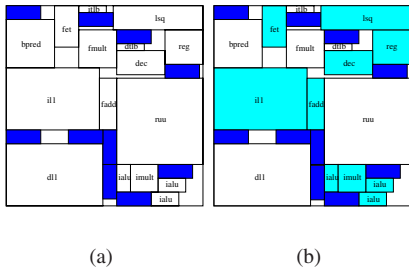


Fig. 6. Floorplan with voltage regulators (a) floorplan (b) assignment of modules to the two different Vdd domains.

B. Result of module assignment

After the floorplan is obtained, we simulate the architecture using the eight benchmark programs contained in the SPEC 2000 suite that cover both integer and floating-point operations, assuming that $V_{dd} = 1.2V$. To speed up the simulations, we utilize SMARTS [8], a periodic sampling technique to obtain a sampled sequence of the average current consumption trace of each module for each of the benchmark programs.

Note that the time averaged total current consumption of the chip may vary significantly while running different programs, and the objective of our algorithm is to obtain a partition of modules that is deemed good across the entire benchmark suite, i.e., we want to ensure that each benchmark program imposes a similar weight in affecting the partition of modules. To achieve this objective, we normalize the current consumption traces associated with each program so that the normalized average total current consumption of the chip while running that program becomes 1. Next, we concatenate the normalized current consumption traces for all of the programs such that a combined current trace is obtained for each module, and the combined current traces are used to build the graph as described in Section II.A.

The result of the partition is shown in Fig. 6(b), where the lightly darkened regions represent the modules operating between the $2V_{dd}$ and V_{dd} rails while the white regions represent the modules operating between the V_{dd} and GND rails.

C. Experimental setup for the validation of the module assignment

To validate that the module assignment obtained by our algorithm indeed reduces the power wasted in voltage regulators, we first use a regular grid structure similar to that shown in Fig. 7(a) to represent the V_{dd} rail, and we assume that current sources are attached to the nodes in the power grid, which model the currents consumed by modules. Next, we associate each node in the power grid with a region on the chip and assume that all the modules located in that region only source(sink) currents from(to) that particular node. For node i with the associated region A_i , if only part of a module M overlaps with A_i , as shown in Fig. 7(b), then only the corresponding portion of the current consumed by M is attached to node i . After the value of the current source attached to each power grid node is calculated, modified nodal analysis (MNA) is used to calculate the voltage droop across the V_{dd} grid and the current flowing through each of the voltage regulators.

We compare three module assignment schemes, one using the algorithm presented in this paper, a second using a bin-packing technique, and a third using the assignment optimized for one particular benchmark program, i.e., *equake*. For the bin-packing technique, we take the combined current traces as described in the previous subsection and calculate the average current consumption of each module. Then the module assignment is obtained such

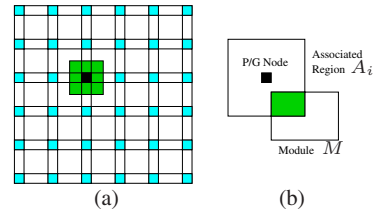


Fig. 7. Testing the actual wasted power (a) the structure of the V_{dd} grid and the region that source(sink) current from(to) a particular node (b) calculating the current source attached to a power grid node when the module only partially overlaps the region associated with the node.

that the difference between the currents consumed by the modules in the two different Vdd domains is minimized. Since the floorplan contains only 16 regular modules, we can afford to enumerate all possible module assignments and obtain the best bin-packing result. For the last approach, the module assignment is generated completely based on the current consumption traces of the *equake* program, and the purpose of this experiment is to see how the assignment performs across the benchmark suite when it is optimized with respect to only one particular program.

D. Result of comparison between different module assignment schemes

Our partition-based approach is highly efficient and the runtime of obtaining the assignment of modules for the DLX architecture is only 0.01sec on a desktop with a 3.2GHz Pentium-4 CPU. In what follows, we show the result of validation using the procedure described in the previous subsection.

Fig. 8(a) shows the comparison of the power wasted in voltage regulators between the three module assignment schemes. In obtaining the figure, we divide the total power wasted in voltage regulators by the total power consumed by regular modules, where the latter is termed “useful power” in the figure. We can see clearly that the design obtained using our approach has about 32% less power waste on average compared with the design where the module assignment is performed using the bin-packing technique. In addition, it is also clear that although the module assignment optimized for *equake* wastes less power in executing that particular program and another benchmark program *art*, it is not as good as our design in general and wastes about 23% more power on average.

It is important to notice that a design optimized for power also tends to achieve low IR noise in the V_{dd} grid. This is because to reduce the power waste, good current balance must be maintained locally as described in Section I, which is beneficial to reducing the IR noise since the current consumed by a module in one Vdd domain will immediately be recycled by some nearby modules in the other Vdd domain without flowing through a long resistive path in the power grid. In Fig. 8(b), we compare the maximum IR noise encountered in the V_{dd} grid when the module assignment is performed using the three different schemes presented above, respectively. It can be seen that our design technique achieves very reasonable IR noise, which demonstrates that while our partition-based module assignment scheme can significantly reduce the power waste, it does not sacrifice the noise performance of the design.

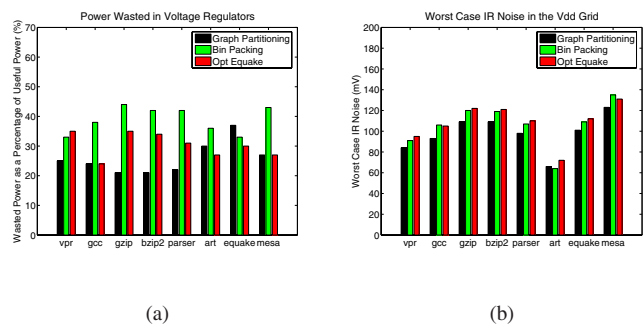


Fig. 8. Comparison between the module assignment using different approaches in terms of (a) the total power wasted in voltage regulators, and (b) the worst case IR noise in the V_{dd} grid, for the DLX architecture.

Note that for a floorplan with n modules, the entire design space contains 2^n different module assignments. It is not practical to test all possible solutions since each test requires a full simulation of the input current traces for each of the benchmark programs, which may take hours to complete, and for 2^n different candidate designs, the overall runtime of the enumeration

Benchmark	Avg P _{waste} (%)	Min P _{waste} (%)	Our P _{waste} (%)	Avg Noise (mV)	Min Noise (mV)	Our Noise (mV)
vpr	51	23	25	102	83	84
gcc	49	29	24	113	93	93
gzip	52	25	21	128	108	109
bzip2	52	26	21	129	109	109
parser	51	25	22	116	96	98
art	48	26	30	75	60	66
quake	47	25	37	118	98	101
mesa	49	28	27	140	121	123

TABLE I

COMPARISON BETWEEN THE RANDOM MODULE ASSIGNMENT AND THE ASSIGNMENT OBTAINED USING THE PARTITION-BASED APPROACH FOR THE DLX ARCHITECTURE.

Benchmark	Avg P _{waste} (%)	Min P _{waste} (%)	Our P _{waste} (%)	Avg Noise (mV)	Min Noise (mV)	Our Noise (mV)
Layer0	17.8	8.8	5.2	108	61	88
Layer1	18.5	9.1	5.8	78	48	57
Layer2	18.6	9.8	4.3	70	41	37

TABLE II

COMPARISON BETWEEN THE RANDOM MODULE ASSIGNMENT AND THE ASSIGNMENT OBTAINED USING THE PARTITION-BASED APPROACH FOR A 3D CIRCUIT.

becomes intractable. However, it is interesting to sample the design space and see how the performance of the module assignment obtained using our partition-based approach compares with other sample designs. To achieve this objective, we have generated 50 different module assignments randomly and calculated the total power wasted in voltage regulators and the worst case IR noise in the V_{dd} grid for each of the benchmark programs.

The result of comparison is shown in Table I, where the total power wasted in voltage regulators is, again, characterized as a percentage of useful power. We can see that the power wasted in our design is rather close to the minimum wasted power obtained through random experiments. We also emphasize that the minimum wasted power and IR noise data shown in Table I are not achieved by a unique module assignment among the 50 randomly generated designs. However, the design generated by our module assignment technique achieves consistently good result across the benchmark programs. This demonstrates the effectiveness of our approach, since in reality, a chip can only implement one design no matter how many different programs it will run in the future.

E. An example of a 3D circuit

We have also studied the effectiveness of using the stacked-Vdd paradigm in 3D circuit design. A three-layer 3D circuit structure is used and the floorplan of the n300 benchmark from the GSRC suite is generated. The three active layers contain 118, 92, and 90 modules, respectively, and the final footprint area of the die is scaled to 1cm^2 . Because of the lack of available current traces for GSRC benchmarks, we have assumed that the mean current consumption of each module is a random number between 100mA and 1000mA, and the instantaneous current consumption of module M_i is assumed to be varying randomly around its mean I_i^{mean} .² The assignment of modules to different Vdd domains is performed individually for each active layer, and the runtimes of the assignments are 1.98sec, 0.76sec, and 0.70sec, respectively. As is done previously, our module assignment strategy is compared with the bin-packing technique. For the bin-packing technique, it is impractical to enumerate all possible module assignments and find the best one in this example because of the large number of modules involved. As an alternative, we use hMetis [9] to perform a balanced two way partitioning of modules with the current consumption of each module as its weight, and the modules belonging to the same partition are assigned to the same Vdd domain.

In Fig. 9, we compare the two module assignment schemes in terms of the total power wasted in voltage regulators and the worst case IR noise in the V_{dd} grid for the three active layers in the 3D circuit. It can be seen that our module assignment approach results in a circuit that wastes far less power than the one generated by the bin-packing technique, and the noise performance of our design is also rather good.

As for the floorplan of the DLX architecture, we have also obtained 50 random module assignments for each active layer of the 3D circuit, and some of the critical statistics of the random experiments are listed in Table II. Again, it is clearly seen that the power wasted in our design is very low compared with other sample designs from the random experiments.

²The instantaneous current consumption of module M_i is given by $I_i(t) = I_i^{mean} \times (1 + \delta_g(t)) \times (1 + \delta_i(t))$, where $\delta_g(t)$ is a random number that remains the same for all of the modules, and $\delta_i(t)$ is a random number that varies from module to module. It is not difficult to see that $\delta_g(t)$ can be used to model the change of operations that affect the entire chip while $\delta_i(t)$ can be used to model the local variation in current consumption as a function of time t .

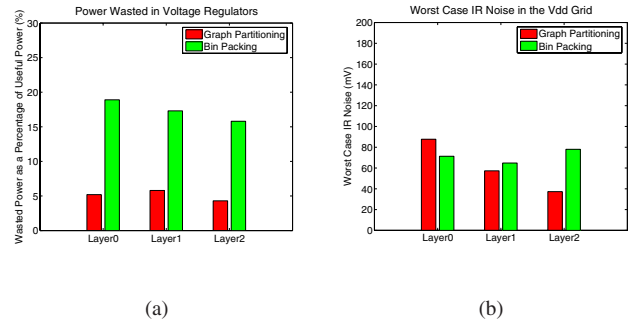


Fig. 9. Comparison between the module assignment using different approaches in terms of (a) the total power wasted in voltage regulators, and (b) the worst case IR noise in the V_{dd} grid, for the 3D circuit example.

V. CONCLUSIONS

In this paper, we presented a partition-based algorithm for efficiently assigning modules to different Vdd domains in a two level stacked-Vdd circuit so as to reduce the power wasted in voltage regulators. Experimental results on a DLX architecture and a 3D IC example show that compared with assigning the modules using a bin-packing technique, the designs generated by our algorithm waste much less power in voltage regulators, and therefore is more suitable for applications where the power constraint is stringent.

REFERENCES

- [1] Semiconductor Industry Association, "International technology roadmap for semiconductors, 2006." <http://public.itrs.net/Links/2006Update/2006UpdateFinal.htm>.
- [2] S. Rajapandian, K. Shepard, P. Hazucha, and T. Karnik, "High-tension power delivery: Operating $0.18\mu\text{m}$ CMOS digital logic at 5.4V," in *Digest of Technical Papers, IEEE International Solid-State Circuits Conference*, pp. 298–299, Feb. 2005.
- [3] E. Chiprout, "Fast flip-chip power grid analysis via locality and grid-shells," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 485–488, Nov. 2004.
- [4] S. M. Sait and H. Yousef, *VLSI Physical Design Automation: Theory and Practice*. Piscataway, NJ: IEEE Press, 1995.
- [5] S. N. Adya and I. L. Markov, "Fixed-outline floorplanning: Enabling hierarchical design," *IEEE Transactions on VLSI Systems*, vol. 11, pp. 1120–1135, Dec. 2003.
- [6] V. Nookala, D. Lilja, and S. S. Sapatnekar, "Temperature-aware floorplanning of microarchitecture blocks with IPC-power dependence modeling and transient analysis," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 298–303, Oct. 2006.
- [7] D. A. Patterson and J. L. Hennessy, *Computer Architecture: A Quantitative Approach*. San Francisco, CA: Morgan Kaufmann Publishers, 2nd ed., 1996.
- [8] R. E. Wunderlich, T. F. Wensisch, B. Falsafi, and J. C. Hoe, "SMARTS: Accelerating microarchitecture simulation via rigorous statistical sampling," in *Proceedings of the Annual International Symposium on Computer Architecture*, pp. 84–97, Jun. 2003.
- [9] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Applications in VLSI domain," *IEEE Transactions on VLSI Systems*, vol. 7, pp. 69–79, Mar. 1999.