

Voltage Island Generation under Performance Requirement for SoC Designs*

Wai-Kei Mak and Jr-Wei Chen
 Department of Computer Science
 National Tsing Hua University
 Taiwan 300 R.O.C.

Abstract— Using multiple supply voltages on a SoC design is an efficient way to achieve low power. However, it may lead to a complex power network and a huge number of level shifters if we just set the cores to operate at their respective lowest voltage levels. We present two formulations for the voltage level assignment problem. The first is exact but takes longer time to compute a solution. The second can be solved much faster with virtually no loss on optimality. In addition, we propose a modification to the traditional floorplanning framework. Unlike previous works [1] [2], we can optimize the total power consumption, the level shifter overhead, and the power network complexity without compromising the wirelength and the chip area. In the experiments, we obtained 17- 53% power savings with voltage island generation.

I. INTRODUCTION

Today more and more devices can be implemented on a single chip. This enables various applications to be realized as System-on-a-Chip (SoC) designs by the integration of various IP cores on a single chip. Power consumption becomes an important issue in deep sub-micron designs. High power consumption shortens the battery life of handheld devices. It may also lead to detrimental thermal effects and reduce the system reliability. The dynamic power P_d is computed by

$$P_d = f k C V^2$$

where f denotes the clock frequency, k denotes the switching activity, C denotes the load capacitance, and V denotes the supply voltage. We can easily see that the supply voltage is a very important factor in determining the dynamic power consumption.

Today's designs usually have multiple clocks running at different rates because of the required performance of all functional blocks are not the same [3]. And the concept of voltage island was proposed in [4] to leverage voltage optimization of individual functional blocks of a SoC design. For example, the most performance-critical block like a processor core requires the highest voltage level while other functions such as memories or control logic which coexist on the SoC just require a low level of voltage.

Voltage island formation can reduce the power consumption of a chip when there is a mixture of cores which need to run at different levels of performance. A voltage island is a group of contiguous on-chip cores which are powered by the same voltage level. Fig. 1 shows an example where core 3 is the most

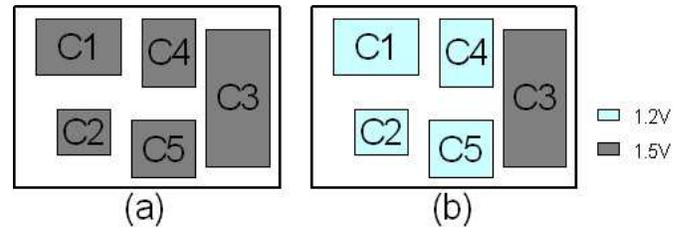


Fig. 1. (a) Without voltage islands. (b) With voltage islands.

performance-critical element in the design. Without voltage islands, the chip voltage level has to be set at 1.5V because core 3 has to operate at 1.5V as shown in Fig. 1(a). However, with voltage islands, the total power consumption can be reduced by operating non-performance-critical cores 1, 2, 4, and 5 at 1.2V as shown in Fig. 1(b) while the overall system performance is still maintained.

But making mixed voltage designs is not without overheads. A level shifter has to be inserted to an interconnect from a low voltage core to a high voltage core because a circuit may suffer from excessive leakage energy if low voltage gates directly drive high voltage gates [5]. As a result, the extra level shifters will cause area and performance overhead as shown in Fig. 2. The level shifters will also consume power. Moreover, the power network will become highly fragmented if we simply set all the cores to operate at their respective lowest voltage levels. This complicates the design of the power delivery network and significantly increases the design overhead [6]. We note that when the power network becomes more complex, power supply noise related issues will get worse.

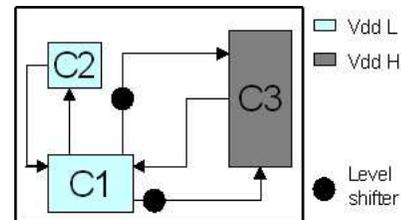


Fig. 2. Level Shifter Insertion.

In our work, we take advantage of the voltage island technique to reduce the power consumption while carefully taking the level shifter overhead and power network complexity into account. Moreover, unlike other previous works on voltage island planning, we do not need to sacrifice the wirelength and chip area. The rest of this paper is organized as follows. In section II, we review some previous works. In section III, we present two models for the Voltage Level Assignment (VLA) problem for a given floorplan. Our proposed voltage island floorplanning framework is discussed in section IV. The experimental results and the conclusion are given in sections V

*This work was partially supported by the National Science Council and the Ministry of Economic Affairs of Taiwan, R.O.C., under NSC 95-2220-E-007-018 and MOEA 95-EC-17-A-01-S1-038.

and VI.

II. PREVIOUS WORKS

Voltage islands were first used in IBM’s CU-08 manufacturing process for application specific integrated processors [7]. The voltage island technique was proposed to reduce power consumption substantially by allowing designers to build processors that vary their voltages across a chip. The advantage of the voltage island technique and other issues related to its use are discussed in details in [4]. Design techniques to physically implement flexible voltage islands for multiple supply voltage designs are presented in [8] [9].

Recently Wu et al. [10] investigated the voltage island generation problem at the cell level. They used a dynamic programming approach and a fast heuristic algorithm for post-placement voltage island generation under performance requirement. On the other hand, we consider voltage island generation for core-based SOC designs under performance requirement.

Hu et al. [1] addressed the voltage island planning problem for core-based SOC designs and proposed a simulated annealing based partitioning and floorplanning approach. Unfortunately, their method has no effective control on the number of voltage islands and the number of level shifters because they only used a cost which is a weighted sum of the core power consumption, the level conversion area overhead, and the number of islands. The optimization is performed without any constraint and the power dissipation of level shifters is ignored. The experimental results in [1] only reported the power consumption but did not report the increase in total wirelength of their solutions compared to those by a traditional floorplanner for wirelength and area optimization. Power savings of 14% to 28% were obtained in [1]. Though the authors of [1] said that wirelength optimization can be considered by adding the wirelength into the cost function, it is likely to bring down their power savings.

In [2], a genetic algorithm(GA)-based voltage island partitioning algorithm is combined with a simulated annealing(SA)-based floorplanning algorithm for voltage island planning. And a temperature estimation tool was implemented so that a weighted sum of chip area, wirelength, and chip temperature can be optimized by the SA-based floorplanner. Its experimental results reported the temperature, wirelength, and the deadspace when the combined GA and SA approach was applied with three optimization objectives : “chip area + wire”, “power + area + wire”, and “temperature + area + wire”. There were noticeable increase in chip areas when the objective function changed from a weighted sum of area and wirelength only to a weighted sum of power, area, and wirelength. The level of power savings was not reported in [2]. It also failed to consider the area and power overheads of level shifters.

Unlike [1] and [2], our proposed approach can optimize the total power consumption of cores and level shifters, the level conversion area overhead, and the power network complexity without compromising the wirelength and chip area of a traditional wirelength and area optimization floorplanner. And we achieved 17% to 53% of power savings.

III. VOLTAGE LEVEL ASSIGNMENT (VLA)

In this work, we would like to consider how to perform voltage island floorplanning with multiple supply voltages for SoC design. A simpler but equally important problem is the voltage level assignment (VLA) problem. In this section, we focus on VLA with a given floorplan. The overall solution approach when no floorplan is given will be presented in section IV. We present two formulations for the VLA problem. The first is

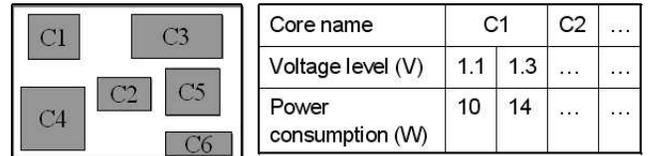
exact but takes longer time to compute a solution. The second can be solved much faster with insignificant sacrifice on solution quality.

The goal of voltage level assignment is to reduce the total power consumption while controlling the level conversion area overhead and the power network complexity without compromising system performance. We note that none of the previous works has introduced any efficient algorithm to perform voltage level assignment of each core in a single chip.

At first glance, it seems that we can reduce the total power consumption to a minimum by setting all the cores to operate at their corresponding lowest voltage level. However, it may result in a complex power network. So we want to have a less complex power network at a minimal expense of power when assigning the voltage levels to the cores.

A. Problem Definition

The voltage level assignment problem is as follows. Given (1) a netlist of cores, (2) a floorplan of the cores, (3) the legal voltage levels¹ of each core and the corresponding power consumptions at different voltage levels (Fig. 3), we want to determine a proper voltage level for each core. Like [1] and [2], our work is applied at the system level, we assume that the inputs and outputs of the cores are registered, therefore no critical path will span across two cores. Hence the legal voltages of each core can be characterized independently as in [1] and [2]. A good solution is determined by the overall power consumption, level shifter area overheads, and power network complexity.



Core name	C1	C2	...
Voltage level (V)	1.1	1.3
Power consumption (W)	10	14

Fig. 3. Input information.

B. Exact 0-1 ILP Formulation

We propose a 0-1 integer linear programming formulation for the voltage level assignment problem. We want to minimize the following cost.

$$cost = P_{total} + \beta \cdot F$$

where P_{total} denotes total power consumption, F denotes the fragmentation cost. The total power consumption P_{total} is equal to the sum of the power consumption of all cores, P_{cores} , and the power consumption of added level shifters, $P_{shifters}$. Note that P_{cores} is determined by the voltage level assigned to each core while $P_{shifters}$ is dependent on the switching activities and the voltage transitions of the signals for which level shifters needed to be inserted. The fragmentation cost F measures the fragmentation of the power network. β is a user specified parameter.

Suppose there are m cores. We define the notations used in our ILP formulation below.

- Let L_i denote the number of legal voltage levels of core i ($1 \leq i \leq m$).

¹The legal voltage levels of a core are the available operating voltages at which the core will satisfy its performance requirement.

- Let the available set of operating voltages for core i be $\{V_{i1}, V_{i2}, \dots, V_{iL_i}\}$ where V_{ip} denotes the voltage of core i when it operates at voltage level p ($1 \leq i \leq m$, $1 \leq p \leq L_i$).
- Let P_{ip} denote the power consumption of core i when it operates at voltage level p ($1 \leq i \leq m$, $1 \leq p \leq L_i$).
- Let S_{ij} denote the number of signals from core i to core j ($1 \leq i \leq m$, $1 \leq j \leq m$).
- Let $P_{ip,jq}$ denote the extra power consumption for the level shifters added (if any) to the interconnects from core i to core j when core i operates at voltage V_{ip} and core j operates at voltage V_{jq} . (Note that $P_{ip,jq}$ is 0 when $i = j$ or $V_{ip} \geq V_{jq}$, since no level shifters are inserted in these cases. For other cases, $P_{ip,jq}$ can be pre-computed using the switching activity information of the signals.)
- Let \mathcal{N}_i denote the set of physically neighboring cores of core i ($1 \leq i \leq m$).

We define binary decision variables b_{ip} ($1 \leq i \leq m$, $1 \leq p \leq L_i$) such that b_{ip} is 1 if core i is assigned voltage level p , and b_{ip} is 0 otherwise. It is easy to see that we should assign a single voltage for each core, so we have

$$\sum_{p=1}^{L_i} b_{ip} = 1 \quad \text{for } 1 \leq i \leq m \quad (1)$$

$$b_{ip} = 0 \text{ or } 1 \quad (2)$$

We can express P_{cores} , $P_{shifters}$, and F as shown below.

The core power consumption P_{cores} is the total power consumption of all cores. We have

$$P_{cores} = \sum_{i=1}^m \sum_{p=1}^{L_i} (P_{ip} \cdot b_{ip}) \quad (3)$$

The level shifter power consumption $P_{shifters}$ is the power consumed by the added level shifters. Level shifters consume power just like conventional buffers. As we discussed before, level shifters must be inserted to the interconnects from low voltage cores to high voltage cores. We have

$$P_{shifters} = \sum_{i=1}^m \sum_{j=1}^m \sum_{p=1}^{L_i} \sum_{q=1}^{L_j} (P_{ip,jq} \cdot b_{ip} \cdot b_{jq})$$

We define the fragmentation cost F to model the power network complexity problem. The fragmentation cost is equal to the number of pair of physically adjacent cores operating at different voltage levels. We have

$$F = \sum_{i=1}^m \sum_{j \substack{\text{s.t. } i < j \leq m \\ \wedge j \in \mathcal{N}_i}} \sum_{\substack{1 \leq p \leq L_i \\ 1 \leq q \leq L_j \\ V_{ip} \neq V_{jq}}} (b_{ip} \cdot b_{jq})$$

Notice that the condition $i < j$ is to ensure that no pair is considered twice.

However, the equations for $P_{shifters}$ and F above are illegal in an ILP because they are non-linear. As a result, we introduce Boolean variables $b'_{ip,jq}$ ($1 \leq i \leq m$, $1 \leq j \leq m$, $1 \leq p \leq L_i$, $1 \leq q \leq L_j$) to replace $b_{ip} \cdot b_{jq}$ by enforcing the following artificial constraints in our ILP:

$$b'_{ip,jq} \geq b_{ip} + b_{jq} - 1 \quad (4)$$

$$b'_{ip,jq} = 0 \text{ or } 1 \quad (5)$$

We can re-write $P_{shifters}$ as

$$P_{shifters} = \sum_{i=1}^m \sum_{j=1}^m \sum_{p=1}^{L_i} \sum_{q=1}^{L_j} (P_{ip,jq} \cdot b'_{ip,jq}) \quad (6)$$

Because of constraints(4) and (5), and the fact that $P_{shifters}$ appears in the cost function to be minimized, $b'_{ip,jq}$ will be equal to 0 unless both b_{ip} and b_{jq} are 1. Similarly, F can be re-written as

$$F = \sum_{i=1}^m \sum_{j \substack{\text{s.t. } i < j \leq m \\ \wedge j \in \mathcal{N}_i}} \sum_{\substack{1 \leq p \leq L_i \\ 1 \leq q \leq L_j \\ V_{ip} \neq V_{jq}}} b'_{ip,jq} \quad (7)$$

Next we discuss three useful constraints to include. The first is the level shifter area constraint, the second is the adjacency constraint, and the last is the power consumption constraint.

(1) Level shifter area constraint

We know that the level shifters will consume power and we have already modeled that in our cost function. Besides, the level shifters also occupy area. So, we set a level shifter area constraint to limit the level shifter area overhead.

$$\sum_{i=1}^m \sum_{j=1}^m \sum_{\substack{i \neq j \\ 1 \leq p \leq L_i \\ 1 \leq q \leq L_j \\ V_{ip} < V_{jq}}} (S_{ij} \cdot b'_{ip,jq}) \leq A \quad (8)$$

The LHS of the equation is the number of level shifters needed. Recall that level shifters only need to be inserted to interconnects from low voltage to high voltage. The RHS of the equation is the user specified upper bound, A , on the level shifter area which may be set equal to the available dead space in the floorplan.

(2) Adjacency constraint

In addition to the fragmentation cost which we included in the cost function, we impose some simple rules to insure a low level of fragmentation in the power network. The adjacency constraints are added to ensure that each core must have some neighboring cores operating at the same voltage as itself. And we propose to adjust the requirement according to the number of neighbors a core has. Therefore, a core with more neighboring cores should have more neighboring cores operating at the same voltage level as itself.

$$\sum_{j \in \mathcal{N}_i} \sum_{\substack{p,q \text{ s.t.} \\ V_{ip} = V_{jq}}} b'_{ip,jq} \geq u_i \quad \text{for } 1 < i \leq m \quad (9)$$

where $u_i=1$ if core i has two to three neighbors, $u_i=2$ if core i has four to seven neighbors, and $u_i=3$ if core i has more than seven neighbors. Note that the values in this constraint are user-specified. We choose the values above because they are shown to yield a low level of fragmentation experimentally.

(3) Power consumption constraint

We may also set a targeted level of power consumption by incorporating one more constraint,

$$P_{total} \leq P_{bound} \quad (10)$$

where P_{bound} is an upper bound on the desired power consumption.

C. Fast Model

In this subsection, we propose a fast model to compute the voltage assignments for all cores with a reduced number of 0-1 integer variables. Experimental results show that there is little loss on optimality compared to the exact model. As a simplification, we assume that a level shifter is inserted whenever an interconnect connects two cores operating at different voltages. We re-write the cost function as

$$cost = P_{cores} + (Max_{shiffters} - R_{shiffters}) + \beta \cdot (Max_{frag} - R_{frag})$$

$Max_{shiffters}$ denotes the maximum possible power consumed by the level shifters if a level shifter is inserted to every interconnect. Max_{frag} denotes the maximum possible fragmentation cost which is equal to the number of pairs of physically adjacent cores. $R_{shiffters}$ denotes the power reduction when the level shifters on interconnects between two cores operating at the same voltage are removed. R_{frag} denotes the reduction in fragmentation cost which is equal to the number of pairs of physically adjacent cores operating at the same voltage.

- Let $\{V_1, V_2, \dots, V_L\}$ denotes the set of voltage levels.
- Let R_{ij} denote the power saving of unnecessary level shifters on the interconnects from core i to core j when core i and core j both operate at the same voltage level. Note that R_{ij} is 0 when $i = j$. Otherwise, R_{ij} can be pre-computed using switching activity information of the signals from core i to core j .

In the fast model, we include constraints (1), (2), (3) as before. And we introduce boolean variable b'_{ijp} ($1 \leq i \leq m$, $1 \leq j \leq m$, $1 \leq p \leq L$)² such that b'_{ijp} is equal to 1 if and only if both core i and core j are assigned to voltage V_p which can be modeled by the following constraints:

$$b'_{ijp} \leq b_{ip} \quad (11)$$

$$b'_{ijp} \leq b_{jp} \quad (12)$$

Since no level shifters are needed between cores operating at the same voltage level, we can model the power saving of unnecessary level shifters using the following constraint.

$$R_{shiffters} = \sum_{i=1}^m \sum_{j=1}^m \sum_{p=1}^L (R_{ij} \cdot b'_{ijp}) \quad (13)$$

R_{frag} which equals the number of pairs of physically adjacent cores operating at the same voltage can be expressed by the following constraint.

$$R_{frag} = \sum_{i=1}^m \sum_{\substack{j \text{ s.t. } i < j \leq m \\ \wedge j \in \mathcal{N}_i}} \sum_{p=1}^L b'_{ijp} \quad (14)$$

In addition, we can impose level shifter area constraint, adjacency constraint, and power consumption constraint as follows.

$$\sum_{i=1}^m \sum_{j=1}^m S_{ij} - \sum_{i=1}^m \sum_{j=1}^m \sum_{p=1}^L (b'_{ijp} \cdot S_{ij}) \leq A \quad (15)$$

$$\sum_{j \in \mathcal{N}_i} \sum_{p=1}^L b'_{ijp} \geq u_i \quad \text{for } 1 \leq i \leq m \quad (16)$$

$$P_{cores} + (Max_{shiffters} - R_{shiffters}) \leq P_{bound} \quad (17)$$

²Note that variable b'_{ijp} is not needed and can be removed if either core i or core j 's legal voltage does not include V_p , since b'_{ijp} will be identically equal to 0.

The parameters u_i and P_{bound} are the same as described in section III-B.

We experimented with different formulations and refined our model several times. The final versions of our exact model(section III-B) and the fast model(section III-C) are described above.

IV. VOLTAGE ISLAND FLOORPLANNING FRAMEWORK

In section III, we showed how to solve the voltage level assignment problem. In this section, we will present the complete framework for voltage island floorplanning. The framework is divided into two phases as shown in Fig. 4. The inputs are the widths and heights of the hard cores, the areas and aspect ratios of the soft cores, and the legal voltage levels and corresponding power consumptions of the cores. In phase 1, we compute several wirelength and area optimized floorplans. In our work, we modified the simulated annealing based floorplanner using the B*-tree representation [11] to keep the top n floorplans in terms of wirelength and chip area.

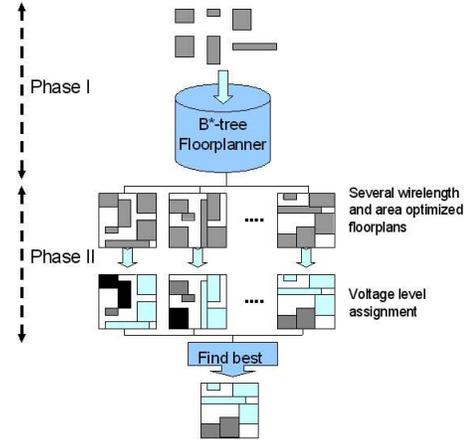


Fig. 4. Voltage island floorplanning framework.

In phase 2, we make voltage assignment for each candidate floorplan computed in phase 1. Finally, the floorplan with the best power and fragmentation cost after voltage assignment is picked. This framework was shown to produce voltage island floorplans of extremely high quality experimentally. It can effectively optimize the total power consumption, the level conversion area overhead, and the power network complexity without sacrificing the wirelength and chip area. The algorithm is summarized in Fig. 5.

V. EXPERIMENTAL RESULTS

We implemented our algorithm in C++ and did the experiments on a workstation with an AMD Opteron 2.1GHz processor and 4GB memory running Red Hat Linux Release 9(Shrike). We used lp_solve version 5.5.0.2[12] to solve the ILPs.

We tested our voltage island floorplanner on five conventional MCNC benchmarks as in [13] [2]. In addition, we created six more artificial benchmarks for testing. The specifications of the benchmarks are shown in Table I. Note that we do not compare our experimental results with [1] because we cannot obtain the benchmarks used in [1]. But the computation of the power consumption is based on the same method suggested by the authors of [1].

We set the user specified parameter β to 25 empirically (see section III-B). We randomly generate the switching activities of

Algorithm	
Inputs :	Given the dimensions/area & aspect ratio, available voltage levels and corresponding power consumptions of each core.
Outputs :	An optimized floorplan with voltage level assignment
Begin	
L1:	Run B*-tree based floorplanner to get n best solutions
L2:	for each floorplan do
L3:	Run ILP-based voltage level assignment
L4:	Get the power and fragmentation cost
L5:	If the cost is smaller than the smallest cost then
L6:	Update the smallest cost
L7:	Store the current floorplan as the best floorplan
L8:	end do
L9:	end

Fig. 5. Algorithm outline.

TABLE I
BENCHMARK SPECIFICATION.

Benchmark	#blocks	Benchmark	#blocks
apte	9	2xerox	20
xerox	10	2hp	22
hp	11	ami75	75
ami33	33	ami99	99
ami49	49	ami200	200
		ami300	300

the nets such that 80% of them are below 0.5 and 20% are above 0.5. We assume that there are 4 different voltage levels which are 1.1V, 1.3V, 1.5V, and 1.8V in each case. Since the area approximately correlates to the number of gates, we assume that the power consumption is proportional to the area of a block. Furthermore, we assume that the power consumption of a block is proportional to the square of its supply voltage. We assume that the dimensions of a level shifter are 10 x 10. We assume that the power consumption of a level shifter is 1 if its switching activity is 1.

Table IV shows the comparison between the exact ILP formulation and the fast model. “ P_{total} diff.” denotes the difference of total power consumption between the exact model and fast model. “T diff.” denotes the run time difference between the exact model and the fast model while calling lp solver. Our fast model can dramatically reduce the run time and the result is virtually the same as the exact model. No result was produced for ami200 and ami300 even after an hour using the exact model but they were handled comfortably with the fast model. The power savings of ami200 and ami300 are 41.90% and 40.69% using the fast model. It shows that our fast model can efficiently solve the VLA problem even if there is a large number of cores with slight power sacrifice.

We modified the B*-tree floorplanner to generate five best floorplans in terms of wirelength and area for each design. Subsequently, we used our ILP-based VLA to compute the voltage level assignment for the five floorplans and picked the one with the least cost. While the difference of the wirelength and area between the five floorplans is no more than 3%, Table II shows that there can be substantial difference in the power and fragmentation cost between the five candidate floorplans after voltage assignment. Table III shows the difference of power and fragmentation using three experimental settings. The “Low” column represents the results when each core operates at its individual lowest legal voltage that still satisfies its timing requirement. The “Single” column represents the results when

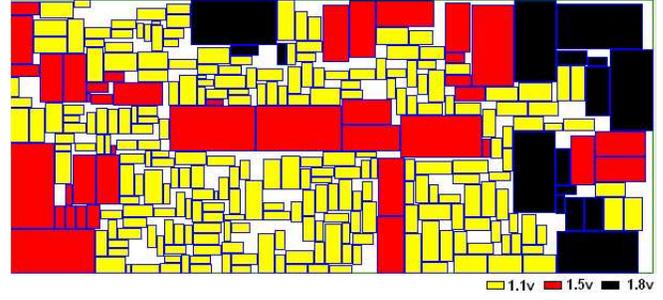


Fig. 6. Floorplanning result for ami300.

the entire design operates at a single voltage that satisfies the timing requirement of all cores. The “VLA” column represents the results which is obtained by our algorithm. Table V shows the overall result. The wirelength, chip area and dead space are obtained from phase 1. The run time means the overall run time(phase 1 + phase 2(fast model)). Notice that total power consumption without VLA is the power consumption when the circuit operates at a single voltage level. Fig. 6 shows the floorplan result of ami300 obtained by our algorithm.

TABLE II
DIFFERENCE IN THE COST AFTER VOLTAGE ASSIGNMENT.

Benchmark	Power and fragmentation cost		
	Best	Worst	Difference(%)
apte	1188.75	1359.72	12.57 %
xerox	2140.99	2342.19	8.59 %
hp	2423.18	2552.44	5.06 %
ami33	5212.14	5757.82	9.47 %
ami49	8254.78	8430.86	2.08 %
2xerox	3634.56	3978.56	8.64 %
2hp	4326.32	4593.05	5.80 %
ami75	12988.1	13551.8	4.16 %
ami99	16546.85	16717.65	1.02 %
ami200	32987.65	33621.65	1.88%
ami300	50065.3	50823.3	1.49%

TABLE III
DIFFERENCE OF POWER AND FRAGMENTATION USING LOW, SINGLE VOLTAGE, AND VLA.

Benchmark	Power			Fragmentation		
	Low	Single	VLA	Low	Single	VLA
apte	1097.05	2409.92	1113.75	2	0	3
xerox	1893.4	2677.69	2065.99	8	0	3
hp	1920.6	2945.45	2198.18	11	0	9
ami33	4437.85	8836.36	4937.14	16	0	11
ami49	7532.85	13120.7	7854.78	22	0	16
2xerox	3287.5	5355.37	3559.56	11	0	3
2hp	3848.6	5090.91	4201.32	20	0	5
ami75	11716.6	20082.6	12238.1	37	0	30
ami99	15167.7	26509.1	15596.85	43	0	38
ami200	30867.25	53553.7	31749.93	118	0	74
ami300	45461.2	80330.6	47640.03	147	0	97

VI. CONCLUSIONS

As the demand for energy efficiency increases, multi-voltage SoC designs will become more and more common. How to assign a proper voltage for each core and create a floorplan

TABLE IV
EXACT MODEL V.S. FAST MODEL.

Benchmark	Exact model		Fast model		P_{total} diff.	T diff.
	P_{total}	run time (s)	P_{total}	run time (s)		
apte	1113.75	0.079	1113.75	0.012	0 %	6.58 X
xerox	2065.99	0.117	2065.99	0.031	0 %	3.77 X
hp	2198.18	0.02	2198.18	0.015	0 %	1.33 X
ami33	4937.14	10.585	4937.14	2.1359	0 %	4.96 X
ami49	7789.56	13.52	7854.78	2.011	0.83 %	6.72 X
2xerox	3559.56	3.476	3559.56	0.343	0 %	10.134 X
2hp	4201.32	0.566	4201.32	0.172	0 %	3.29 X
ami75	12040.963	195.003	12238.1	12.134	1.61 %	16.07 X
ami99	15596.85	147.16	15596.85	37.571	0 %	3.91 X
ami200	N/A	>3600	31112.65	49.359	N/A	>72.94 X
ami300	N/A	>3600	47640.03	125.258	N/A	>28.74 X

TABLE V
EXPERIMENTAL RESULTS OF WIRELENGTH, CHIP AREA, DEAD SPACE, POWER CONSUMPTION, AND EXECUTION TIME.

Benchmark	Wirelength	Chip area	Dead space %	Power (W/O VLA)	Power (With VLA)		Power saving	Run time (s)	
					P_{cores}	$P_{shifters}$		phase I	phase II
apte	435.502	48.2118	3.42285	2409.92	1093.95	19.8	53.78%	5.3435	0.1390
xerox	411.096	20.4245	5.25927	2677.69	1658.94	407.05	22.85 %	6.8626	0.2173
hp	211.032	9.3639	5.70000	2945.45	2119.88	78.3	25.37 %	122.8375	0.1564
ami33	69.172	1.2274	5.78443	8836.36	4567.74	369.4	44.12 %	76.4935	12.9033
ami49	926.389	37.8852	6.4400	13120.7	7800.03	54.75	41.13 %	80.7366	9.7319
2xerox	1078.37	40.2150	3.7658	5355.37	3055.36	504.2	33.53 %	73.0364	1.7174
2hp	373.938	18.7180	5.6500	5090.91	3909.27	292.05	17.47 %	25.8321	1.0357
ami75	1566.36	62.2056	6.3300	20082.6	12026.2	211.9	39.06 %	251.8367	64.7361
ami99	2228.55	77.0529	7.6664	26509.1	15432	164.85	41.16 %	494.5228	189.9016
ami200	4439.84	160.32	10.8827	53553.7	31247.83	502.113	41.90 %	3606.9174	244.7326
ami300	6898.99	249.707	12.0281	80330.6	47211	429.03	40.69 %	7023.543	357.1431
average	—	—	—	—	—	—	36.46 %	—	—

with voltage islands is a non-trivial problem. The area and power dissipation of level shifters in mixed voltage designs cannot be ignored. The power network complexity also needs to be carefully controlled. In this paper, we proposed a new effective floorplanning framework targeting multiple supply voltage core-based designs such that the total power consumption of cores and level shifters, the level conversion area overhead, and the power network complexity can be effectively optimized without compromising the wirelength and chip area.

ACKNOWLEDGEMENT

We would like to thank Youngsoo Shin for answering our inquiry about the experimental results in [1].

REFERENCES

- [1] Jingcao Hu, Youngsoo Shin, Nagu Dhanwada and Radu Marculescu, "Architecting Voltage Island in Core-based System-on-a-Chip Designs", *International Symposium on Low Power Electronics and Design*, 9-11 Aug. 2004, pp. 180-185, 2004
- [2] W.-L. Hung, G. M. Link, Yuan Xie, N. Vijaykrishnan, N. Dhanwada, J. Corner, "Temperature-Aware Voltage Islands Architecting in System-on-Chip Designs", *International Conference On Computer Design*, 2-5 Oct. 2005, pp. 689-694, 2005
- [3] Hsiao-Pin Su, Wu, A.C.-H., Youn-Long Lin, "A timing-driven soft-macro placement and resynthesis method in interaction with chip floorplanning", *Proceedings of the 36th conference on Design automation*, 21-25 June 1999, pp. 262-267, 1999
- [4] D. E. Lackey, P. S. Zuchowski, T. R. Bednar, D.W. Stout, S.W. Gould, and J. M. Cohn, "Managing power and performance for System-on-Chip designs using voltage islands", *IEEE/ACM International Conference on Computer Aided Design*, 10-14 Nov. 2002, pp. 195-202, 2002
- [5] Abdulkadir U. Diril, Yuvraj S. Dhillon, Abhijit Chatterjee, Adit D. Singh, "Level-Shifter Free Design of Low Power Dual Supply Voltage CMOS Circuits Using Dual Threshold Voltages", *Proceedings of the 18th International Conference on VLSI Design*, 2005, pp. 159-164, 2005
- [6] Aveek Sarkar, "A methodology for IC power grid design", <http://EEdesigns.com>, 2005
- [7] John G. Spooner, "Chip designers voyage voltage island", [http://news.com.com/Chip designers voyage to voltage island/2100-1001-3-934355.html](http://news.com.com/Chip%20designers%20voyage%20to%20voltage%20island/2100-1001-3-934355.html), 2002
- [8] Ruchir Puri, David Kung, Leon Stok, "Minimizing Power with Flexible Voltage Islands", *IEEE International Symposium on Circuits and Systems*, 23-26 May 2005, pp. 21-24, 2005
- [9] Ruchir Puri, Leon Stok, John Cohn, David Kung, David Pan, Dennis Sylvester, Ashish Srivastava, Sarvesh Kulkarni, "Pushing ASIC Performance in a Power Envelope", *Proceedings of the 40th conference on Design automation*, 2-6 June 2003, pp. 788-793, 2003
- [10] Huaizhi Wu, I-Min Liu, Martin D.F. Wong, Yusu Wang, "Post-Placement Voltage Island Generation under Performance Requirement", *IEEE/ACM International Conference on Computer-Aided Design*, 6-10 Nov. 2005, pp. 309-316, 2005
- [11] B*-tree floorplanner, <http://cc.ee.ntu.edu.tw/~ywchang/research.html>
- [12] ILP Solver, http://groups.yahoo.com/group/lp_solve/, 2005
- [13] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, and S.-W. Wu, "B*-tree: A new representation for non-slicing floorplans.", *Proceedings of the 37th conference on Design automation*, 5-9 June 2000, pp. 458-463, 2000