Voltage-Island Driven Floorplanning Considering Level-Shifter Positions

Bei Yu , Sheqin Dong
Department of Computer Science & Technology
Tsinghua University
Beijing, China 100088
yubei.dongsq@gmail.com

Satoshi GOTO, Song Chen
Graduate School of IPS
Waseda University, Kitakyushu, Japan 808-0135
goto@waseda.jp
chensong@aoni.waseda.jp

ABSTRACT

Power optimization has become a significant issue when the CMOS technology entered the nanometer era. Multiple-Supply Voltage (MSV) is a popular and effective method for power reduction. Level shifters may cause area and Interconnect Length Overhead(ILO), and should be considered during floorplanning and post-floorplanning stages. In this paper, we propose a two phases framework VLSAF to solve voltage and level shifter assignment problem. At floorplanning phase, we use: a convex cost network flow algorithm to assign voltage; a minimum cost flow algorithm to assign level shifter. At post-floorplanning phase, a heuristic method is adopted to redistribute white spaces and calculate the positions and shapes of level shifters. Experimental results show VLSAF is effective.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids-Placement and routing.

General Terms

Algorithm, Design

Keywords

Voltage-Island, Voltage Assignment, Convex Network Flow, Level Shifter Assignment, White Space Redistribution

1. INTRODUCTION

Power optimization has become a significant issue when the CMOS technology entered the nanometer era. High power consumption not only shorten the battery life for handheld devices but also cause thermal and reliability problems. As a result, many techniques were introduced to deal with power optimization. Among the existing techniques, *Multiple-Supply Voltage* (MSV)[1] is a popular and effective method for both dynamic and static power reduction while maintaining performance.

In the MSV design, high voltage is assigned to timing critical blocks while lower voltage is assigned to noncritical blocks, so the power can be saved without degrading the overall circuit performance. Accordingly, MSV aware floorplanning includes two major problems: voltage assignment and floorplanning, which make the design process much more complicated.

Level-shifter [1] has to be inserted to an interconnect when a low voltage module drives a high voltage module or a circuit may suffer from excessive leakage energy. From [5] we can observe that the number of level shifters increase rapidly as modules increase and the area level-shifters consume can not ignore. As a result, level-shifters should be considered during floorplanning and post-floorplanning stages.

There are a number of works addressing island generation and voltage assignment in floorplanning and placement. Among these works, MSV is considered at various stages, including voltage assignment before floorplanning[4][5]; during floorplanning[6][7]; and post-floorplaning / post-placement [8][9] [10][11].

Lee et al.[5] handle voltage assignment by dynamic programming, and level shifters are inserted as soft block according to the voltage assignment result before floorplanning. At last, power network resource are considered during floorplanning. However, there are some deficiencies in the work: first, voltage assignment is handled before floorplanning, so physical information such as the distances among modules are not able to be taken into account; secondly, the search space is large if level-shifters are considered as a module.

An approach based on ILP is used in [9] for voltage assignment at the post-floorplanning stage. Level-shifter planning and power-network resources are considered. However, their approach does not consider level-shifter's area consumption and relies on the floorplanning result.

To make use of physical information such as the length of interconnects among modules, voltage assignment problem should be addressed during floorplanning. Ma et al.[7] transform voltage assignment problem into a convex cost network flow problem, and integrate it into floorplanning stage. However, their approach consider neither level-shifters' area overhead nor level-shifters' positions.

Besides area and power issue, it is also important to minimize the number of voltage islands for the purpose of reducing the cost of power-network resource. [3] points out that although more supply voltages may lead to more power

GLSVLSI'09, May 10–12, 2009, Boston, Massachusetts, USA.

saving, two supply voltages are sufficient for practical circuit designs. Therefore, in this paper, we consider two voltages domains.

In this paper, we propose a two phases framework. At floorplanning phase, we use: a convex cost network flow algorithm to assign voltage; a minimum cost flow algorithm to assign level shifter. At post-floorplanning phase, a heuristic method is adopted to redistribute white spaces and calculate the positions and shapes of level shifters.

The remainder of this paper is organized as follows. Section 2 defines the voltage-island driven floorplanning problem. Section 3 presents our algorithm flow. Section 4 reports our experimental results. At last, Section 5 concludes this paper.

2. PROBLEM FORMULATION

DEFINITION 1 (ROOM). *Given a chip, it can be dissected into rectangular areas and each area is assigned at most one module. The rectangular area is defined as a Room.*

DEFINITION 2 (WHITE SPACE). *In each room, the space not occupied by the module is defined as White Space.*

In this paper, we use CBL[2] to represent every floorplan generated. CBL is a topological representation dissecting the chip into rectangular rooms. Besides, all the nets are two-pin nets, and multi-pin nets can be decomposed into a set of source-sink two-pin nets. The wire length of every net is calculated by half-perimeter model.

DEFINITION 3 (INTERCONNECT LENGTH OVERHEAD). *Each level-shifter belongs to a net, if level-shifter is outside net's bounding box, its net's interconnect length would increase. The increased length is denoted as Interconnect Length Overhead (ILO).*

For every candidate floorplan, to meet the performance constraint, timing-critical modules are assigned a high voltage, and the other non-timing-critical modules are assigned a lower voltage to maximize power saving. Besides, each level-shifter is assigned to a rough position to minimize interconnect length overhead. We refer to the problem as the Voltage and Level-Shifter Assignment driven Floorplanning (VLSAF).

PROBLEM 1. *(VLSAF) We are given*

1) *A set of m modules: $N = \{n_1, n_2, \ldots, n_m\}$. Each module n_i is hard block(fixed size and aspect ratio), and is given two available supply voltage, and power -delay tradeoff is represented as a delay-power curve (DP–curve, as shown in Fig.3).*
2) *A netlist, which can be denoted as a directed acyclic graph(DAG), $\hat{G} = (\hat{V}, \hat{E})$, where $\hat{V} = \{n_1, n_2, \ldots, n_m\}$, and $e(i,j) \in \hat{E}$ denotes an interconnect from n_i to n_j.*
3) *A timing constraint T_{cycle}.*
4) *Level-shifter's area, power and delay, denoted as a_{ls}, p_{ls}, d_{ls}.*[1]

After VLSAF, a chip floorplanning is generated to meet several objectives: First, minimize the area and power cost. Secondly, satisfying timing constraint. Third, insert all the level-shifters in need and minimize the wire length and the interconnect length overhead.

[1]In this paper, only 2 voltages are supplied, so we assume all level shifters have the same area, power and delay.

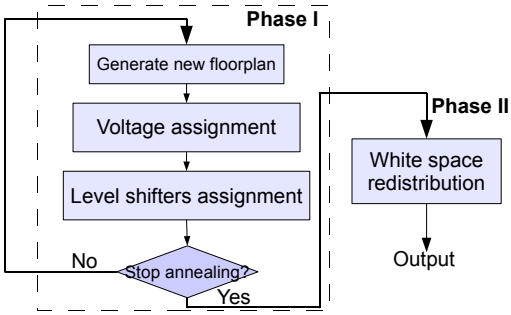

Figure 1: Overall of VLSAF

3. VLSAF ALGORITHM

3.1 Overview of VLSAF

As shown in Fig.1, algorithm VLSAF consists of two phases: (I)voltage and level-shifters assignment during floorplanning, (II) White Space Redistribution(WSR) at post-floorplanning.

In Phase I, we modify the model in [7] to handle voltage assignment and present a Min-Cost Max-Flow based method to solve the level-shifters assignment problem. When generate a new packing, we carry out voltage and level-shifter assignment. After voltage assignment(VA), each module is assigned a voltage to reduce power consumption as much as possible yet satisfies the performance constraint. After level-shifter assignment(LSA), as many level-shifters as possible are assigned a room to minimize Interconnect Length Overhead(ILO).

In Phase II, a heuristic method is adopted to calculate every module's relative position in room. Besides, every room's white space is divided into grids, and each level-shifter is decided its aspect ratio and inserted to a grid. Finally, if a level-shifter can not assign a room in LSA, it can be inserted into a room in order to reduce interconnect length overhead(ILO).

3.2 Voltage Assignment

During floorplanning, when a new floorplan is generated, we can estimate the interconnect length between module i and module j, denoted as len_{ij}. Similar to [7], len_{ij} can be scaled to delay $delay_{ij}$ according to $delay_{ij} = \delta \times len_{ij}$, where δ is a constant scaling factor. We check every $delay_{ij}$, if $delay_{ij} \geq T_{cycle}$, then time constraint can not be satisfied, so another new floorplan is generated. Otherwise we carry out voltage assignment.

Given netlist $\hat{G} = (\hat{V}, \hat{E})$, voltage assignment problem can be formulated as (1):

$$Minimize \sum_{i \in \hat{V}} P_i(d_i) \tag{1}$$

$$s.t. \begin{cases} \mu_j - \mu_i \geq delay_{ij} + d_i & \forall e(i,j) \in \hat{E} \quad (1a) \\ d_i \in \{d_i^1, d_i^2\} & \forall i \in \hat{V} \quad (1b) \\ 0 \leq \mu_i \leq T_{cycle} & \forall i \in \hat{V} \quad (1c) \end{cases}$$

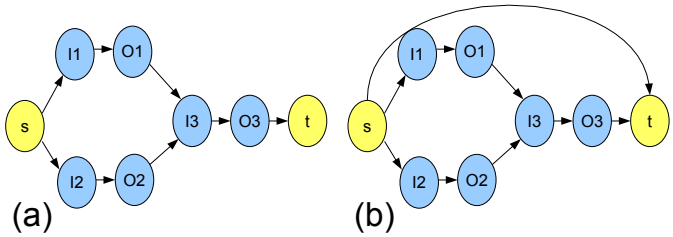

Figure 2: (a)$\bar{G} = \{\bar{V}, \bar{E}\}$, after adding nodes s, t and diving nodes N_i into I_i and O_i (b)Transformed $\bar{G} = \{\bar{V}, \bar{E}\}$ by adding edge $e(s,t)$ to remove constraint $\mu_t - \mu_s \leq T_{cycle}$ in equation (2).

where μ_i is the arrival time of vertex i in DAG, and d_i is the delay of vertex i.

To solve the formulation by network flow algorithm, we transform $\hat{G}$ into $\bar{G} = (\bar{V}, \bar{E})$. First, a start node s and an end node t are added to $\hat{V}$, s interconnect the nodes whose in-degree are zero, and nodes with zero out-degree interconnect t. We set $\bar{V} = \{s,t\} \cup \hat{V} = \{s, t, n_1, n_2, \ldots, n_m\}$. Besides, $n_i (i = 1, \ldots, m)$ are divided into two nodes: I_i and O_i, so $\bar{V} = \{s, t, I_1, O_1, I_2, O_2, \ldots, I_m, O_m\}$. And I_i is connected to O_i by a directed edge. We denote these new created edges $\{e(I_i, O_i) | I_i, O_i \in \bar{V}\}$ as $\bar{E}_1$, denote edges $\{e(s, I_k) | I_k \in \bar{V}\}$ as $\bar{E}_3$, and other edges as $\bar{E}_2$, and $\bar{E} = \bar{E}_1 \cup \bar{E}_2 \cup \bar{E}_3$. The DAG $\bar{G} = (\bar{V}, \bar{E})$ is shown in Fig. 2 (a).

The mathematical program is in (2),where d_{ij} is delay from node i to node j.

$$Minimize \sum_{e(i,j)\in\bar{E}} P_{ij}(d_{ij}) \tag{2}$$

$$s.t. \begin{cases} \mu_j - \mu_i \geq d_{ij} & \forall e(i,j) \in \bar{E} \quad (2a) \\ \mu_t - \mu_s \leq T_{cycle} & \quad (2b) \\ d_{ij} \in \{d_{ij}^1, d_{ij}^2\} & \forall e(i,j) \in \bar{E}_1 \quad (2c) \\ d_{ij} = delay_{ij} & \forall e(i,j) \in \bar{E}_2 \quad (2d) \\ d_{ij} = 0 & \forall e(i,j) \in \bar{E}_3 \quad (2e) \end{cases}$$

Compare with [7], which has more constraints as follows:

$$\begin{cases} 0 \leq \mu_i \leq T_{cycle} & \forall i \in \bar{V} \\ l_{ij} \leq d_{ij} \leq u_{ij} & \forall e(i,j) \in \bar{E} \end{cases}$$

we introduce some modifications. First, timing constraint used to be estimated as $T_{cycle} - L \times d_{ls}$, where L is the longest path in DAG. To reduce tolerance of timing constraint, in module's DP-curve, we add d_{ls} to lower voltage's delay and add p_{ls} to lower voltage's power(as shown in Fig. 3), and time constraint can be set as T_{cycle}. Since there are only two possible supply voltages, power function $P_{ij}(d_{ij})$ still be convex function. Secondly, we add start node s and end node t to remove constraint $0 \leq t_i \leq T_{cycle}$. Third, since DP-curve is a linear function, in other word, for $e(i,j) \in E_1$, d_{ij} has only two choices: d_{ij}^1 and d_{ij}^2. We can prove later that we can solve the program optimally even if we remove the constraint $l_{ij} \leq d_{ij} \leq u_{ij}$.

We can incorporate constraints $(2b)$ and $(2a)$ by transforming $(2b)$ into $\mu_s - \mu_t \geq -T_{cycle}$, and define d_{st}, s.t. $\mu_t - \mu_s = d_{st}$ & $d_{st} \leq T_{cycle}$. Accordingly, $\bar{E}_3 = \{\bar{E}_3 \cup e(s,t)\}$, and the transformed DAG $\bar{G}$ is shown in Fig.2(b). Besides, we dualize the constraints $(2a)$ using a nonnegative Lagrangian multiplier vector $\bar{x}$, obtaining the following Lagrangian subproblem:

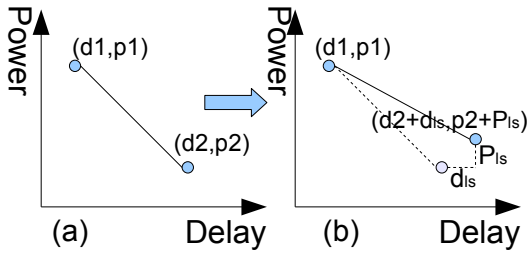

Figure 3: For a module, (a)original DP-curve, (b)modified DP-curve, adding the power and delay of level-shifter.

$$L(\vec{x}) = \min \sum_{e(i,j)\in\bar{E}} P_{ij}(d_{ij}) - \sum_{e(i,j)\in\bar{E}} (\mu_j - \mu_i - d_{ij}) x_{ij} \tag{3}$$

It is easy to show that

$$\sum_{e(i,j)\in\bar{E}} (u_i - u_j) x_{ij} = \sum_{i\in\bar{V}} x_{si} \times \mu_i \tag{4}$$

where

$$x_{si} = \sum_{j:e(i,j)\in\bar{E}} x_{ij} - \sum_{j:e(j,i)\in\bar{E}} x_{ji}, \forall i \in V \tag{5}$$

Lagrangian subproblem (3) can be restated as follows:

$$L(\vec{x}) = \min \sum_{e(i,j)\in\bar{E}} [P_{ij}(d_{ij}) + x_{ij} d_{ij}] + \sum_{i\in\bar{V}} x_{si}\mu_i \tag{6}$$

We set $V = \bar{V}$, remove $e(i,j) \in E_3$, and add an edge $e(s,i)$ for each node $i \in V$. The newly edges are denoted as E_3, and $E_1 = \bar{E}_1$, $E_2 = \bar{E}_2$. Now $E = E_1 \cup E_2 \cup E_3$, and the transformed DAG is denoted as $G = (V, E)$.

For every $e(s,i) \in E_3$, we set $d_{si} = \mu_i, P_{si}(d_{si}) = 0, l_{si} = 0, u_{si} = \begin{cases} K, & if \quad i \neq t \\ T_{cycle}, & if \quad i = t \end{cases}$, where k is a huge coefficient.

We define function $H_{ij}(x_{ij})$ for each $e(i,j) \in E$ as follows:

$$H_{ij}(x_{ij}) = \min_{dij}\{P_{ij}(d_{ij}) + x_{ij} d_{ij}\} \tag{7}$$

For the $e(i,j) \in E_1$, because $P_{ij}(d_{ij})$ is linear function

$$P_{ij}(d_{ij}) = -k \times d_{ij}, \quad d_{ij} \in [d_{ij}^1, d_{ij}^2] \tag{8}$$

where $k \geq 0$ and $-k$ denotes slope of the function, $k = \frac{P_{ij}(d_{ij}^1) - P_{ij}(d_{ij}^2)}{d_{ij}^2 - d_{ij}^1}$.

$$\begin{aligned} H_{ij}(x_{ij}) &= min\{P_{ij}(d_{ij}) + x_{ij} dij\} \\ &= min\{(x_{ij} - k) \times d_{ij}\} \\ &= \begin{cases} (x_{ij} - k) \times d_{ij}^2 & 0 \leq x_{ij} \leq k \\ (x_{ij} - k) \times d_{ij}^1 & k \leq x_{ij} \end{cases} \\ &= \begin{cases} P_{ij}(d_{ij}^2) + d_{ij}^2 x_{ij} & 0 \leq x_{ij} \leq k \\ P_{ij}(d_{ij}^1) + d_{ij}^1 x_{ij} & k \leq x_{ij} \end{cases} \end{aligned}$$

For the $e(i,j) \in E_2$, $H_{ij}(x_{ij}) = d_{ij} x_{ij}, x_{ij} \geq 0$.

For the $e(i,j) \in E_3$,$H_{ij}(x_{ij}) = \begin{cases} K_j \times x_{ij} & x_{ij} \leq 0 \\ 0 & x_{ij} \geq 0 \end{cases}$, where $K_j = T_{cycle}$ if $j = t$; and if $j \neq t$, K_j equals K.

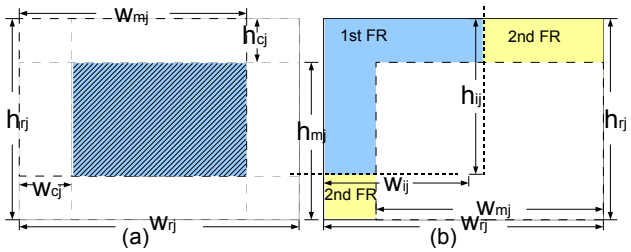

Figure 4: (a)No matter how to move the module, dark area can not insert level-shifter can not be inserted into the dark , while blank area is Potential White Space(PWS) of R_j (b)1st and 2nd Feasible Region of FR_{ij}.

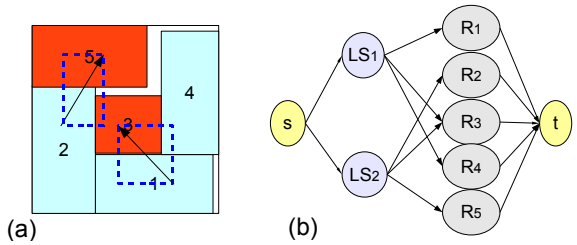

Figure 5: (a)LS1 drives from module 1 to module 3 and LS2 drives from module 2 to module 5. (b)Corresponding network graph, LS1 can be assigned to room 1,3,4, while LS2 can be assigned to room 2,3,5.

To transform the problem into a minimum cost flow problem, we construct an expanded network $G' = (V', E')$. There are three kinds of edges to consider:

- $e(i,j)$ in E1:we introduce 2 edges in G', and the costs of these edges are: $-d^2_{ij}, -d^1_{ij}$; upper capacities: $k, M-k$; lower capacities are both 0.
- $e(i,j)$ in E2: cost, lower and upper capacity is $-d_{ij}$, 0, M.
- Edge in E3: two edges are introduced in G', one with cost, lower and upper capacity as $(-K_j, -M, 0)$, another is $(0, 0, M)$.

Using the cost-scaling algorithm, we can solve the minimum cost flow problem in G'. For the given optimal flow x^*, we construct residual network $G(x^*)$ and solve a shortest path problem to determine shortest path distance $d(i)$ from node s to every other node. By implying that $\mu(i) = d(i)$ and $d_{ij} = \mu(i) - \mu(j)$ for each $e(i,j) \in E_1$, we can finally solve voltage assignment problem.

3.3 Level Shifters Assignment

After voltage assignment, the number of level-shifters n_{ls} is determined, and the chip is dissected into set of rooms $R = \{r_1, r_2, \ldots, r_m\}$. Then we carry out level-shifters assignment to try to assign every level-shifters one room. We define sets of level shifters $LS = \{LS_1, LS_2, \ldots, LS_n\}$, every set $LS_i (i = 1, \ldots, n)$ contain $size_i$ level shifters with same source module and the same sink module, and $\sum_{i=1}^{n} size_i = n_{ls}$.

To check whether a room has extra space to insert level-shifter, we denote the White Space in room r_j as ws_j, whose area can be calculated as follow:

$$Area(ws_j) = w_{rj} \times h_{rj} - w_{mj} \times h_{mj} \tag{9}$$

where $w_{rj}(h_{rj})$ denotes the width(height) of room r_j, $w_{mj}(h_{mj})$ denotes the width(height) of module n_j.

Each level-shifter belongs to a net, and is inserted into white space. Since we assume all modules are hard blocks, some space of room must belong to a module(as shown in Fig.4(a), center dashed area can not insert level shifter no matter how to put the module).

Definition 4 (Potential White Space (PWS)). *The space of room r_j can be white space through module moving is denoted as Potential White Space(pws_j).*

pws_j can be considered as two horizontal channels and two vertical channels, as shown in Fig. 4 (a), we denote the width of vertical channel as $w_{cj} = w_{rj} - w_{mj}$, and the width of horizontal channel as $h_{cj} = h_{rj} - h_{mj}$.

Definition 5 (Feasible Region (FR)). *For a net requiring level shifter i, its bounding box is denoted as b_i, we define level-shifter i's feasible region as FR_i and $FR_i = \{ws_j | \forall j, b_i \cap r_j \neq 0\}$.*

For room r_j, if its white space ws_j belongs to level-shifter i's Feasible Region FR_i, the part of ws_j in b_i is denoted as 1st Feasible Region($fr1_{ij}$), while the other part of ws_j is denoted as 2nd Feasible Region($fr2_{ij}$).

We set $w = max(w_{ij} - w_{cj}, 0)$ and $h = max(h_{ij} - h_{cj}, 0)$, then the area of $fr1_{ij}$ can be calculated as follows:

$$Area(fr1_{ij}) = w_{ij} \times h_{ij} - w \times h \tag{10}$$

We construct a network graph $G^* = (V^*, E^*)$, and then use a min-cost max-flow algorithm to determine which room each level shifter belong to. A simple example is shown in Fig.5.

- $V^* = \{s, t\} \cup LS \cup R$.
- $E^* = \{(s, LS_i) | LS_i \in LS\} \cup \{(LS_i, r_j) | \forall fr_{ij}\} \cup \{(r_j, t) | r_j \in R\}$.
- Capacities: $C(s, LS_i) = size_i, C(LS_i, r_j) = size_i, C(r_j, t) = \frac{Area(ws_j)}{a_{ls}}$.
- Cost: $F(s, LS_i) = 0, F(r_j, t) = 0; F(LS_i, r_j) = F_{ij}$, which will discussed below.

We define area percent of $fr1_{ij}$ as p_{ij}, $0 \leq p_{ij} \leq 1$.

$$p_{ij} = \begin{cases} \frac{Area(fr1_{ij})}{Area(ws_j)}, & Area(ws_j) \neq 0 \\ 0, & others \end{cases} \tag{11}$$

Define cost of edge $e(LS_i, r_j)$, F_{ij} is a function of p_{ij}:

$$F_{ij}(p_{ij}) = \lceil \frac{1}{p_{ij} + \mu} + (1 - p_{ij}) \times k \times (Term1_{ij} + Term2_{ij}) \rceil \tag{12}$$

where μ is a small coefficient, k is a undetermined coefficient and $Term1_{ij}, Term2_{ij}$ is penalty terms, and $Term1_{ij} = \begin{cases} \frac{h_{rj} - hij}{w_{cj}}, & w_{cj} \neq 0 \\ 0, & w_{cj} = 0 \end{cases}$, $Term2_{ij} = \begin{cases} \frac{w_{rj} - w_{ij}}{h_{cj}}, & h_{cj} \neq 0 \\ 0, & h_{cj} = 0 \end{cases}$.

Equation (12) has some special characters. First, it is a monotonically decreasing function of p_{ij}, which means we are inclined to put level-shifter in the room which has higher percentage of 1st FR. Besides, it can not be too large even $fr1_{ij}$ is very small, so we add coefficient μ and $maxF_{ij}(p_{ij}) \simeq \lceil \frac{1}{\mu} \rceil$. Third, we observe that even two room have the same p_{ij} and $p_{ij} \leq 1$, if level shifter is inserted in $fr2_{ij}$, the room has longer $fr2_{ij}$ may cause longer length. Consequently, in equation (12), we add the penalty term $Term1_{ij}$ and $Term2_{ij}$.

3.4 White Space Redistribution (WSR)

After floorplanning, most level-shifters are assigned to rooms. We define set ELS containing level-shifters that can not be assigned to a room. In room r_j, we should pack a module n_j, and a group of level shifters $Ls_j = \{ls_1, ls_2, \ldots, ls_{pi}\}$, and the condition below must be satisfied.

$$Area(n_j) + \sum_{k=1}^{pi} Area(ls_k) \leq Area(r_j)$$

Traditional room-based floorplanner will pack the modules at the lower-left corner or the center of the rooms. Different from the traditional block planning method, to favor the level-shifters insertion, a heuristic method(called as WSR) is adopted to calculate modules' and level-shifters' relative positions in rooms. The framework of algorithm WSR is below:

Algorithm WSR:
1. For each module n_j, calculate its relative position in r_j.
2. For each room r_j, generate grids in white spaces, and then insert Ls_j into grids.
3. Insert level-shifters in ELS.
4. Move modules again under demand of Power Network.

3.4.1 Relative Position Calculation

If a level-shifter ls_i is assigned into room r_j, a prefer region is provided. If ls_i is inserted in the prefer region, then interconnect would not lengthen. For each level-shifter to insert in room r_j, a force is produced to push the module n_j apart from the level-shifter. We consider the force produced by ls_i in x- and y-direction separately, denoted as F_{ix} and F_{iy}. For example, as shown in Fig. 6(a), if ls_i prefers to locate in the lower-left corner of r_j, then F_{ix} pushes n_j to right and F_{iy} pushes n_j to upper. To calculate F_{ix} and F_{iy}, prefer area is defined as a quaternion $(w_{1ij}, w_{2ij}, h_{1ij}, h_{2ij})$, where $w_{1ij}(w_{2ij})$ is the distance from prefer area to left(right) boundary of r_j, $h_{1ij}(h_{2ij})$ is the distance from prefer area to upper(lower) boundary of r_j, as shown in Fig. 6(b).

F_{ix} and F_{iy} can be calculated as equation (13).

$$F_{ix} = \frac{w_{2ij} - w_{1ij}}{w_{rj}}, \quad F_{iy} = \frac{h_{2ij} - h_{1ij}}{h_{rj}} \tag{13}$$

To calculate the position of module n_j, we define four variables $F_{right}, F_{left}, F_{up}, F_{down}$ as follows:

$$\begin{cases} F_{right} & = \sum_i F_{ix}, \quad \forall F_{ix} \geq 0 \\ F_{left} & = \sum_i F_{ix}, \quad \forall F_{ix} < 0 \\ F_{up} & = \sum_i F_{iy}, \quad \forall F_{iy} \geq 0 \\ F_{down} & = \sum_i F_{iy}, \quad \forall F_{iy} < 0 \end{cases} \tag{14}$$

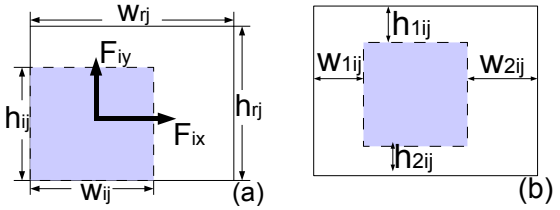

Figure 6: In room r_j, (a)if level-shifter ls_i prefers to locate in lower-left corner (dark area is prefer region), then ls_i produces forces (F_{ix}, F_{iy}) to pushes module n_j upper and right. (b)$w_{1ij}, w_{2ij}, h_{1ij}, h_{2ij}$ are defined to calculate forces (F_{ix}, F_{iy}).

Relative position of n_j in room r_j is denoted as (X_{nj}, Y_{nj}), then

$$X_{nj} = \frac{(w_{rj} - w_{mj}) \times F_{right}}{F_{right} - F_{left}} \tag{15}$$

$$Y_{nj} = \frac{(h_{rj} - h_{mj}) \times F_{up}}{F_{up} - F_{down}} \tag{16}$$

3.4.2 Grids Generation and LS Insertion

In room r_j, after calculating module n_j's relative position, at most four rectangular white spaces are generated. We divide each white spaces into rectangular grids, whose area are all a_{ls}. So room r_j records a set of grids $G_j = \{g_1, g_2, \ldots, g_m, m \times a_{ls} \leq Area(r_j) - Area(n_j)\}$, and each grid has its position. Level-shifters in set Ls_j are sorted by area of prefer region. Smaller prefer region, higher priority. Then each level-shifter picks one grid in order.

After every level-shifter assigned choosing a grid, each level-shifter in ELS picks free grid to insert for the purpose of minimizing wire length.

Finally, in room r_j, if not all the grids are inserted by level shifter, module n_j may remove. If n_j is in low voltage, it removes toward left and down to reduce total area, while if n_j is in high voltage, it removes toward the center of power network to minimize power network resource.

4. EXPERIMENTAL RESULTS

We implemented algorithm VLSAF in the C++ programming language and executed on a Linux machine with a 3.0GHz CPU and 1GB Memory. Fig. 7 shows the experimental results of the benchmarks n50 and n200.

In simulated annealing, we use the cost function as follow:

$$\Phi = \lambda_A A + \lambda_W W + \lambda_P P + \lambda_R R + \lambda_N N \tag{17}$$

where A and W represent the floorplan area and wire length; P represents the total power consumption; R represents the power network resource; and N records the number of level shifters that can not be assigned. When temperature is higher than a coefficient we assume all the level shifters can be assigned, so λ_N is set 0 at the beginning of annealing.

The previous work [5] is the recent one in handling floorplanning problem considering voltage assignment and level-shifter insertion. Table 1 shows comparisons between our experimental result and [5]. The column Power Cost means the actual power consumption, column PNR means power network resource consumption. VLSAF can save 17% power

Table 1: The Comparison Between the VLSAF and the Previous Work

Benchmark	Max Power	Power Cost		PNR		LS Number		White Space (%)		Run Time(s)	
		[5]	VLSAF	[5]	VLSAF	[5]	VLSAF	[5]	VLSAF	[5]	VLSAF
n10	216841	216840	189142	965	1007	0	9	4.87	9.44	6.001	3.24
n30	205650	190717	146483	1369	1436	57	25	9.03	11.32	115.07	35.11
n50	195140	172884	135316	1514	1460	119	114	21.10	16.66	569.36	116.97
n100	180022	179876	123526	1671	1354	92	153	34.07	26.71	1768	688.13
n200	177633	174818	130050	2040	1763	399	203	46.52	29.66	4212	1969.12
n300	273499	219492	234389	2147	1997	452	337	44.10	37.74	4800	2392.8
Avg	-	192438	159818	1617.7	1502.8	186	140.2	26.61	21.92	1911.74	857.56
Diff	-	-	-17%	-	-7.2%	-	-24.7%	-	-17.6%	-	-55.2%

Table 2: VLSAF v.s. VAF+LSI

	Wire Length w. LS		ILO(%)		White Space(%)		Run Time(s)	
	VLSAF	VAF+LSI	VLSAF	VAF+LSI	VLSAF	VAF+LSI	VLSAF	VAF+LSI
n10	13552	17937	0.89	2.29	9.44	10.46	3.24	2.09
n30	44225	43282	0.31	0.85	11.32	10.75	35.11	23.13
n50	92678	95666	1.20	2.27	16.66	18.12	116.97	39.81
n100	185622	191522	1.03	2.40	26.71	26.40	688.13	327.01
n200	366003	365792	1.64	4.28	29.66	30.06	1969.12	1304.3
n300	560042	600348	0.67	1.37	37.74	35.36	2392.8	1772.03
Avg	210404	219091	0.96	2.24	21.92	21.86	857.56	578.06
Diff	-	+4%	-	+133%	-	-0.3%	-	-32.5%

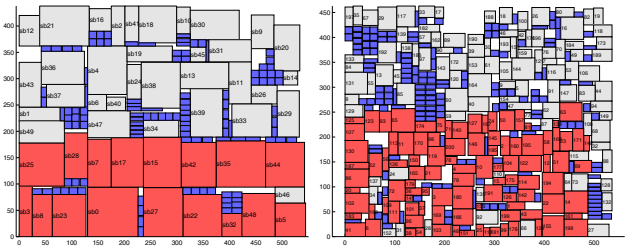

Figure 7: Experimental results of n50 and n200

and 7.2% PNR. The column White Space and the column Run Time show our framework is about 2X faster while white space can be saved by 17.6%.

In Table 2, we compare VLSAF with VAF+LSI, which solves level-shifter assignment and insertion only at post-floorplan stage. We can see that in VAF+LSI, wire length and interconnect length overhead(ILO) will be increased by 4% and 133%. High ILO may cause delay estimation among modules inaccurate, or even lead to timing constraint violation. Accordingly, VLSAF is effective and significant with a reasonable more runtime.

5. CONCLUSIONS

We have proposed a two phases framework to solve voltage assignment and level shifter insertion: phase one is voltage and level-shifter assignment driven floorplanning; phase two is white space redistribution at post- floorplanning stage. Experimental results have shown that our framework is effective in reducing power cost while considering level shifters' positions and areas.

6. REFERENCES

[1] David Lackey, Paul Zuchowski and J. Cohn. Managing power and performance for system-on- chip designs using voltage islands. *ICCAD*, pages 195–202, 2002.

[2] Xianlong Hong, Sheqin Dong. Non-slicing floorplan and placement using corner block list topological representation. *IEEE Transaction on CAS*, 51:228–233, 2004.

[3] M.Hamada and T.Kuroda. Utilizing surplus timing for power reduction. *CICC*, pages 89–92, 2001.

[4] W.L.Hung, G.M.Link and J.Conner. Temperature-aware voltage islands architecting in system-on-chip design. *ICCD*, 2005.

[5] W.P.Lee and Y.W.Chang. Voltage island aware floorplanning for power and timing optimization. *ICCAD*, pages 389–394, 2006.

[6] J.Hu, Y.Shin and R.Marculescu. Architecting voltage islands in core-based system-on-a-chip designs. *ISLPED*, pages 180–185, 2004.

[7] Q.Ma and F.Y.Young. Network flow-based power optimization under timing constraints in msv-driven floorplanning. *ICCAD*, 2008.

[8] W.K.Mak and J.W.Chen. Voltage island generation under performance requirement for soc designs. *ASP_DAC*, 2007.

[9] W.P.Lee and Y.W.Chang. An ILP algorithm for post-floorplanning voltage-island generation considering power-network planning. *ICCAD*, pages 650–655, 2007.

[10] H.Wu, I.M.Liu and Y.Wang. Post-placement voltage island generation under performance requirement. *ICCAD*, 2005.

[11] R.Ching and F.Y.Young. Post-placement voltage island generation. *ICCAD*, 2006.