

# Logic and Layout Aware Voltage Island Generation for Low Power Design

Liangpeng Guo, Yici Cai, Qiang Zhou, Xianlong Hong

EDA Lab, Department of Computer Science and Technology, Tsinghua University, Beijing, China

**Abstract**— Multiple supply voltage (MSV) is one of the most effective schemes to achieve low power, but most works are based on logic level. A few recent works are based on physical level but all of them do not consider level converters which have an important effect in dual-vdd design. In this work we propose a logic and layout aware approach for voltage assignment and voltage island generation in placement process to minimize the number of level converters and to implement voltage islands with minimal overheads. Experimental results show that our approach uses much less level converters than the approach in [1] (reduced by 59.50% on average) when achieving the same power savings. The approach is able to produce feasible placement with a small impact to traditional placement goals.

## I. INTRODUCTION

As VLSI technology advances, power dissipation has become one of the most challenging constraints in modern IC design. Since the dynamic power is proportional to the square of supply voltage, reducing supply voltage can significantly lower the dynamic power consumption. To reduce total power without changing circuit performance, multiple supply voltage (MSV) has been widely used, which applies the reduced voltage only to the nodes on the non-critical paths.

Related algorithms have been heavily studied, but most works are based on logic level. Usami and Horowitz first proposed the idea of clustered voltage scaling (CVS) [2] and then lower power is achieved by Extended Clustered Voltage Scaling (ECVS) [3]. Following work further reduced the power by translating the power reduction into maximal-weighted-independent-set problem [4] or using an efficient heuristic methods [5]. Other successful solutions include any combination of supply voltage scaling, threshold voltage scaling, gate-sizing and retiming [6, 7].

Though existing algorithms above can reduce power effectively, overheads increase significantly when arranging  $V_{ddH}$  cells and  $V_{ddL}$  cells separately in interleaving rows or half rows [8]. Even in 'region based' design style which is considered to be more flexible, clustering the cells with predefined assignment of voltages is likely to cause large wirelength and delay penalty. On the other side, since interconnect delay dominates the gate delay, critical paths

cannot be identified until the cell locations are approximately determined. In addition to that, the power consumed by nets should not be ignored. Thus it is more suitable to exploit the slack for power savings at physical level.

The root of large overheads stems from voltage scaling at logic level as logical boundaries limit the flexibility of placement. To solve the problem, two very recent work [1, 9] introduce physical boundaries in voltage partition to generate voltage islands. However, a large number of level converters (LC) are needed to be inserted at the logical boundaries where a  $V_{ddL}$  cell drives a  $V_{ddH}$  cell. Since level converters (LC) consume additional power, introduce additional delay and occupy more area [10], they decrease the advantage of voltage scaling to a large extent. So nearly all algorithms at logic level try to minimize the number of LCs, which is not taken into consideration in the only works at physical level [1, 9].

Logical or physical boundaries have their own disadvantages. In the method based on logical boundaries, to minimize the number of level converters (LC), voltage scaling algorithm tends to group  $V_{ddH}$  ( $V_{ddL}$ ) cells together in signal paths. However, these cells are often scattered in placement. In order to form a voltage island, the placer clusters the  $V_{ddH}$  ( $V_{ddL}$ ) cells. Since a  $V_{ddH}$  ( $V_{ddL}$ ) island must contain  $V_{ddH}$  ( $V_{ddL}$ ) cells purely, a strong force is needed to move the  $V_{ddH}$  ( $V_{ddL}$ ) cells to the confined region while moving the  $V_{ddL}$  ( $V_{ddH}$ ) cells out, which leads to a significant impact to traditional placement goals. In the method based on physical boundaries, regions are selected to be the  $V_{ddH}$  ( $V_{ddL}$ ) islands, which involve a lot of cells that are not adjacent in signal paths and lead to a number of LCs needed to be inserted.

To tackle this problem, we propose a logic and layout aware approach for voltage assignment and voltage island generation in placement process. Firstly, we integrate the algorithm at logic level to produce the initial voltage assignment while considering interconnect power and delay. Secondly, we propose a logic and layout aware voltage assignment algorithm combined with placement to cluster the cells with the same supply voltage ( $V_{ddL}$  or  $V_{ddH}$ ) in both logical and physical spaces, which helps to generate voltage islands without increasing the number of level converters. Thirdly, we develop a new soft clustering strategy based on placement feedback to exploit voltage island boundaries with minimal wirelength penalty. Experimenten-

tal results on a set of MCNC benchmarks show that our approach can significantly reduce total power without loss of performance and uses much less level converters.

The rest part of paper is organized as follows: Section II gives a brief review of level conversion and top-down min-cut placement. Section III describes our design flow of voltage island generation. Section IV, Section V and Section VI present our approach in three parts: initial voltage assignment, logic and layout aware voltage assignment and soft clustering separately. At the end of this paper, experimental results and conclusion are presented.

## II. BACKGROUND

### A. Level Conversion

Since the logic 'HIGH' signal of the  $V_{ddL}$  cell cannot completely turn off the PMOS pull-up network of the subsequent  $V_{ddH}$  cell without level conversion, the circuit requires level converters (LC) at the boundaries where a  $V_{ddL}$  cell drives a  $V_{ddH}$  cell. We utilize a kind of level converter (LC) which requires only one supply voltage and can be placed as ordinary  $V_{ddH}$  cells [11]. The functionality of level conversion implemented at flip-flop boundaries can be embedded into the flip-flop circuit [10]. The special flip-flop which integrated with a level converter is called level converting flip-flop.

### B. Top-Down Min-Cut Placement

This work is based on Capo [12], which is a multilevel cut-based placement tool. Initially, the placement region is represented as a block (referred as to bin) and the circuit is represented as a hypergraph. The placer recursively divides each bin, partitions its associated hypergraph and assigns the subhypergraphs to subbins, aiming at minimizing the total number (weight) of nets incident to nodes in multiple partitions for total wirelength reduction. The top-down min-cut placement methodology proceeds from a coarse top level to a fine bottom level.

The placement feedback is a novel technique which improves the total wirelength by merging and repartitioning bins under guidance of information from previous partition results [13]. The idea that improves present determination quality by using future information is an inspiration of our soft clustering strategy, which is described in details in Section VI.

## III. VOLTAGE ISLAND GENERATION METHODOLOGY

The voltage island generation algorithm is established upon a partition based placement platform. It consists of three primary processes:

- **Process 1** Initial voltage assignment based on the slacks of the circuit nodes. The purpose of this process is to assign as many nodes as possible to  $V_{ddL}$  without impairing the minimum slack.

- **Process 2** Logic and layout aware voltage assignment where some of the  $V_{dd}$  assignments in the previous process are flipped to reduce the number of level converters while keeping the cells with the same supply voltage within the the same local area.
- **Process 3** Voltage island generation based on soft clustering where small adjacent placement bins are merged and re-partitioned so that cells with the same  $V_{dd}$  assignment tend to reside in the same bin.

The three processes are combined with the partition based placement. The methodology could be summarized by a high level pseudo-code as follows (to keep it simple, the details of placement are omitted):

### Voltage Island Generation Methodology

```

while (!done)
  if ( $m > m_t$ )
    traditional placement process;
  end if
  if ( $m = m_t$ )
    Process 1; Process 2; Process 3;
  end if
  if ( $m < m_t$ )
    Process 2; Process 3;
  end if
end while

```

where the out most loop corresponds to the partition hierarchy;  $m$  is the current level of the partition;  $m_t$  is a constant determined by the scale of circuit. The traditional placement steps are performed at the first  $m_t - 1$  levels of the partition, which aims at producing a coarse placement. The Process 1 is performed only once and the last two processes are performed at every level of the rest of the partition to generate voltage islands with minimal number of LCs and physical overheads.

## IV. COARSE PLACEMENT AND INITIAL VOLTAGE ASSIGNMENT

As interconnect delay dominates gate delays, we need a coarse placement to estimate wirelength and interconnect delay. In top-down placement, the coarse placement is accomplished when the long nets that are often responsible for critical paths can be identified. This process is like the first several passes of a global placement with all cells operating under  $V_{ddH}$ . As described in [1], it is useful to employ a timing optimization strategy in coarse placement. But to obtain an accurate wirelength penalty caused by our algorithm, the experiments in Section VII are based on a version (CapoLLV) without any timing optimization strategy. Because the net weighting strategy [1] for timing optimization introduces additional wirelength penalty and this penalty is sometimes very large [1].

The objective of initial voltage assignment is to identify the cells which tend to operate at  $V_{ddL}$ . Since the

distribution of  $V_{ddL}$  cells in placement largely defines the pattern of voltage islands, it is desirable to allow as many cells as possible working at  $V_{ddL}$ . To make full use of timing slack, we implement a heuristic algorithm similar to [4] while considering interconnect power and delay.

In static timing analysis, the circuit can be represented by a graph  $G = (V, E)$ . We denote  $a(v)$ ,  $r(v)$  and  $slack(v)$  as the arrival time, required time and slack of node  $v$  respectively. We denote  $d(v)$  as the delay of node  $v$  and denote  $d(u, v)$  as the wire delay between  $u$  and  $v$ . An edge  $(u, v) \in E$  is called sensitive if either  $a(v) - a(u) = d(v) + d(u, v)$  or  $r(v) - r(u) = d(v) + d(u, v)$ . A directed path is called sensitive if the path consists of only sensitive edges and the slacks of all nodes on the path are monotonously distributed in the direction of the path. The sensitive transitive closure graph  $G_s = (V, E_s)$  of  $G$  is a directed graph such that there is an edge  $(u, v) \in E_s$  if and only if there is a directed sensitive path from  $u$  to  $v$  in  $G$ . Two nodes  $u, v \in V$  are called slack sensitive if there exists a sensitive path from  $u$  to  $v$  or from  $v$  to  $u$  in  $G$ . Otherwise, they are called slack insensitive.

Suppose the delay of a node increases by a small positive amount  $\epsilon > 0$ , then its slack  $slack(v)$  is reduced by exactly  $\epsilon$ . The decreases in slack of other nodes can be summarized as follows:  $\forall u \in V$ ,  $slack(u)$  remains unchanged if there is no directed path from  $u$  to  $v$  or from  $v$  to  $u$ ;  $slack(u)$  decreases by  $\epsilon$  if  $u, v$  are slack sensitive and  $slack(u) \geq slack(v)$ ;  $slack(u)$  decreases by  $slack(u) - slack(v) + \epsilon$  if  $u, v$  are slack sensitive and  $slack(v) - \epsilon < slack(u) < slack(v)$ ;  $slack(u)$  remains unchanged if  $u$  and  $v$  are slack insensitive or  $slack(u) \leq slack(v) - \epsilon$ . These properties have been proved in [4].

We construct the subgraph  $G_m = (Q_m, E_m)$  of  $G_s$ , where  $Q_m$  is the set of nodes with the maximum slack. According to the properties above, if we increase the delay of any node  $v \in Q_m$  by  $\epsilon < slack_{max} - slack_{max-1}$ <sup>1</sup>, only the slack of  $v$ 's neighbors in  $G_m$  decreases by  $\epsilon$ , the slack of other nodes keep unchanged. Because if any node  $w$  is not neighbors of  $v$ , then  $w, v$  are slack insensitive or  $slack(w) < slack(v) - \epsilon$ .

This motivates us to select the nodes that give the best power saving per unit delay penalty in  $Q_m$  and minimize the number of nodes whose slacks are reduced. If we relate the node weight to the power reduction under unit delay penalty, this can be achieved by finding a set of nodes that are independent, not pairwise connected by an edge, and such that the sum of their weights is maximum, which is known as maximal-weighted-independent-set problem. A heuristic measure for this problem is given as follows:

$$tendency(v) = \frac{\Delta power}{\Delta delay \cdot N_n(v)} \quad (1)$$

<sup>1</sup> $slack_{max}$  and  $slack_{max-1}$  denote the largest and the second largest slack in  $G$  separately.

where  $\Delta power$  is the change in the total power of the cell  $v$  and related wires,  $\Delta delay$  is  $\epsilon < slack_{max} - slack_{max-1}$  and  $N_n(v)$  is the number of neighbors of  $v$  in  $G_m$ .

At every iteration of the algorithm, we select the node  $v$  with the maximum  $tendency(v)$  in  $Q_m$  and increase the delay of  $v$  by  $\epsilon$ . Then we update related slack,  $Q_m$  and the state of the circuit. The nodes are assigned  $V_{ddL}$  when their delays are large enough. This process continues until  $\epsilon = 0$  when the circuit comes to a state that every node has increased its delay as much as possible without decreasing the minimum slack.

## V. LOGIC AND LAYOUT AWARE VOLTAGE ASSIGNMENT

After the initial voltage assignment, the cells under the same voltage are scattered on the placement region. Clustering these cells is likely to cause large wirelength and delay penalty even if flexible clustering strategies are employed in the placer. So supply voltage adjustments are needed to fit the layout. To minimize the number of LCs, we must consider both logical and physical adjacency.

### A. Heuristic Measure for Reducing the Number of LCs

The circuit can be represented as a hypergraph  $G(V, E)$ . The vertex set  $V$  represents the cells in the circuit and the hyperedge set  $E$  represents the connection of the circuit.

We denote  $Snk(e), e \in E$  as the set of sink nodes of super edge  $e$  and denote  $s_e \in V$  as source node of edge  $e$ .  $x_v$  is a boolean variable indicating the voltage assignment of node  $v$ , where  $x_v = 0$  and 1 correspond to  $V_{ddL}$  and  $V_{ddH}$  respectively.

We denote  $LC(e)$  as level converter assignment on edge  $e$ , where  $LC(e) = 1$  means LC is used on  $e$  and  $LC(e) = 0$  without LC. If we assume the voltages of nodes on the same edge are assigned independently, the expectation of  $LC(e)$  could be expressed as follows:

$$E(LC(e)) = 1 \cdot P(x_{s_e} = 0) \left(1 - \prod_{v \in Snk(e)} P(x_v = 0)\right) \quad (2)$$

where  $P(x_{s_e} = 0)$  is the probability that the source node is assigned  $V_{ddL}$ .  $(1 - \prod_{v \in Snk(e)} P(x_v = 0))$  is the probability that at least one sink is assigned  $V_{ddH}$ .

If we modify the supply voltage of node  $v$ , the possible level converter reduction on edge  $e$  ( $v$  is a source or sink node of  $e$ ) can be obtained by calculating  $E(LC(e))$  on the condition that  $v$  operates at different voltages. For instance, if  $v$  is a source node for edge  $e$  and the voltage of  $v$  is modified from  $V_{ddH}$  to  $V_{ddL}$ ,  $E(LC(e))$  changes from 0 to  $1 - \prod_{u \in snk(e)} P(x_u = 0)$ ; if  $v$  is a sink node for edge  $e$  and the voltage of  $v$  is modified from  $V_{ddH}$  to  $V_{ddL}$ ,  $E(LC(e))$  changes from  $P(x_{s_e} = 0)$  to  $P(x_{s_e} = 0)(1 - \prod_{u \in Snk(e), u \neq v} P(x_u = 0))$ . The reduction of  $E(LC(e))$  can be expressed as follows:

$$\Delta E(LC(e)) = E(LC(e)|x_v) - E(LC(e)|\bar{x}_v) \quad (3)$$

where  $E(LC(e)|x_v)$  is the expectation of  $LC(e)$  on the condition that  $v$  operates at current voltage and  $E(LC(e)|\bar{x}_v)$  is the expectation of  $LC(e)$  on the condition that  $v$  operates at the changed voltage.

On the assumption that every node has the same probability  $p$  to keep the current voltage and  $1 - p$  to be modified,  $P(x_v = 0) = 1 - p$  if  $v$  is a  $V_{ddH}$  cell and  $P(x_v = 0) = p$  if  $v$  is a  $V_{ddL}$  cell. Thus we can calculate  $\Delta E(LC(e))$  to evaluate any voltage modification by Equ. 2 and Equ. 3. This model has some good properties: (a)  $\Delta E(LC(e))$  is the number of LCs on edge  $e$  immediately removed by a modification when  $p = 1$ ; (b)  $\Delta E(LC(e)) \geq 0$  when one sink is modified from  $V_{ddH}$  to  $V_{ddL}$  but some other sinks of  $e$  are still  $V_{ddH}$  (the source is  $V_{ddL}$ ). Though no LC on  $e$  is removed immediately by the modification, the LC tends to be removed easier in future and the more  $V_{ddL}$  sinks on  $e$  remain, the less  $\Delta E(LC(e))$  we gain from the modification; (c)  $\Delta E(LC(e))$  is larger when the source node modified (from  $V_{ddL}$  to  $V_{ddH}$ ) has more  $V_{ddL}$  sinks. Therefore, we can regard  $\Delta E(LC(e))$  as the extent to which the modification helps in decreasing the number of LCs.

### B. Constrained Logic and Layout Aware Voltage Assignment

The main objective of our voltage assignment is to cluster cells under the same voltage in both logical and physical space. To evaluate the impact of any modification, we define the profit for a cell  $v$  as follows:

$$profit(v) = (\Delta ELC(v) + \alpha) \cdot N_v \quad (4)$$

where  $\Delta ELC(v)$  is the sum of  $\Delta E(LC(e))$ ,  $e \in E$  when the voltage of  $v$  is changed, which evaluates not only the number of LCs that are immediately removed but also the tendency of the reduction. From  $\Delta ELC(v)$ , we can identify the voltage modifications that remove or help remove LCs. Though a modification with a negative  $\Delta ELC(v)$  implies the tendency of increasing the number of LCs, accepting a modification with negative  $\Delta ELC(v)$  usually introduces more nodes with positive  $\Delta ELC(v)$ . So  $\alpha$  is used to allow a modification with negative  $\Delta ELC(v)$ , thus opening possibility of uncovering better solutions.  $N_v$  is the number of cells with the other supply voltage in the  $v$ 's bin. From the definition of profit, we observe that  $\Delta ELC(v)$  provides us the candidates of voltage modifications that decrease or keep the number of LCs and  $N_v$  gives a priority to every candidate that the modification of making a bin contain purely  $V_{ddL}$  or  $V_{ddH}$  cells can first be selected. The profit function is aware of logic and layout information, because  $\Delta ELC(v)$  and  $N_v$  respectively reflects the voltage assignment in the signal paths (logical space) and placement (physical space).  $\alpha$  is used to leverage the two aspects: when  $\alpha$  is small (e.g.  $\alpha = 0$ ), the determined factor is  $\Delta ELC(v)$ , which determines whether the *profit* is positive or negative, because  $N_v$  is always positive; when  $\alpha$  is large enough (e.g.

$\alpha = \max_v(|\Delta ELC(v)|)$ ), the determined factor is  $N_v$ , because one cell is usually related to only several LCs while a bin usually contains at least dozens of cells.

At each stage of the algorithm, we calculate Equ. 4 for all cells, select the cell with the maximum value and modify the voltage. Note that every modification may change level converter assignment in the vicinity. The arrival time, required time, slack and overall power dissipation will also change. All these must be updated before the next iteration.

Our algorithm can be summarized in pseudo-code form as follows:

### Constrained Logic and Layout Aware Voltage Assignment Algorithm

**Require:** voltage-scaled solution  
 calculate profit for all nodes  
**while** (*profit* > 0 and the modification does not violate timing or power constraint)  
   choose the gate with the maximum profit;  
   make the modification;  
   update the delay, slack;  
   update the profits for nodes in the vicinity;  
**end while**

The voltage assignment is performed at each placement level to cluster the cells under the same voltage while not increasing the number of LCs. As the bins get smaller (associated with less cells), we increase  $\alpha$  to further exploit physical adjacency. To avoid exploiting over trivial voltage boundaries, it is necessary for a small bin to contain  $V_{ddH}$  cells or  $V_{ddL}$  cells purely. So when the bin is small enough at the bottom level, for every bin with the portion of  $V_{ddH}$  larger than a threshold, all other  $V_{ddL}$  cells are replaced by  $V_{ddH}$  cells even a few LCs must be added.

## VI. VOLTAGE ISLAND GENERATION BASED ON SOFT CLUSTERING

Though the logic and layout aware voltage assignment tries to boost existing dominance of  $V_{ddL}$  or  $V_{ddH}$  cells in each bin. The soft clustering technique helps to generate voltage islands. For instance, a bin contains 60%  $V_{ddL}$  cells and replacing other cells with  $V_{ddL}$  cells may violate timing constraint or add a large number of LCs. If the placer partition the bin into one subbin contains 90%  $V_{ddL}$  cells and the other contains 30%  $V_{ddL}$  cells, it is more convenient for the voltage assignment at the next placement level to make the subbins contain  $V_{ddH}$  or  $V_{ddL}$  cells purely.

An effective method to cluster specific cells is adding pseudo hyperedges connecting the cells. If we add pseudo hyperedges connecting the cells with the same supply voltage, the min-cut placer tends to place the cells close to each other. However, the number of  $V_{ddL}$  or  $V_{ddH}$  cells to be clustered could be large. Adding pseudo edges

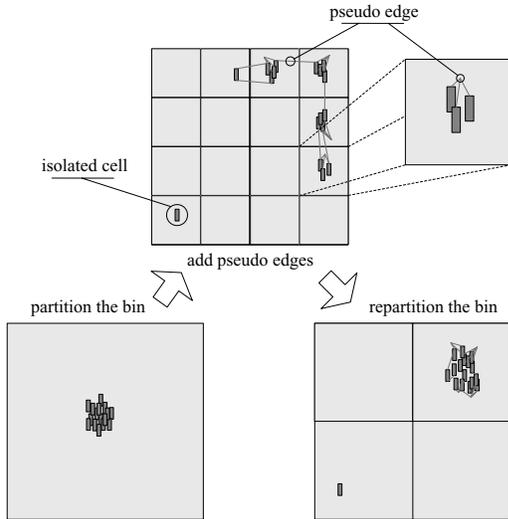


Fig. 1. Proposed clustering strategy.

randomly is likely to cause a large wirelength and delay penalty.

We develop a new soft clustering strategy based on placement feedback. Fig. 1 shows the main idea of our clustering strategy. The initial bin in Fig. 1 is one of the bins at current placement level, which contains both  $V_{ddL}$  and  $V_{ddH}$  cells (only  $V_{ddH}$  cells are drawn). The bin is recursively partitioned with no pseudo hyperedges first. When the subbin is small enough, we add a pseudo hyperedge for each cell, which connects the cell with its neighbors that operate at the same supply voltage. The neighbors of a cell  $v$  include the cells which are placed near  $v$  in the adjacent bin. Then, we merge the subbins that were originally partitioned to be one bin. Finally, the bin is repartitioned with pseudo hyperedges.

The clustering strategy could significantly reduce the negative impact to traditional placement goals, because the clustered cells tend to be placed together in original wirelength-driven placement. On the other hand, isolated cells (the cells that have no adjacent neighbors with the same supply voltage) are not clustered, because dragging the isolated cells closer to each other may lead to a large wirelength penalty. Note that the voltage assignment, together with the soft clustering strategy, is performed at every level except top ones of the partition hierarchy. The isolated cells will have a high priority to be flipped in the layout aware voltage assignment at the next level.

The voltage island generation versus wirelength cost trade-off can also be tuned by changing the weight of pseudo nets. A larger pseudo net weight implies a stronger force of pulling the cells to generate voltage islands.

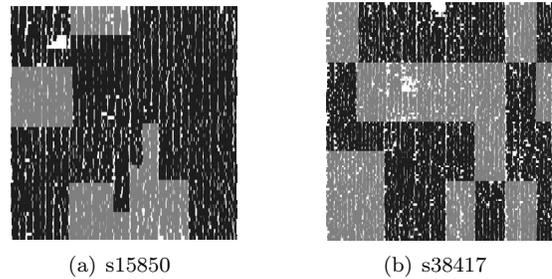


Fig. 2. Placement results of two benchmarks with voltage islands.  $V_{ddH}$  areas are marked with dark blue and  $V_{ddL}$  areas with light gray.

## VII. EXPERIMENTAL RESULTS

We implement the logic and layout aware approach (referred to as CapoLLV) and the approach in [1] (referred to as CapoV) with C++ based on Capo 9.2. In the experiments, we will compare the approaches in power reduction and the number of LCs. We used the same set of benchmarks as in [1] except c6288 as it is a completely balanced circuit and hence it was found equally ineffective in power reduction for all dual VDD approaches.

Table I shows the results of three algorithms: 1) original Capo (version 9.2), which aims at minimizing total half perimeter wirelength; 2) CapoV algorithm in [1]; 3) the proposed algorithm CapoLLV. It can be noticed that our method is as effective as CapoV in reducing total power, and the increase of total wirelength is within 10% (within 5% in most cases) compared to Capo. CapoLLV increases the slack in 8 out of 12 cases, this is probably because the cells on critical paths are assigned  $V_{ddH}$  and clustered, which helps to shorten the longest path. The decreases of slack in the other 4 cases are controlled within 6%. We think it is not fair to compare CapoLLV and CapoV on slack, because CapoV works with a timing-driven strategy in order to promote timing closure at the cost of total wirelength increase.

Table II compares the number of LCs used within a given power bound. The bound is defined as a certain percentage of the original power, which corresponds to the total power when all cells are using  $V_{ddH}$ . We compute the power bounds according to the results of CapoV in order to have a fair comparison. As expected, the results by CapoLLV use much less level converters than CapoV (reduced by 59.50% on average). Since the total power here does not include the power consumed by level converters, CapoLLV actually achieves more power savings.

Fig. 2 shows the placement results of two benchmarks with voltage islands. The boundaries are in a reasonable manner and thus power grid design and verification will not be much complicated.

TABLE I  
EXPERIMENTAL RESULTS OF THREE ALGORITHMS.

name	# of cells	period (ns)	Capo 9.2			CapoV			CapoLLV		
			power	slack	HPWL	power	slack	HPWL	power	slack	HPWL
c880	383	2.4	0.022	-0.20	1.15E6	0.019	-0.134	1.26E6	0.017	-0.211	1.19E6
c1355	546	3.05	0.022	0.06	1.33E6	0.016	0.080	1.31E6	0.015	0.068	1.33E6
c1908	880	4.0	0.024	-0.24	2.08E6	0.018	0.108	2.29E6	0.018	-0.087	2.17E6
c2670	1269	4.0	0.046	-1.02	4.62E6	0.038	-0.398	4.97E6	0.034	-0.768	4.68E6
c3540	1669	5.5	0.039	-0.39	5.29E6	0.021	-0.213	5.25E6	0.022	-0.404	5.33E6
c5315	2307	4.8	0.067	-1.001	7.58E6	0.045	-0.064	8.80E6	0.044	-0.618	7.94E6
c7552	3513	5.0	0.090	-0.21	10.61E6	0.065	-0.046	11.60E6	0.067	-0.082	11.39E6
s1488	661	3.9	0.022	-0.05	1.84E6	0.015	0.125	1.93E6	0.016	-0.025	1.88E6
s15850	10811	8.6	0.123	-3.88	24.94E6	0.097	-1.270	32.49E6	0.094	-3.788	27.25E6
s35932	19843	20.6	0.107	0.44	48.74E6	0.066	0.388	62.11E6	0.055	0.418	50.57E6
s38417	24846	5.6	0.433	-3.30	55.49E6	0.250	-0.942	63.98E6	0.274	-3.384	58.33E6
s38584	21369	10.5	0.259	-1.40	68.15E6	0.174	0.796	72.53E6	0.183	-0.738	70.92E6

period: clock period(ns); slack: worst slack(ns).

TABLE II  
COMPARISON OF CAPOV AND CAPOLLV.

name	power bound	# of LC			HPWL	
		CapoV	CapoLLV	Reduction	CapoV	CapoLLV
c880	85%	60	33	45.00%	1.24E6	1.19E6
c1355	70%	76	38	50.00%	1.33E6	1.33E6
c1908	75%	109	45	58.72%	2.19E6	2.18E6
c2670	80%	113	33	70.80%	4.96E6	4.66E6
c3540	50%	106	67	36.79%	5.48E6	5.27E6
c5315	70%	156	66	57.69%	8.99E6	7.77E6
c7552	70%	205	85	58.54%	11.87E6	11.12E6
s1488	70%	161	48	70.19%	1.89E6	1.90E6
s15850	80%	491	217	55.80%	32.82E6	26.27E6
s35932	60%	1613	393	75.64%	62.98E6	51.17E6
s38417	60%	1667	475	71.51%	64.84E6	57.33E6
s38584	70%	836	307	63.28%	72.81E6	69.49E6

## VIII. SUMMARY AND CONCLUSIONS

In this paper we propose a logic and layout aware approach combined with placement to minimize the number of LCs and overheads in voltage island generation. Several techniques are proposed to consider logical and physical goals simultaneously. In the experimental comparison of this new approach to the approach in [1], our approach has shown great effectiveness in reducing the number of LCs and the implementation of voltage islands with minimal wirelength penalty.

## IX. ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their comments, which improved the quality of the paper. This work is supported by the National Natural Science Foundation of China (NSFC) 60476014.

## REFERENCES

- [1] Bin Liu, Yici Cai, Qiang Zhou and Xianlong Hong, "Power driven placement with layout aware supply voltage assignment for voltage island generation in dual-Vdd designs," in *ASP-DAC'06*, Vol. 17, pp.582-587.
- [2] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," in *Proc. ISLPED95*, pp.3-8.
- [3] K. Usami, M. Igarashi, F. Minami, M. Ishikawa, M. Ichida and K. Nogami, "Automated low-power technique exploiting multiple supply voltages applied to a media processor," *IEEE JSSC*, pp.463-472, Max. 1998.
- [4] C. Chen and M. Sarrafzadeh, "Provably good algorithm for low power consumption with dual supply voltages," in *Proc. ICCAD'99*, pp.76-79.
- [5] S.H. Kulkarni, A.N. Srivastava, and D. Sylvester, "A new algorithm for improved VDD assignment in low power dual VDD systems," in *Proc. ISLPED'04*, pp.200-205.
- [6] A. Srivastava, D. Sylvester and D. Blaauw, "Power minimization using simultaneous gate sizing, dual-Vdd and dual-Vth assignment," in *Proc. DAC'04*, pp.783-787.
- [7] Mongkol Ekpanyapong and Sung Kyu Lim, "Integrated retiming and simultaneous Vdd/Vth scaling for total power minimization," in *Proc. ISPD'06*, pp.142-148.
- [8] C. Yeh, Y. Kang, S. Shieh and J. Wang, "Layout techniques supporting the use of dual supply voltages for cell-based designs," in *Proc. DAC'99*, pp.62-67.
- [9] H. Wu, I.-M. Liu, Martin D. F. Wong, and Y. Wang, "Post-placement voltage island generation under performance requirement," in *Proc. ICCAD'05*, pp.309-316.
- [10] F. Ishihara, F. Sheikh and B. Nikolic, "Level conversion for dual-supply systems," *IEEE Trans. VLSI Syst.*, vol. 12, pp.185-195, Feb. 2004.
- [11] R. Puri, L. Stok, J. Cohn, D. Kung, D. Pan, D. Sylvester, A. Srivastava and S. Kulkarni, "Pushing ASIC performance in a power envelope," in *Proc. DAC'03*, pp.788-793.
- [12] A. E. Caldwell, A. B. Kahng and I. L. Markov, "Can recursive bisection alone produce routable placements?," in *Proc. DAC'00*, pp.477-482.
- [13] A.B. Kahng and S. Reda, "Placement feedback: a concept and method for better min-cut placements," in *Proc. DAC'04*, pp.357-362.