

# Power Driven Placement with Layout Aware Supply Voltage Assignment for Voltage Island Generation in Dual-Vdd Designs

Bin Liu, Yici Cai, Qiang Zhou, Xianlong Hong

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

**Abstract**— In this paper we propose a method for standard cell placement with support for dual supply voltages, aiming to reduce total power under timing constraints and to implement voltage islands with minimal overheads. The method begins with timing and power driven coarse placement, followed by a few iterations between voltage assignment and placement refinement to generate voltage islands. Several techniques, including timing and power driven net weighting, seed growth based voltage assignment, and soft clustering strategy for placement refinements are employed in our implementation. Experimental results on a set of MCNC benchmarks show that our approach is able to produce feasible placement for dual-Vdd designs and significantly reduce total power with a wirelength increase within 14% compared to a power and timing driven placer without voltage islands.

## I. INTRODUCTION

Due to rapidly increasing on-chip power density with technology evolution and the growing market for battery powered devices, power dissipation has become one of the most critical concerns in modern chip designs. Researchers have proposed many low power design styles in recent years, among which multiple supply voltage (MSV) is a promising scheme to achieve significant reduction on both dynamic and static power while maintaining performance [1–4]. It is reported in [4] that dynamic power can be reduced by about 30% on average if an additional supply voltage is available. MSV can be combined with other low power techniques [5] and has been successfully implemented in some commercial chips [6, 7]. The design of MSV circuits can work at either macro module level during design planning [8] or cell level after logic synthesis [1, 2, 7]. In this paper, we focus on cell based designs with two supply voltages.

The basic idea behind MSV is to trade timing slacks for power reduction by using high voltage on cells with negative (or little) slack to maintain performance and using low voltage on others to save power. Previous works have demonstrated that the number of gates on critical paths accounts for only a small portion of total gates, while the majority of other gates have relatively large slacks, leaving much room for potential power reduction using MSV [2].

Despite its effectiveness and flexibility, MSV introduces some particular electrical and physical constraints. Level converters are needed whenever the supply voltage of a driver cell is lower than that of the receiver [9]. Moreover, cells under different voltages should be carefully placed so as to facilitate power network design and to reduce chip complexity. To save power while minimizing overheads induced by these effects, efforts must be made in two major operations in MSV design, voltage assignment and placement.

Most published algorithms for voltage assignment begin by setting all cells to VddH, and try to lower the supply voltages for some cells based on static timing analysis. Clustered voltage scaling (CVS) tries to reduce the supply voltages of cells from primary outputs to primary inputs in reverse topological order and does not allow voltage converters in the middle of a path [1]. A more flexible approach allowing level converters is called enhanced clustered voltage scaling (ECVS) [2], which proves to provide appreciably larger power reduction compared with CVS [4]. Other techniques for timing and power optimization, including gate sizing, multiple threshold voltages, can be combined with MSV for better timing-power tradeoff [5].

Cell placement is a critical step for MSV designs in that it greatly influences power grid complexity, as well as path delay, which can challenge timing closure especially when most slacks are traded for power saving after voltage assignment. There are two kinds of placement schemes for MSV designs: one is *row based* [10], where there are interleaving rows or half-rows for VddH cells and VddL cells; the other is *region based*, where the cells in each region (called voltage island) operate under the same supply voltage. Recently voltage island approach has been widely recognized as the state of the art in MSV design for its structural flexibility [3, 8, 11]. To generate voltage islands, cells with the same supply voltage must be physically clustered during placement, which is a new requirement for the placer. Although there are industrial efforts on tool development supporting MSV layout [7, 12, 13], detailed algorithms focusing on the physical implementation of fine-grained voltage islands are not seen in open literature.

As we examine the methodology for voltage island design, it is interesting to notice that voltage assignment is

often performed prior to layout (e.g., during design planning or logic synthesis). Known voltage assignment algorithms tend to work without specific physical information [2, 4, 5], which may result in at least the following problems.

1. Interconnect delay, which dominates total path delay in deep submicron technology, can hardly be estimated accurately before placement. Thus, there can be either too many VddL cells, causing trouble to timing closure, or too few VddL cells, wasting slack that can be otherwise traded for power saving.
2. With final locations of cells unknown, the predefined assignment of voltages is likely to cause large wire-length and delay penalty even if flexible clustering strategies are employed in the placer.

The penalties caused by pre-placement voltage assignment give rise to the idea of layout aware voltage assignment. A concept of voltage assignment exploiting both logical and physical adjacency is mentioned in [7], but no detailed algorithm is described. Another work attempting to combine voltage assignment with placement is now in progress [14].

The purpose of this work is to develop a practical method for fine-grained voltage island generation in dual-Vdd designs. Specifically, we focus on two major aspects in design and implementation of chips with voltage islands: reducing total power under timing constraints, and reducing electrical and physical overheads. We propose a practical design flow for voltage island generation (outlined in Fig. 1). Our method begins with timing and power driven coarse placement, followed by a few iterations between layout aware voltage assignment and placement refinement. This flow is general enough to embrace many practical techniques and considerations. Preliminary algorithms are developed to support the flow. Experimental results on a set of MCNC benchmarks have demonstrated the effectiveness of our approach.

## II. BACKGROUND

### A. Timing and Power Analysis

In order to support flexible voltage assignment in dual-Vdd designs, at least some cells in the library should be designed to work with both supply voltages, possibly with different implementations. Gate level dual-Vdd design style allows replacing VddH cells with VddL cells along uncritical paths. Thus, exhaustive static timing analysis is a prerequisite to voltage assignment [1, 2, 4, 5].

In static timing analysis, the combinatorial part of a circuit is modelled as a weighted directed acyclic graph (DAG) called timing graph, where every node represents a signal pin in the netlist, and the weight of every directed edge represents either gate delay or wire delay between

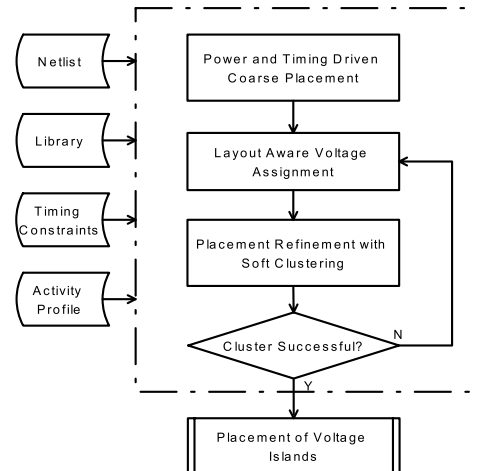


Fig. 1. Flow for voltage island generation.

two pins. Usually delay constraints are imposed by specifying the maximum delay between primary inputs and primary outputs. After a graph traversal in topological order (forward pass) and another in reverse topological order (backward pass), the arrival time, required time and slack at every node can be calculated. This process takes  $O(|N| + |E|)$  time, where  $N$  is the set of nodes and  $E$  is the set of edges.

In this work, we use table-based models for gate delay, as well as leakage and short-circuit power for every cell. We use a Elmore-star model to computer wire delays [15]. Switching power is calculated with the Eqn. 1.

$$P_{sw} = \alpha f V_{dd}^2 C_{load} \quad (1)$$

Where  $\alpha$  is the activity factor, indicating the probability that the signal changes in one clock cycle;  $f$  is clock frequency;  $C_{load}$  is load capacitance of the gate, including gate capacitance and wire capacitance. Given the quadratic dependency of switching power on supply voltage, the efficacy of lowering supply voltage in power reduction is beyond controversy. Eqn. 1 also indicates other ways to save power by reducing the length of high activity wires, or using VddL cells to drive these nets, both of which are considered in our approach.

### B. Cut-based Placement Paradigm

In this subsection, we review the framework of cut-based placement algorithms, and focus on the outline of Capo [16], which forms the base for our placement algorithm. A min-cut placement instance contains: 1) a rectangular region (referred to as *bin*) where cells are to be placed; 2) a hypergraph, with each node representing a cell and each hyperedge representing a signal net connecting two or more cells. A min-cut placer recursively partitions each bin and its associated hypergraph at current level, and assigns the subhypergraphs to subbins,

minimizing total (weighted) net cuts for total (weighted) wirelength reduction. The cut direction usually alternates between horizontal and vertical cuts. For wirelength estimation during the placement process, the nodes in each bin can be considered as being placed at the geometric center of the bin.

Capo is an elegant cut-based placer, with several techniques to improve total wirelength including placement feedback, weighted terminal propagation, etc. The idea of feedback in placement [17], i.e., merging adjacent bins and perform repartition under the guidance of information from last iteration is an inspiration of our idea on placement refinement.

### III. VOLTAGE ISLAND GENERATION METHODOLOGY

As illustrated in Fig. 1, our voltage island design methodology does not require pre-layout voltage assignment. Instead, layout aware voltage assignment is performed after a coarse placement result is available, so that delay due to interconnect effects can be more accurately captured in timing analysis, and physical adjacency information is available to guide the assignment of supply voltages.

It is natural to cluster cells with the same supply voltage in a region to form a voltage island. However, aggressively clustering cells is likely to result in wirelength increase and timing violation, especially when the region is relatively large and the number of VddH cells and VddL cells are close. We hereby further exploit the flexibility of voltage island generation by iteratively updating the voltage assignment and the layout. The placement of cells can be adjusted by merging adjacent bins and repartitioning the netlist, aiming to physically cluster cells with the same voltage. In our implementation, we do not require that the island be generated after adjusting placement. Instead, repartition is done after with some additional hyperedges on cells to be clustered (this is called *soft clustering*). After repartition, the assignment of voltages is also adjusted to fit changes in layout. Convergence of the iterations can be guaranteed with increasing weights of additional hyperedges and a decreasing threshold parameter used in incremental voltage assignment.

### IV. COARSE PLACEMENT WITH TIMING AND POWER DRIVEN NET WEIGHTING

Coarse placement is like the first several passes of a timing and power driven global placement with all cells operating under VddH. While the final objective of this work is to minimize power, it is helpful to optimize timing, instead of merely meeting timing constraints or optimizing path delay on a few critical paths during coarse placement, because additional slacks can provide more opportunities to use VddL cells. It is particularly advantageous that even the most critical path has some slack, which

makes the exploitation of physical adjacency possible for cells along critical paths.

Net weighting is a popular technique for large scale placement due to its flexibility and efficiency. There have been extensive works on net weighting strategies for timing optimization in placement [18, 19]; a few other works employ switching activity based net weighting to minimize total switching power [20]. Different from strategies that focus either timing or power, we seek a weighting scheme that improves both. The empirical formula in Eqn. 2 is used for net weighting taking account of both slack and switching power.

$$W = \begin{cases} (1 + c \times \alpha) \times (1 + \frac{T_0}{T_0/N + slack}), & slack \geq 0 \\ (1 + c \times \alpha) \times (1 + N), & else \end{cases} \quad (2)$$

Here  $\alpha$  is the switching probability;  $slack$  is the minimum slack at the input of downstream cells;  $T_0$  is typically several times larger than the gate delay;  $N$  and  $c$  are constant parameters. Other net weighting methods can also be used in coarse placement. It should be emphasized that coarse placement not only try to reduce delay along a few most critical paths, but also attempt to create more slacks along even moderately critical paths, because slacks can probably be traded for power saving afterward.

Net weights are incrementally updated in the feedback procedure at every partition level. The weight of an edge is updated considering both its previous value and new value.

$$W = \beta W_{new} + (1 - \beta) W_{orig} \quad (3)$$

Our experience shows that  $\beta$  should be kept below 0.3 to maintain consistency and avoid oscillation of path delays.

### V. INITIAL VOLTAGE ASSIGNMENT

Initial voltage assignment largely defines the final pattern of voltage islands. The proposed algorithm for initial assignment works in a seed-growth manner, exploiting both physical adjacency and logical adjacency [7]. The algorithm flow is described in Algorithm 1.

We associate a tendency value to every VddH cell. Similar to [4], the tendency is defined as follows.

$$tendency = \begin{cases} \frac{(G_{self} + \gamma G_{LC}) \times slack}{\Delta delay}, & slack > \Delta delay \\ 0, & else \end{cases} \quad (4)$$

Here  $G_{self}$  is the power reduction due to the use of VddL;  $G_{LC}$  is the reduction of needed level converters;  $slack$  is the minimum slack at the output pins;  $\Delta delay$  is the increase of gate delay (measured with the maximum increase on pin-to-pin delay).

Note that the tendency of a cell depends on the status of its logical neighbors as well as its slack, both of which can change dynamically in the assignment process. Thus, incremental updates on the timing graph and cell tendencies are required. For efficiency consideration, these

updates are performed lazily, and only the tendencies of relevant unprocessed cells are updated.

The seed growth process begins by selecting *seed bins*. We calculate a priority value for every bin by investigating the tendencies of cells in it. Bins with priority larger than a threshold are selected as seed bins. Priority of a bin is defined as the average tendency of cells in the bin, as shown in Eqn. 5, where  $N$  is the number of cells in the bin.

$$priority = \frac{1}{N} \sum_{i=1}^N tendency(cell_i) \quad (5)$$

After seed bins are selected, the algorithm enters a procedure of selecting cells to work under VddL. This is done in two phases. The first phase tries to assign VddL to every cell in seed bins if timing constraints are not violated, while the second phase tries to assign VddL to more cells across the chip. The purpose of the first phase is to generate some physical clusters of VddL cells with no wirelength penalty, which form the bases of VddL islands. The second phase can be viewed as a logical expansion of VddL cells according to on existing VddL cells, aiming to reduce power dissipation without adding level converters.

---

#### Algorithm 1 Initial Voltage Assignment

---

**Require:** *threshold*

```

sort bins and cells according to priority and tendency;
while there are unprocessed bins do
  currentBin ← the unprocessed bin with highest priority;
  if currentBin.priority < threshold then
    break;
  else
    try to use VddL for every cell in currentBin without timing violation;
  end if
  update priorities of unprocessed bins;
end while
while there are unprocessed cells do
  c ← the unprocessed cell with highest tendency;
  try to use VddL for c if timing budgets are met;
end while

```

---

## VI. ITERATIVE VOLTAGE ASSIGNMENT AND PLACEMENT REFINEMENT

### A. Placement Refinement

Placement refinement is a procedure that locally adjusts the locations of cells, so that cells with the same supply voltage get closer to each other. The refinement procedure is much the same as the feedback mechanism used in Capo [17]. After voltage assignment, it is probably that most bins contain both VddH cells and VddL cells. In placement refinement stage, neighboring bins are merged to generate a new larger bin, containing all

cells in original bins. The new bin is repartitioned, taking into account both total wirelength and requirement for clustering. Instead of modifying the hypergraph partitioning algorithm, a simple method is adopted to incorporate clustering consideration into the partitioning problem. Pseudo hyperedges connecting cells with the same supply voltage are added to the hypergraph to be partitioned. Since min-cut placement minimizes total weighted wirelength naturally, cells connected with the pseudo hyperedges tend to get close to one another and clustering can be realized. This approach is referred to as *soft clustering*, because it differs from a strong clustering method (hard clustering) that combines all cells to be clustered into a soft macro-module and performs mixed-size placement afterward. Empirically, although hard clustering can be easily implemented with existing tools, it tends to increase total wirelength significantly; similar conclusion has been validated in the research of integrated floorplanning and placement [21]. The weights of pseudo edges reflect the desire of clustering. In order to reduce wirelength overheads, the total weights of the added edges should be kept small, at least in the first a few iterations, when complete isolation of VddH cells and VddL cells is not required. In order to accelerate convergence, these weights are increased iteration by iteration.

### B. Incremental Update of Voltage Assignment

In the proposed flow, a voltage island is generated if a bin contains purely VddH cells or VddL cells. However, there can be many bins containing both kinds of cells even after placement refinement due to the inadequacy of soft clustering in some regions. Intuitively, if most part of a region is filled with VddH cells, it is desired that the supply voltages for all VddL cells in the region are raised to VddH. If only a small portion of cells are powered by VddH in a region, it is probable that the supply voltages of VddH cells in this region can be lowered after replacing some VddL cells with VddH cells in some other bins. By increasing and decreasing the supply voltages region by region, timing slacks can be concentrated into some regions to form VddL islands, and other regions that can not work at VddL without timing violations eventually become VddH islands. This is done incrementally with the iterations.

The algorithm is illustrated in Algorithm 2. A queue containing all bins with both VddH cells and VddL cells is constructed, and the portion of VddH cells in every bin is monitored. At each iteration, the algorithm tries to boost existing dominance of VddH or VddL cells in some bins. For every bin with with the portion of VddH cells larger than a threshold, all other VddL cells in the bin are replaced with VddH cells, and the bin is removed from the queue. Then every other bin in the queue is examined in reverse order of VddH portion to see if all cells in it can be powered by VddL while meeting the constraints.



TABLE I  
EXPERIMENTAL RESULTS OF THREE ALGORITHMS.

name	period (ns)	Capo 9.2				Capo 9.2+net weighting				Proposed algorithm			
		power	slack	time	HPWL	power	slack	time	HPWL	power	slack	time	HPWL
c880	2.4	0.022	-0.2	2.06	1.15E6	0.022	-0.12	5.74	1.23E6	0.019	-0.134	7.46	1.26E6
c1355	3.05	0.022	0.06	3.95	1.33E6	0.021	0.19	7.36	1.32E6	0.016	0.080	11.38	1.31E6
c1908	4.0	0.024	-0.24	8.71	2.08E6	0.025	0.02	14.78	2.26E6	0.018	0.108	22.30	2.29E6
c2670	4.0	0.046	-1.02	14.14	4.62E6	0.049	-0.64	33.89	5.02E6	0.038	-0.398	57.56	4.97E6
c3540	5.5	0.039	-0.39	15.12	5.29E6	0.040	0.19	26.13	5.41E6	0.021	-0.213	41.62	5.25E6
c5315	4.8	0.067	-1.001	28.70	7.58E6	0.072	0.25	53.54	8.59E6	0.045	-0.064	367.22	8.80E6
c6288	9.5	0.028	-2.31	23.76	5.21E6	0.029	-2.16	34.47	5.39E6	0.027	-1.965	86.25	5.30E6
c7552	5.0	0.090	-0.21	46.31	10.61E6	0.097	-0.07	91.95	11.21E6	0.065	-0.046	165.93	11.60E6
s1488	3.9	0.022	-0.05	5.51	1.84E6	0.022	0.15	7.46	1.84E6	0.015	0.125	14.83	1.93E6
s15850	8.6	0.123	-3.88	167.25	24.94E6	0.143	-1.78	325.18	32.03E6	0.097	-1.270	591.04	32.49E6
s35932	20.6	0.107	0.44	291.23	48.74E6	0.114	0.613	514.27	54.98E6	0.066	0.388	824.02	62.11E6
s38417	5.6	0.433	-3.30	402.51	55.49E6	0.471	-1.46	722.20	64.36E6	0.250	-0.942	1041.88	63.98E6
s38584	10.5	0.259	-1.40	376.92	68.15E6	0.267	0.743	666.47	71.23E6	0.174	0.796	975.89	72.53E6

period: clock period(ns); slack: worst slack(ns); time:running time(s).

---

### Algorithm 2 Incremental Voltage Reassignment

---

**Require:** *threshold*

construct a queue  $Q$ , containing all bins with both VddH cells and VddL cells, order by the portion of VddH cells;

**while**  $Q$  is nonempty **do**

$currentBin \leftarrow$  the bin with highest portion of VddH cells;

**if** the portion of VddH cells is lower than *threshold* **then**

**return;**

**end if**

assign VddH to all cells in  $currentBin$ ;

update timing graph;

remove  $currentBin$  from  $Q$ ;

**for all**  $bin$  in  $Q$  in reverse order **do**

try to use VddL for  $currentBin$ ;

**end for**

**end while**

---

The threshold parameter is initially close to 1, and decreases toward 0 with iterations between placement refinement and incremental voltage assignment, which guarantees the convergence of the voltage island generation algorithm.

## VII. EXPERIMENTAL RESULTS

The proposed algorithms have been implemented with C++ based on Capo 9.2. We create our dual-Vdd library based on a industrial 0.18um library with VddH=1.8V. We add a VddL(1.2V) alternative for every cell in the original library and compute its delay based on alpha-power law model.

Experiments are performed on a set of ISCAS85 and IS-CAS89 benchmarks with specified timing constraints and activity profiles. In order to evaluate the effectiveness of the proposed method, we examine the results of three

algorithms: 1) original Capo(version 9.2), which aims at minimizing total half perimeter wirelength; 2) Capo with the net weighting strategy described in Section IV (referred to as CapoW), aiming at improving delay and power by minimizing the weighted total wirelength; 3) the proposed algorithm with support for dual-Vdd designs (referred to as CapoV). We measure total power, maximum slack, total half perimeter wirelength and running time for all the three algorithms (Table I).

When comparing CapoW with Capo, it can be noticed that timing is consistently improved in all the benchmarks, and the increase of total wirelength is within 12% except C5315, C6288, C15850 and C38417, which all have tight timing constraints. While it is evident that our net weighting strategy is effective in promoting timing closure, the total power is not remarkably reduced; actually it is increased in many designs (mostly by within 10%). This is probably because the timing constraints in many benchmarks are rather tight and the values of net weighting parameters in our implementation are tuned mainly for timing optimization. A lot of slacks are created on both critical paths and some uncritical paths, thus enlarging the room for power optimization in voltage assignment.

As expected, the results by CapoV show significant power saving with the use of dual-supply voltage, which corroborates previous works. What we are especially concerned is the implementation penalty on wirelength. Since CapoV and CapoW are based on the same coarse placement algorithm with the same net weighting parameters, and CapoV works with some additional clustering constraints while no further constraints are imposed on CapoW, the total wirelength produced by CapoW can be viewed as an ‘‘upper bound’’ for that by CapoV. (Our experiments show that for some benchmarks CapoV produces slightly shorter wirelength than CapoW, which is probably because the additional merge-and-repartition iterations are helpful to optimize wirelength (as well as tim-

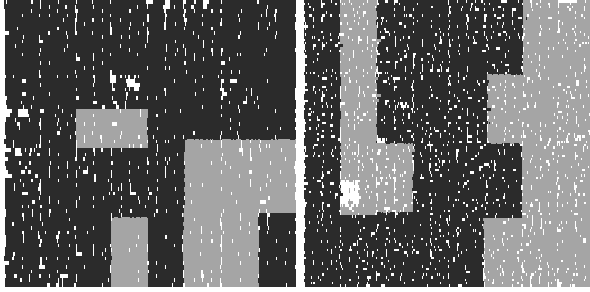


Fig. 2. Typical placement results with voltage islands. Left is s15850, and right is s38417. VddH areas are marked with dark purple and VddL areas with light yellow.

ing) at the cost of more running time.) Table I shows that CapoV produces results with wirelength increase of -2.96% to 13.7% compared with CapoW. We think it is not fair to compare CapoV and original Capo on wirelength, because CapoV works with net weights due to timing and power considerations, which inherently is inconsistent with wirelength optimization.

Fig. 2 illustrates the placement results of two benchmarks with voltage islands. Power grid design and verification for these designs will not be much complicated because the number of voltage islands are not large.

## VIII. CONCLUSION

In this paper we present an effective methodology together with algorithms to reduce power dissipation under timing constraints in placement and to provide physical-level support for voltage island designs. The proposed layout aware voltage assignment and iterative adjustment on placement and voltage assignment have shown great effectiveness in the implementation of voltage islands with minimal wirelength penalty. Our results probably indicate that it is necessary to perform voltage assignment, or at least adjustment on voltage assignment, during placement in order to reduce physical overhead. In the current implementation, level converters are not dynamically inserted and deleted during voltage assignment. ECO placement should be performed afterward taking level converters into account. Future works can make efforts to accelerate design convergence and better consider level converter issues.

## ACKNOWLEDGEMENT

The authors would like to thank Prof. Igor Markov, Prof. Patrick Madden and Prof. David Pan for their kind help. This work is supported by the Hi-Tech Research & Development (863) Program of China (No. 2005AA1Z1230) and the National Natural Science Foundation of China (NSFC) (No. 60476014).

## REFERENCES

- [1] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," in *Proc. ISLPED'95*, pp. 3-8.
- [2] C. Chen, A. Srivastava and M. Sarrafzadeh, "On gate level power optimization using dual-supply voltages," *IEEE Trans. on VLSI Syst.*, vol. 9, pp. 616-629, Oct. 2001.
- [3] 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," in *ICCAD'02*, pp.195-202, Nov. 2002.
- [4] 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.
- [5] 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.
- [6] S.K. Mathew, M.A. Anders, B. Bloechel, T. Nguyen, R.K. Krishnamurthy and S. Borkar, "A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 40, pp.44-51, Jan. 2005.
- [7] 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.
- [8] J. Hu, Y. Shin, N. Dhanwada and Radu Marculescu, "Architecting voltage islands in core-based system-on-a-chip designs," in *Proc. ISLPED'04*, pp.180-185.
- [9] F. Ishihara, F. Sheikh and B. Nikolic, "Level conversion for dual-supply systems," *IEEE Trans. VLSI Syst.*, vol. 12, pp.185-195, Feb. 2004.
- [10] 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.
- [11] R. Puri, L. Stok, S. Bhattacharya, "Keeping hot chips cool," in *Proc. DAC'05*, pp.285-288.
- [12] Synopsys inc., Galaxy design platform multi-voltage Design, available online, [http://www.synopsys.com/products/power/multivoltage\\_bkgrd.pdf](http://www.synopsys.com/products/power/multivoltage_bkgrd.pdf).
- [13] Cadence inc., Cadence/TSMC Reference Flow 6.0.
- [14] P.H. Madden, private communication.
- [15] A.B. Kahng, S. Mantik and I.L. Markov, "Min-max placement for large-scale timing optimization," in *Proc. ISPD'02*, pp.143-148.
- [16] A.E. Caldwell, A.B. Kahng, and I.L. Markov, "Can recursive bisection alone produce routable placements?," in *Proc. DAC'00*, pp.477-482.
- [17] A.B. Kahng and S. Reda, "Placement feedback: a concept and method for better min-cut placements," in *Proc. DAC'04*, pp.357-362.
- [18] T. Kong, "A novel net weighting algorithm for timing-driven placement," in *Proc. ICCAD'02*, pp.172-176.
- [19] A.B. Kahng and Q. Wang, "Implementation and extensibility of an analytic placer," *IEEE Trans. Computer-Aided Design*, vol. 24, pp.734-747, May 2005.
- [20] Y. Cheon, P.H. Ho, A.B. Kahng, S. Reda and Q. Wang, "Power aware placement," in *Proc. DAC'05*, pp.795-800.
- [21] J.A. Roy, S.N. Adya, D.A. Papa and I.L. Markov, "Min-cut floorplacement," *IEEE Trans. on Computer-Aided Design*, to appear.