

Voltage Island Aware Floorplanning for Power and Timing Optimization *

Wan-Ping Lee[†], Hung-Yi Liu[†], and Yao-Wen Chang^{†‡}

Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan[†]
Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan[‡]
{planet, daniel}@eda.ee.ntu.edu.tw; ywchang@cc.ee.ntu.edu.tw

ABSTRACT

Power consumption is a crucial concern in nanometer chip design. Researchers have shown that multiple supply voltage (MSV) is an effective method for power consumption reduction. The underlying idea behind MSV is the trade-off between power saving and performance. In this paper, we present an effective voltage assignment technique based on dynamic programming. Given a netlist without reconvergent fanouts, the dynamic programming can guarantee an optimal solution for the voltage assignment. We then generate a level shifter for each net that connects two blocks in different voltage domains, and perform power-network aware floorplanning for the MSV design. Experimental results show that our floorplanner is very effective in optimizing power consumption under timing constraints.

1. INTRODUCTION

As the CMOS technology enters the nanometer era, power dissipation is a key challenge in nanometer chip design. Power consumption generally breaks down into two sources, dynamic power and static power. While static power in modern technology mainly comes from leakage current, dynamic power P_{switch} is incurred from a device's switching activities. It can be computed by

$$P_{switch} = k \cdot C_{load} \cdot V_{dd}^2 \cdot f,$$

where k is the switching rate, C_{load} is load capacitance, V_{dd} is the supply voltage, and f is the clock frequency. Compared with static power, dynamic power often dominates the total power consumption in high frequency circuit design.

In a VLSI design, power consumption and performance optimizations often conflict with each other. How to minimize power consumption and simultaneously satisfy the performance constraint is a challenging problem. Researchers have proposed many low supply voltage approaches, among which *multiple supply voltage (MSV)* [11] is a popular technique for power consumption reduction. The underlying

idea behind MSV is the trade-off between the power saving and performance. Under the performance constraints, it is desired to assign cells along non-critical paths with lower power supply voltages for power saving. Thus the timing slack available on non-critical paths can be effectively converted to power saving.

There are two major categories of existing algorithms for the VDD assignment, Clustered Voltage Scaling (CVS) [11] and Extended Clustered Voltage Scaling (ECVS) [12]. Both algorithms assign appropriate supply voltages to gates by traversing a combinational circuit from the primary outputs to the primary inputs in leveled order. CVS does not allow low-VDD (VDDL) gates to drive high-VDD (VDDH) gates. Relaxing this restriction, ECVS uses *level shifters* for VDDL gates to drive VDDH ones. As a result, ECVS can provide appreciably larger power reduction compared with CVS. For example, Kulkarni et al. [9] recently presented a heuristic based on ECVS for power saving. In addition to CVS and ECVS, Chang and Pedram [4, 5] applied dynamic programming for voltage assignment. In physical design, Wu et al. [13] minimized the number of voltage islands after placement. (Each voltage island is composed of cells/blocks with the same supply voltage.) They focused on the minimization of the number of voltage islands and did not consider the constraint imposed by the architecture of the power/ground (P/G) network. For practical applications, we shall consider the voltage island constraints and the P/G network architecture for simultaneous timing and power optimization.

In this paper, we propose a reference flow that includes three phases from voltage island partitioning, level-shifter generation, to power-network aware floorplanning. In Phase I, we handle voltage island partitioning by dynamic programming (DP). Given a netlist without reconvergent fanouts, the DP can guarantee an optimal solution for the voltage assignment in linear time. Since level shifters are needed when a VDDL block drives a VDDH block, level shifters are introduced and treated as soft blocks during floorplanning in Phase II. In Phase III, we conduct power-network aware floorplanning for the original hard blocks and the additional level-shifter (soft) blocks together to make the critical paths satisfy the timing constraint. Experimental results show that our power-network aware floorplanner is very effective in optimizing power consumption under timing constraints. Satisfying the timing constraint, for example, it reduces the power-network resource by 16% on average with a reasonable overhead of 4% in area.

The remainder of this paper is organized as follows. Section 2 gives the formulation of voltage-island partitioning and power-network aware floorplanning. The reference flow for solving this problem is proposed in Section 3. Experi-

*This work was supported in part by TSMC Inc. and NSC of Taiwan under Grant No's. NSC 94-2215-E-002-030, NSC 94-2220-E-002-001, and NSC 94-2752-E-002-008-PAE.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD'06 November 5-9, 2006, San Jose, CA
Copyright 2006 ACM 1-59593-389-1/06/0011 ...\$5.00.

mental results are reported in Section 4. Finally, we give conclusions in Section 5.

2. PROBLEM FORMULATION

We formulate a netlist as a directed acyclic graph (DAG). A vertex represents a primary input, a primary output, or a block, while an edge denotes an interconnect net.

Given k choices of supply voltages, $VDDj$, $1 \leq j \leq k$, an n -vertex DAG, $G = (V, E)$, and delay d_i for each vertex $v_i \in V$, $d_i \in \{d_i^1, d_i^2, \dots, d_i^k\}$, where d_i^j denotes the delay of a vertex v_i operated at the j -th voltage domain $VDDj$, according to static timing analysis (STA), the arrival time a_i and the required time r_i of v_i are derived as follows:

$$a_i = \begin{cases} \max_{v_j \in FI_i} a_j, & FI_i \neq \phi \\ 0, & FI_i = \phi \end{cases}, \quad (1)$$

and

$$r_i = \begin{cases} \min_{v_j \in FO_i} a_j - d_i, & FO_i \neq \phi \\ T_{cycle}, & FO_i = \phi \end{cases}, \quad (2)$$

where FI_i and FO_i are sets of the fanin and fanout vertices of v_i respectively, and T_{cycle} is the clock cycle time of the netlist. Using the STA model, we define the static-timing constraint as follows.

DEFINITION 1. (Static-Timing Constraint) *Given a clock cycle time and a DAG, $G = (V, E)$, corresponding to a netlist, the static-timing constraint of the netlist is $a_i \leq r_i, \forall v_i \in V$, where a_i and r_i are given in Equations (1) and (2).*

For nanometer VLSI design, the interconnect delay dominates the circuit performance. However, STA cannot model the interconnect delay without physical information. In the floorplanning stage, since block positions are determined (and so is wirelength), we can further estimate timing more accurately. For efficient estimation, we base on the STA result and transform the slack of each block b into wirelength [6]. The length upper-bound o_i of the net, whose source is b_i , is derived from the following linear normalization:

$$o_i = \zeta \cdot s_i = \zeta \cdot (r_i - a_i), \quad (3)$$

where s_i is the slack of block i and ζ is a constant to scale timing to wirelength.

DEFINITION 2. (Floorplan-Timing Constraint) *A floorplan satisfies floorplan-timing constraint if and only if for each interconnect whose source is block b_i , the interconnect length is less than or equal to o_i in Equation (3).*

Another important cost metric in an MSV design is power network resource cost. As shown in Figure 1, the floorplan in Figure 1(a) needs more power/ground lines than that in Figure 1(b). It should be noted that, in practical designs, a power/ground mesh is synthesized in uniform pitch. Therefore, even lower-power blocks inside a higher-power ring would be masked by higher-power lines, and vice versa. This is the reason why the second and third (from left) vertical power lines in the right side of Figure 1(a) are still needed. Accordingly, we propose the cost metric *power-network resource requirement* as follows.

DEFINITION 3. (Power-Network Resource Requirement) *Given a floorplan of a set of blocks $B = B_1 \cup B_1 \cup \dots \cup B_k$, $B_i \cap B_j = \phi, i \neq j$, where B_i is the set of blocks operated at voltage $VDDi$, the power-network resource requirement of the floorplan equals $\sum_{i=1}^k u_i$, where u_i is the half perimeter wirelength of the bounding box of B_i .*

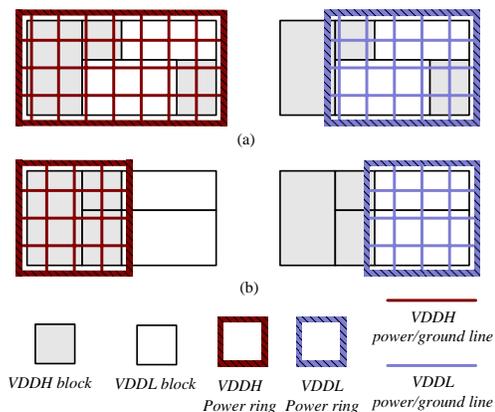


Figure 1: An example dual-voltage floorplan with uniform-structured power mesh. The power-network resource requirement of (b) is smaller (requires fewer power/ground lines), and thus (b) is a better floorplan.

According to Definition 3, the power-network resource requirement of the floorplan in Figure 1(a) is greater than that in Figure 1(b) since both bounding boxes of VDDH and VDDL blocks in the floorplan of Figure 1(a) are larger than those of Figure 1(b). Consequently, the floorplan in Figure 1(b) is more desirable.

However, a floorplan satisfying the static- and floorplan-timing constraints, consuming low power, and requiring modest power-network resource, may have an undesirable shape, e.g., all blocks are in a row. Therefore, we need a fixed-outline constraint to limit the shape of the floorplan. Further, fixed-outline floorplanning is more popular for modern VLSI design [2, 7].

DEFINITION 4. (Fixed-Outline Constraint) *Given a fixed outline (W^*, H^*) of a desired rectangle bounding box, where W^* (H^*) is the width (height) of the box, any block of a floorplan must be placed inside the bounding box.*

Based on the above definitions, the problem addressed in this paper is formulated as follows.

DEFINITION 5. (Multi-Voltage Floorplanning [MVF] Problem) *Given multiple supply-voltage choices, a set of blocks, a netlist, a static-timing and a fixed-outline constraints, assign each block with a supply voltage and its coordinate in a floorplan so that the power consumption and the power-network resource requirement are minimized and both the static-timing and fixed-outline constraints are satisfied.*

3. ALGORITHM

3.1 Overview

Figure 2 shows our flow for solving the MVF problem. The flow consists of three phases: (I) voltage assignment, (II) level-shifter (block) insertion, and (III) power-network aware floorplanning. For Phase I, we present a dynamic-programming (DP) based method to solve the voltage assignment problem. As supply voltages are assigned to the circuit blocks in Phase I, we check in Phase II whether a net needs a level shifter and insert one as a *soft* block if needed. Finally in Phase III, we transform the precomputed slack into the wirelength constraint and perform floorplanning on all blocks, circuit blocks and level shifters (soft blocks), to minimize the power-network resource requirement. The

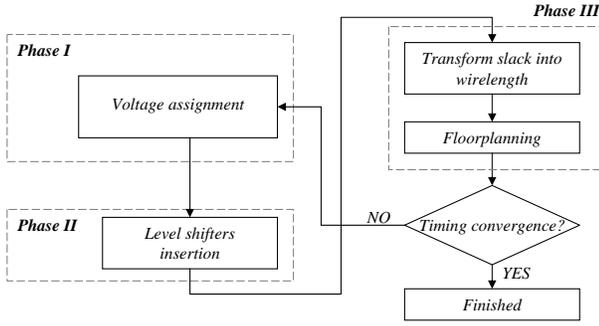


Figure 2: Algorithm flow for the MVF problem.

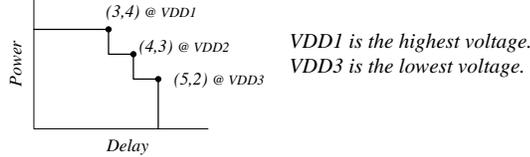


Figure 3: An example DP-curve. The three points of the DP-curve represent the delay-power characteristics of different supply voltages.

floorplanning is based on simulated annealing (SA) [8] using the B*-tree floorplan representation [1, 2].

After the floorplanning, we check if the timing converges. If not, we feed back the current physical information to Phase I and make the timing constraint (T_{cycle}) more stringent to reserve more timing slack for floorplanning. Note that the iteration will eventually terminate; in the worst case, all blocks are assigned the highest supply voltage, and thus the resulting timing must satisfy the timing constraint (unless the given timing constraint is over constrained, for which no feasible solution is possible).

3.2 Dynamic Programming for Voltage Assignment

In this section, we propose a dynamic-programming method to assign a supply voltage for each block. We represent the delay-power characteristics of a block as a *DP-curve* (Delay-Power-curve). For each block b , a DP-curve of b is a power-consumption function of the circuit delay.

PROPERTY 1. *Given a set of candidate supply voltages for a block, the DP-curve of the block is a discrete monotonic-decreasing power-consumption function of delay.*

The property is followed by the natural characteristic of the tradeoff between power saving and performance. To have a smaller delay, a block has to consume more power, and vice versa. See Figure 3 for an example DP-curve.

Given a netlist, we integrate the DP-curves from primary inputs (PIs) to primary outputs (POs) by using dynamic programming. This problem is very similar to delay constrained technology mapping [3]. The difference is that we must consider the level shifters' effects. Section 3.2.1 presents an efficient method for generating the points like those used in delay constrained technology mapping; the algorithm for solving MSV is elaborated in Sections 3.2.2 to 3.2.4.

3.2.1 Lower-bound Merge Operation

Our algorithms extends the *lower-bound merge operation* proposed by Chaudhary and Pedram [3] for area and delay technology mapping. Initially, the DP-curve of each block is set according to its original delay-power characteristics, as

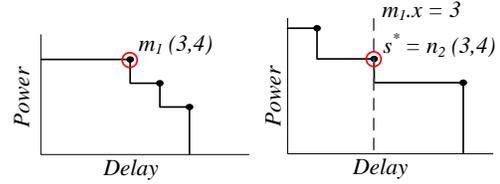


Figure 4: (a) The s^* point of m_1 is n_2 (see Definition 6).

shown in Figure 3. After the initialization, we topologically sort the netlist. For each block b_i in the topological order, we combine the DP-curves of all fanin blocks of b_i to derive a DP-curve of b_i . Excluding the power and delay of b_i , let $\tilde{\delta}_i$ and $\tilde{\rho}_i$ denote the accumulated fanin delay and power of b_i , respectively. We calculate $\tilde{\delta}_i$ and $\tilde{\rho}_i$ as follows:

$$\tilde{\delta}_i = \max_{j \in FI_i} \tilde{\delta}_j, \quad (4)$$

and

$$\tilde{\rho}_i = \sum_{j \in FI_i} \tilde{\rho}_j. \quad (5)$$

When combining points from the DP-curves of a fanin, if a point i has longer or equal delay compared with points in $S = \{s_1, s_2, \dots, s_k\}$ from another fanin, we should select a point s^* from S , such that s^* consumes the least power, shown in Figure 4. This selection guarantees that the resulting delay (Equation (4)) will not be over the delay of i , and the resulting power (Equation (5)) is minimized. We define the desired point s^* as follows.

DEFINITION 6. (s^* Point) *Given a point i of a fanin DP-curve and another fanin DP-curve C , assuming $S = \{s_j | s_j \in C, s_j.x \leq i.x\}$, then the s^* point of i is the point $s_j \in S, s_j.x > s_k.x, \forall s_k \in S, k \neq j$, where $s.x$ denotes the x -coordinate of s .*

By selecting only the s^* points, the number of points in the intermediate DP-curve grows only linearly, since every point has at most one s^* point in any other fanin's DP-curve.

3.2.2 Generating Points of DP-curves with Level Shifters

To calculate the accumulated delay δ_i and power ρ_i of a block b_i , including the delay and power of b_i , we need to simultaneously consider the contribution of delay and power from level shifters. Thus, δ_i and ρ_i are calculated by

$$\delta_i = \tilde{\delta}_i + d_i + x_{ij} \cdot d_s, \quad (6)$$

and

$$\rho_i = \tilde{\rho}_i + p_i + x_{ij} \cdot p_s, \quad (7)$$

where x_{ij} is a 0-1 variable indicating whether a level shifter is needed from block j to block i (1 if needed; 0, otherwise), d_i (p_i) is the delay (power) of b_i , and d_s (p_s) is the delay (power) of a level shifter.

Directly combining all fanin DP-curves may lose some useful points when level shifters are considered. In Figure 5, taking (m_1, n_1, f_1) and (m_1, n_2, f_1) for example, if b_m combines with b_n first, point $(3,11)$ constructed from (m_1, n_1) is dominated by point $(3,8)$ constructed from (m_1, n_2) . So point $(3,11)$ is pruned. However, this pruning is incorrect, since the effects of level shifters are not considered. Assuming that the delay and power of a level shifter are 2, point p constructed from (m_1, n_1, f_1) does not need any level shifters, but point q constructed from (m_1, n_2, f_1) needs a level shifter

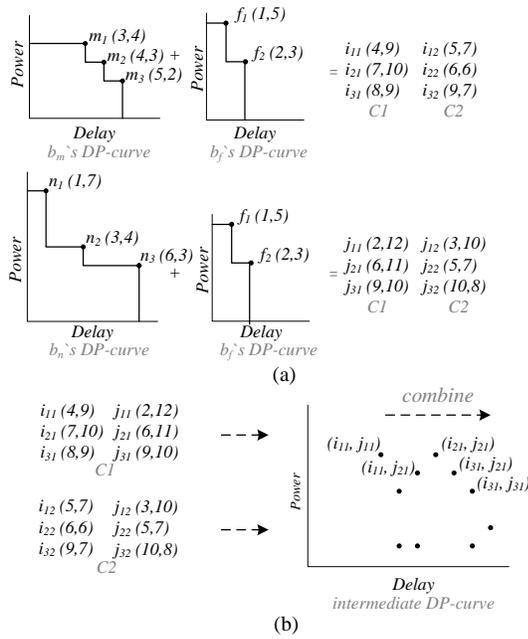


Figure 5: Suppose that b_m and b_n are two fanins of b_f . (a) Generate an individual joint DP-curve with b_f for each supply voltage and check if a level shifter is needed. Due to the space limit, the individual joint DP-curves (C_1 and C_2) are represented in text. (b) Combine all joint DP-curves for each supply voltage by using the lower-bound merge operation.

between b_n and b_f . Thus, p is $(4, 16) = (3, 11) + (1, 5)$, and q is $(6, 15) = (3, 8) + (1, 5) + (2, 2)$. In the final result, p cannot be dominated by q , but p cannot be held if we combine all fanin DP-curves first.

The following procedure prevents from over-pruning points when considering level shifters. Suppose a block b_f has two fanins b_m and b_n , shown in Figure 5. Join b_m with b_f and derive a joint DP-curve C_k for each supply voltage k of b_f ($VDD1$ and $VDD2$). The points $i_{h,k} = (\delta_i, \rho_i)$ of C_k is produced by m_h in b_m and f_k in b_f using Equations (6) and (7). Then, joining b_n with b_f in the same way results in the points $j_{h,k}$'s.

After deriving the joint DP-curves for each candidate supply voltages (C_1 and C_2), we can derive the intermediate DP-curve for each supply voltage by combining the joint DP-curves of the same voltage domain, using the lower-bound merge operation mentioned in the preceding section. However, it should be noted that because the power of b_i is added for each fanin repeatedly, the over-added power must be subtracted.

3.2.3 Constructing a Monotonic Decreasing DP-curve

After producing points of a new DP-curve, a monotonic decreasing DP-curve can be constructed by a line-sweeping algorithm. The line-sweeping algorithm consists of two steps: sorting and pruning. First, sort all points by the y -coordinate from the smallest to the largest, shown in Figure 6(a). In this figure, point i_j means that the point is the j th lowest in a DP-curve.

DEFINITION 7. (Point Dominance) In a DP-curve, a point i dominates another point j iff $i.x < j.x$ and $i.y < j.y$, where $i.x$ and $i.y$ denote the x - and y -coordinates of i , respectively.

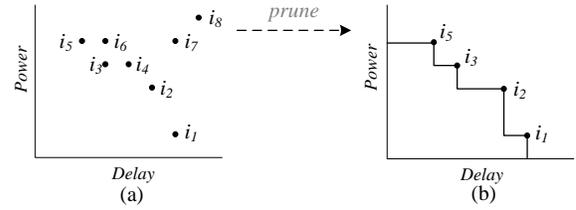


Figure 6: (a) Sort all points by y -coordinate. i_j represents that this point is the j th low in a DP-curve. (b) The final result.

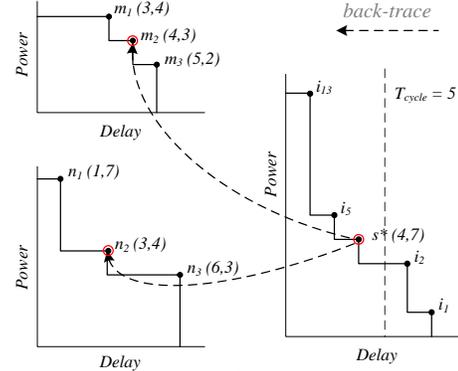


Figure 7: The backtracing procedure for getting a solution. (a) Determine the best result according to PO 's DP-curve and T_{cycle} .

After sorting, we prune the points which are dominated. Since points have been sorted by their y -coordinates, a point i is in front of a point j ($i.y \leq j.y$), such as i_1 is in front of i_2 . Thus, if $i.x \leq j.x$, j is dominated by i . More precisely, check a point if its x -coordinate is larger than that of the previous one. Figure 6 illustrates the process of the monotonic decreasing chain generation.

3.2.4 Backtracing to Find a Solution

Having generated a new DP-curve, we need to trace a netlist and get an optimal solution of voltage assignment. We determine the solution point s^* according to T_{cycle} , and the delay and power of this circuit are decided simultaneously. Repeat tracing solutions until PIs.

THEOREM 1. Given a netlist without reconvergent fanouts, an optimal solution for the voltage assignment problem can be obtained by our dynamic programming in linear time.

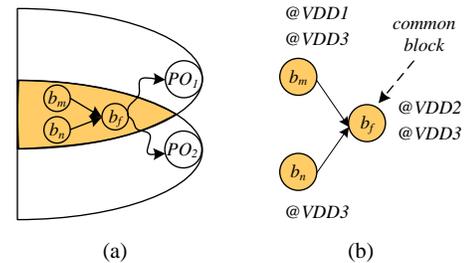


Figure 8: The shaded portion indicates common blocks. (a) PO_1 and PO_2 share some blocks, as in the shaded portion. (b) After backtracing a solution, these common blocks may be set in several different voltages.

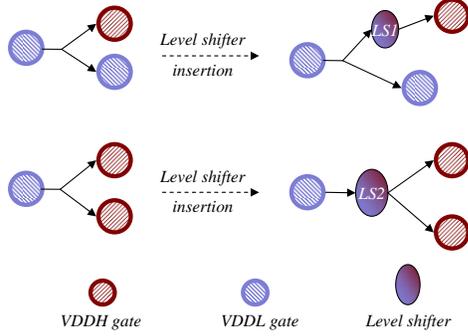


Figure 9: An example level-shifter block insertion. LS1 is smaller than LS2 since the fanout load of LS1 is smaller than that of LS2.

DEFINITION 8. (Common Block) From POs to PIs, different timing paths re-converge in some blocks. Among all these blocks, the block which is closest to POs is defined as a common block.

Due to common blocks, we need two passes to deal with the voltage assignment problem. The first pass works the same as described in Section 3.2.2. After the first pass, a common block may be assigned several different voltages, since different paths may set the common block in different voltages. For those voltages, we assign a highest one to a block, and then apply dynamic programming from the common block to POs. The second pass can make a solution better by helping us use more timing budget which is saved from common blocks. Avoiding wasting timing budgets, the second pass is thus needed.

3.3 Level Shifter (Soft Block) Insertion

This is the Phase II of our proposed algorithm flow. Level shifters are inserted into a net that connects two blocks in different power domains. After voltage assignment, we trace the circuits from PI's to PO's to search the nets that need level shifters by breadth-first search (BFS).

In this paper, we treat level shifters as soft blocks. A soft block in an interconnection contains all needed level shifters. The number of level shifters in an interconnection is equal to the number of bits in an interconnection. Thus, we insert a level-shifter block according to the interconnection width (in bits). Another issue is that a larger fanout load needs a larger level shifter to drive it. See Figure 9 for an illustration. The fanout load in Figure 9(a) is smaller than that of Figure 9(b), and so is the level shifter in Figure 9(a) than that of Figure 9(b).

3.4 Power-Network Aware Floorplanning

The objective in this phase is to find a floorplan which simultaneously minimizes the power-network resource requirement (Definition 3) and satisfies the timing (Definition 2) and the fixed-outline constraints. Hence, we propose a cost function (Equation (8)) to minimize the power-network resource without violating the constraints. Given a B*-tree T representing a floorplan of a set of blocks $B = \{b_1, b_2, \dots, b_n\}$,

$$\Phi(T) = \alpha \Phi_{PNR} + (1 - \alpha) \Phi_{area} + \Phi_{timing} + \Phi_{outline}, \quad 0 \leq \alpha \leq 1, \quad (8)$$

where Φ_{PNR} is the power-network resource of B , Φ_{area} is the area of the floorplan, and α is a weighting factor. Note that the four terms are all normalized to the same scale

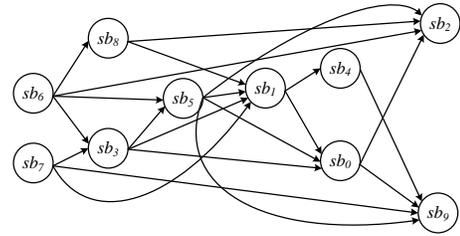


Figure 10: The netlist of n10.

order in advance.

In addition, for each net i , $1 \leq i \leq p$, net i has q fanout blocks $\{f_{i1}, f_{i2}, \dots, f_{iq}\}$, a fanin block b_i , and a wirelength upper-bound o_i (see Equation (3)). Let l_{ij} be the half-perimeter wirelength (HPWL) of the bounding box of b_i and f_{ij} . Then the timing violation penalty Φ_{timing} is defined as

$$\Phi_{timing} = \sum_{i=1}^p \max\left(\sum_{j=1}^q l_{ij} - o_i, 0\right). \quad (9)$$

Similarly, we give a floorplan the fixed-outline violation penalty, $\Phi_{outline}$, if the floorplan exceeds the desired fixed-outline, by

$$\Phi_{outline} = (R - R^*)^2, \quad (10)$$

where R^* (R) is the aspect ratio of the desired fixed-outline (the current floorplan).

4. EXPERIMENTAL RESULTS

Our algorithm was implemented in the C++ programming language and executed on a Linux machine with a 3.20 GHz CPU and 2GB Memory. We tested on the GSRC floorplan benchmarks. Since the information in the GSRC benchmark is not sufficient for voltage island optimization, we need to add some additional information for the experiment. For each testcase, it was carried out in the following steps:

Step 1: We assign the direction (input/output) for each PAD and each net; then the GSRC benchmarks can be modelled by a directed acyclic graph (DAG).

Step 2: After constructing the corresponding DAG, we assign the timing and power consumption for each block.

Table 1 shows the voltage assignment results. There are two factors that affect the experimental results. One is non-critical blocks and the other is common blocks. The third and fourth columns show the respective number of critical and non-critical blocks in each testcase. We find that the ratio of critical blocks to non-critical blocks in n30 is 2:3, and that in n300 is 1:4. In a small testcase, if the ratio is high, we cannot achieve much power saving. On the other hand, all the testcases have many common blocks. For example, Figure 10 shows the DAG of n10, in which there are many common blocks in n10. Those common blocks will decrease the power saving (see Section 3.2.4). In the sixth column, we show the total power saving of each testcase; the results show that our algorithm is effective to reduce power consumption by up to 19.75%. Further, practical designs will be simpler than our testcases (more non-critical blocks and fewer common blocks), so we expect that our algorithm will achieve more power saving for practical designs.

Table 2 shows the effectiveness of our power-network aware floorplanner (PN-FP, setting α in Equation 8 to 0.6). Compared with a traditional area-aware floorplanner (A-FP, set-

Table 1: Phase I: Voltage assignment results using the DP method

Ckts	Original Design			Dynamic Programming						
	Total Power (in VDDH)	Critical blocks	Non-Critical blocks	Total Power (with LS)	Power Saving(%)	VDDL blocks #	VDDH blocks #	LS blocks #	Ratio (VDDL/Non-Critical)	Runtime (sec)
n10	216841	10	0	216841	0	0	10	0	0	0.001
n30	205650	12	18	190717	7.26	6	24	57	0.333	0.069
n50	195140	29	21	172884	11.40	19	31	119	0.904	65.360
n100	180022	34	66	179876	0.10	39	61	92	0.590	664
n200	177633	42	158	174818	1.58	120	80	399	0.759	1637
n300	273499	60	240	219492	19.75	147	153	452	0.613	844

Table 2: Phase III: Floorplanning results of a traditional area-aware floorplanner (A-FP, $\alpha = 0$) and our power-network aware floorplanner (PN-FP, $\alpha = 0.6$). The fixed-outline constraint is set to [800, 800].

Netlist Information					Power-Network Resource		Area		Wirelength		Runtime (sec)	
Name	Net	VDDL	VDDH	Level Shifter	A-FP	PN-FP	A-FP	PN-FP	A-FP	PN-FP	A-FP	PN-FP
n10	118	0	10	0	965	965	233024	233024	1729	1729	8	6
n30	406	6	24	57	1650	1369	225379	229289	8184	8202	132	115
n50	604	19	31	119	1964	1514	242243	251678	16423	16395	600	504
n100	977	39	61	92	2024	1671	259918	272265	18716	18734	1430	1104
n200	1842	120	80	399	2232	2040	314924	328517	20104	20128	2992	2575
n300	2231	147	153	452	2693	2147	457173	488684	26977	27026	4787	3956
Average					1921.3	1617.7	288776.8	300576.2	15355.5	15369.0	1658.2	1376.7
Difference (%)					-15.81		+4.09		+0.09		-16.98	

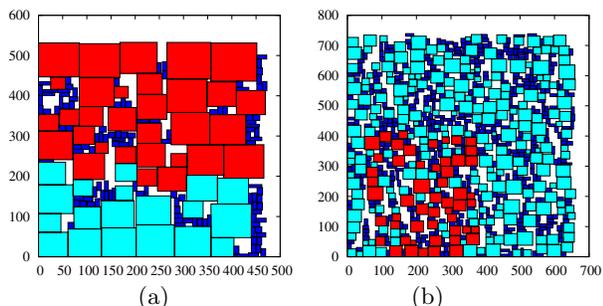


Figure 11: The power-network aware floorplans of n50 and n300 are shown in (a) and (b) respectively. VDDH blocks, VDDL blocks, and level shifters are colored in red, light blue, and deep blue, respectively.

ting α to 0), PN-FP indeed reduces the power-network resource by 16% with a reasonable overhead of 4% more area, on the average. As for timing requirement, both floorplanners produce timing-satisfied floorplans with a negligible difference of total wirelength. Besides the effectiveness, PN-FP even runs faster than A-FP by 17% less runtime. This could result from that, during SA, the cost function simultaneously considering area and power-network resource may have a faster converging rate than that considering area alone. Empirically, PN-FP significantly reduces power-network usage with a slight overhead of area.

Figure 11 shows two resulting floorplans. Blocks of the same supply voltage are almost clustered together to reduce the power-network resource, while level shifters are spread around to meet the timing constraint. Interestingly, if the areas of different voltage islands are balanced, e.g., Figure 11(a), the distribution of islands are nearly bi-partitioned to reduce the power-network resource. Otherwise, the smaller-area voltage island would be grouped, surrounded by the larger-area island, e.g., Figure 11(b). These experimental results reveal that our PN-FP is very effective.

5. CONCLUSIONS

In this paper, we have proposed a dynamic programming

based voltage scaling algorithm and a power-network aware floorplanning for the MSV design. The experimental results have shown that our algorithm is very effective in reducing power (up to 19.75%) and power resource (15.81%) with a reasonable area overhead of 4%.

6. REFERENCES

- [1] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, and S.-W. Wu, "B*-trees: A New Representation for Non-slicing Floorplans," *Proc. DAC*, pp.458–463, 2000.
- [2] T.-C. Chen and Y.-W. Chang, "Modern Floorplanning Based on Fast Simulated Annealing," *Proc. ISPD*, pp. 104–112, April 2005.
- [3] K. Chaudhary and M. Pedram, "Computing the Area Versus Delay Trade-Off Curves in Technology Mapping," *IEEE Trans. on Computer-Aided Design*, vol. 14, Dec. 1995.
- [4] J. Chang and M. Pedram, "Energy Minimization Using Multiple Supply Voltages," *Proc. ISLPED*, pp. 157–162, 1996.
- [5] J. Chang and M. Pedram, "Energy Minimization Using Multiple Supply Voltages," *IEEE Trans. on VLSI Systems*, vol. 5, Dec 1997.
- [6] M.-C. Wu and Y.-W. Chang, "Placement with Alignment and Performance Constraint Using the B*-tree Representation," *Proc. ICCD*, pp.568–571, 2004.
- [7] A. B. Kahng, "Classical Floorplanning Harmful?" *Proc. ISPD*, pp. 207–213, April 2000.
- [8] Kirkpatrick, Gelatt, and Vecchi, "Optimization by Simulated Annealing," *Science*, May 1983.
- [9] S. H. Kulkarni, A. N. Srivastava, and D. Sylvester, "A New Algorithm for Improved VDD Assignment in Low Power Dual VDD Systems," *Proc. ISLPED*, pp. 200–205, 2004.
- [10] R. Puri, L. Stok, J. Cohn, D. Kung, D. Pan, D. Sylvester, A. Srivastava, and S. Kulkarni, "Pushing ASIC Performance in a Power Envelope," *Proc. DAC*, pp.788–793, 2003.
- [11] K. Usami and M. Horowitz, "Clustered Voltage Scaling Technique for Low-Power Design," *Proc. ISLPED*, pp.3–8, 1995.
- [12] 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 Trans. on Solid-State Circuits*, pp.463–472, 1998.
- [13] H. Wu, I. M. Liu, Martin D. F. Wong, and Y. Wang, "Post-Placement Voltage Island Generation Under Performance Requirement," *Proc. ICCAD*, pp.309–316, 2005.
- [14] Y.-J. Yeh and S.-Y. Kuo, "An Optimization-Based Low-Power Voltage Scaling Technique Using Multiple Supply Voltage," *Proc. ISCAS*, pp.535–538, 2001.