High-Quality Global Routing for Multiple Dynamic Supply Voltage Designs

Wen-Hao Liu[1], Yih-Lang Li[1], and Kai-Yuan Chao[2]
[1]Department of Computer Science, National Chiao-Tung University, Hsin-Chu, Taiwan
[2]Intel Architecture Group, Intel Corporation, Hillsboro, OR
dnoldnol@gmail.com, ylli@cs.nctu.edu.tw, kaiyuan.chao@intel.com

***Abstract* - Multiple dynamic supply voltage (MDSV) provides an effective way to reduce dynamic power and is widely used in high-end or low-power designs. The challenge of routing MDSV designs is that the net in MDSV designs needs to be planned carefully to avoid electrical problems or functional failure as a long interconnect path pass through the shut-down power domains. As the first work to address the MDSV global routing problem, power domain-aware routing (PDR) problem is defined and the point-to-point PDR algorithm is also presented herein with look-ahead path selection method and look-up table acceleration approach. For multi-pin net routings, a novel constant-time table-lookup mechanism by invoking four enhanced monotonic routings to fast compute the least-cost monotonic path from every node to the target sub-tree is presented to speed up the query about routing cost (including driven-length slack) to target during multi-source multi-target PDR. Experimental results confirm that the proposed MDSV-based global router can efficiently identify legally optimized routing results for MDSV designs, and can effectively reduce overflow, wire length, inserted level shifters and runtime.**

1. INTRODUCTION

As the device density in a die increases, the consumed power increases significantly. High power consumption shortens the battery life of handheld devices and causes thermal and reliability problems. Total power consumption generally includes both dynamic power and static power. Static power is produced by leakage current while dynamic power is produced by the switching activities of the device. In modern circuit design, most of the total power consumption is dynamic power consumption, which is proportional to the square of supply voltage V_{dd}. Although reducing supply voltage can substantially reduce dynamic power consumption, it also slows device speed. Multiple supply voltage (MSV) is commonly used in high-end or low-power applications because it effectively reduces dynamic power with a sophisticated control on different function units' voltages. MSV may also apply power gating to further reducing leakage power. The sophisticated dynamic voltage control usually needs to be implemented in architectural function units with physical power supply controlled by sophisticated logics. Therefore, it is common to implement layout in a region-based power domains to save silicon resources (metal, level shifters, gated devices, IO-to-package, etc,).

In region-based MSV designs, circuits are partitioned into several power domains (or voltage islands), each of which occupies a contiguous physical space and operates at a main supply voltage or power-down mechanism (except few mixed voltage local areas such as level shifter blocks). Power domain floorplan structures may be either block-based (such as slicing structure with iterative bisecting generation process), or non-block-based structure. Figures 1(a) and 1(b) are the placement results of [1] and [2], that compare the block-based structure and non-block-based structure, respectively. Because different power domains can operate at different supply voltage levels, performance-critical devices such as processing units usually import higher supply voltages while other devices such as memories can operate at low supply voltage. Recently, multiple dynamic supply voltage (MDSV) technique is adopted to further reduce power consuming. In MDSV designs, the supply voltage of each power domain dynamically changes according to the power mode. In some power modes, such as waiting and sleeping modes, some power domains may even be shut down completely to save power. Figure 2 shows an MDSV design with ten power domains and four power modes. For instance, the operation modes for power domain *P1* are under $1.2V$ or under $1.0V$. In Active 3 mode, *P1* is turned off.

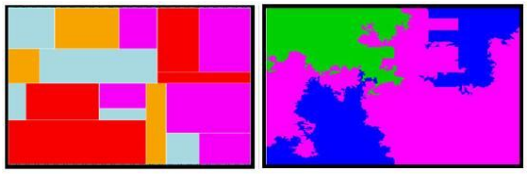


Fig. 1. (a) Block-based structure of multi-supply voltages, there are 17 power domains; (b) non-block-based structure of multi-supply voltages, there are 5 power domains.

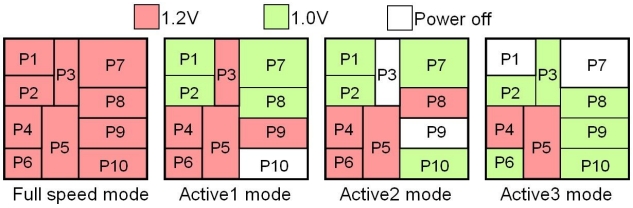


Fig. 2. An MDSV design with ten power domains and four power modes.

In traditional global routers [3-11], efficient routing algorithms have been developed to reduce overflows, wirelength, and computation time. The work in [12] is the first study to address MSV issue in global routing and presented a mathematical formulation to minimize power consumption of a given global routing for a MSV design, where integer programming-based techniques were proposed for solving the formulation. However, these routing algorithms cannot solve global routing problems in MDSV designs. To our knowledge, this work is the first to propose an MDSV-based global router. The nets in an MDSV design

* This work was partially supported by Synopsys and the National Science Council of Taiwan by Grant Nos. NSC 99-2220-E-009 -012 and NSC 99-2220-E-009 -036.

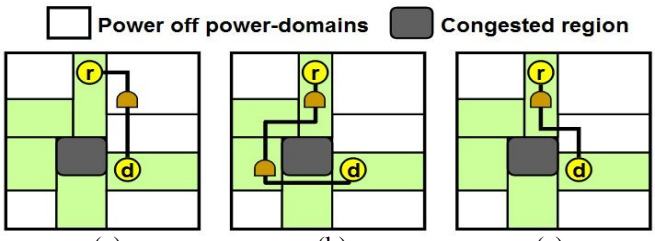


Fig. 3. (a) the routing result of traditional global routing approaches; (b) a legal routing result as the routing path is restricted in non-shutdown regions; (c) another legal routing result with less wirelength.

may cross more than one power domain and some power domains may be turned off while the others are remained in active modes. An active net that travels long distances in a shut-down power domain may cause a function violation if its repeaters are inserted in a power domain that are shut down during the active period of the net. Hence, limiting the walking length of an active net in shut-down power domains is an important global routing issue in MDSV designs. A recent study [13] presented a bounded-length maze routing algorithm for controlling the wire length of routing path, but it does not determine whether the length constraints at different power domains are different. Hence, the global router in [13] still may produce an illegal routing result in MDSV designs, as revealed in our experiments.

The contribution of this study is to present an MDSV-based global router that can efficiently identify a legally high-quality routing result for MDSV designs with overflow, wirelength and level shifter minimization based on the following proposed methods.

(a) A point-to-point power domain-aware routing (PPPDR) with look-ahead based acceleration approach is proposed to involve identifying a minimal-cost path from a net's source to its target under the limitation of MDSV designs.
(b) A novel constant-time table-lookup mechanism by invoking four enhanced monotonic routings to fast compute the least-cost monotonic path from any node to a target sub-tree is proposed to hasten the query about routing cost to target, including driven-length slack, during multi-source multi-target power domain-aware routing (MMPDR).
(c) A power domain-aware minimum spanning tree (PDMST) is proposed for use as the initial tree topology for each net. In MDSV designs, nets adopting this topology can decrease the final routing wire length and the number of level shifters required.

The rest of this paper is organized as follows. Section 2 describes the global routing problem and the routing limitation of MDSV designs. Section 3 and Section 4 present the proposed power domain-aware routing and power domain-aware minimum spanning tree, respectively. Section 5 displays the design flow of the proposed MDSV-based global router. Section 6 summarizes the experimental results. Finally, Section 7 draws conclusions.

2. PROBLEM DESCRIPTION

2.1. Traditional Global Routing Problem

The global routing is formulated as the routing problem on a grid graph $G(V, E)$, where V denotes the set of grid cells, and E denotes the set of grid edges. Typically the layout is partitioned into an array of global cells (G-cells). Each grid edge is termed by the adjacency of the related G-cells to its two end nodes. The capacity $c(e)$ of a grid edge e indicates the number of routing tracks that cross the abutting boundary. The number of wires that pass through grid edge e is called the demand of the grid edge $d(e)$. The overflow of a grid edge e ($overflow(e)$) is defined as the amount of demand in excess of capacity. Moreover, the total overflow is the sum of overflows on all grid edges of E. The optimization order of global routing problem is to minimize the total overflow, total wirelength, and runtime.

2.2. The Routing Limitations of Multiple Dynamic Supply Voltage Designs

In MDSV designs, the routing path of a net is considered legal if the path satisfies the following two constraints:

(a) The path cannot pass through any domains in which its metal can be changed at the different product configurations.
(b) If the wire length of the path exceeds a threshold value, repeaters must be inserted on the routing path to prevent signal degradation. As the net is active, the repeaters must be placed in the non-shutdown power domains.

The first constraint is easily to be satisfied, those domains can just be regarded to the forbidden regions, and the nets are prevented from flowing through those forbidden regions. In the second constraint, a repeater can be a conventional buffer, an inverter, or even a flop-repeater if the global router is used for architectural floorplan pipeline planning [14]. Placement of a repeater in shut-down power domains produces signal distortion in the net. Thus, an important issue in MDSV-based global routing is limiting the walking length of an active net in shut-down power domains. Because of their resulting complexities, existing global routing approaches often produce illegal routing results in MDSV designs. For example, in Fig. 3(a), net n_i consists of a driver d and a receiver r, the white tiles are the shut-down power domains, the gray rectangle indicates the congested region and the routing wires are not encouraged to pass through the congested region. Figure 3(a) shows the minimum-cost routing path connecting the two terminals, which is found by traditional global routing approaches with a repeater inserted at post-global-route stage. However, as the supply voltage of the repeater is turned off, the signal of the net will degrade. Thus the routing path in Fig. 3(a) is illegal. Figure 3(b) demonstrates a legal but longer routing result with two repeaters that is limited to be placed in non-shut-down power domain, and thus more consumed routing resources than that in Fig. 3(a), which increases the difficulty of identifying overflow-free routing paths in subsequent routings. In order to overcome the drawbacks of Figs. 3(a) and 3(b), we defined a *driving length constraint*. If a net conforms to the

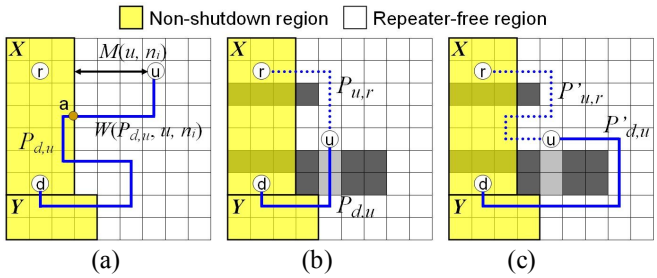


Fig. 4. (a) Define the notations used in the work; (b) a routing path $P_{d,u}$ with higher routing cost and shorter $W(P_{d,u}, u, n_i)$; (c) a routing path $P'_{d,u}$ with less routing cost and longer $W(P'_{d,u}, u, n_i)$.

driving length constraint, it can be legally enhanced by inserting repeaters with shorter wire length. The terms *repeater-free region* and *driving length constraint* are defined as follows.

Definition 1. ***Repeater-free region***: In any power mode of an MDSV design, if a net n_i is active when a power domain p is shut down, p is the repeater-free region associated with net n_i. For example, in Fig. 2, if a net n_i belongs to the power domain P9 (the driver of n_i is in P9), n_i is active in full speed mode, active1 mode and active3 mode. Thus, the repeater-free regions for n_i are P1, P7 and P10. Namely, other regions are the *non-shutdown regions* associated with n_i because they have supply voltages when n_i is active. The repeaters of n_i can be inserted in the non-shutdown regions but cannot be inserted in the repeater-free regions.

Definition 2. ***Driving length constraint***: The driving length of a repeater is the maximum wire length that can be driven by the repeater without causing signal distortion or violating design timing requirements. The driving length of a repeater increases as its supply voltage increases. In the case where the driver of net n_i is in power domain X and power domain Y is a non-shutdown region for n_i, $V_x(Y)$ represents the lowest voltage level of Y when X is active, and $L_x(Y)$ represents the driving length of the repeater in the voltage $V_x(Y)$. If the path of n_i enters its repeater-free regions from power domain Y, the continuous wire length in the repeater-free regions cannot exceed $L_x(Y)$.

The driving length constraint satisfies the minimum requirements for a legal routing path. Because, if a repeater is inserted on the legal path within the boundaries of the power domain Y, the path, after passing through repeater-free regions, can return to non-shutdown regions before signal distortion occurs. Figure 3(c) shows that the routing path conforms to the driving length constraint, in which the wire passes through the repeater-free region, and in which the repeater is inserted on the wire within the non-shutdown regions. The wire length of Fig. 3(c) is shorter than that in Fig. 3(b). To make all routing paths are legal, the **MDSV-based global routing problem** is formulated as follows.

Input: Given a 5-tuple (G, PDS, M, F, N), G represents a grid graph, and PDS represents a set of block-based-structure power domains. Each power domain is a polygon on the G, and no power domains overlap. M denotes a set of power modes and F denotes a set of forbidden regions. $N=\{n_1,...,n_m\}$ represents a set of nets. A net n_i, $1\le i\le m$, consists of a driver and a set of receivers. If the driver of n_i is in the power domain X, n_i is active as X is active. The repeater-free regions for each net can be identified by *Definition 1*.

Objective: The MDSV-based global routing problem routes N onto G with minimized total overflow first and then total wirelength. Note that, if a net connects a driver of lower voltage to a receiver of higher voltage, level shifters need to be inserted on the path at the boundaries of higher voltage power domain. Because level shifter demands extra penalty in area, delay and power, we presents an initial tree construction scheme to reduce required level shifters in section 4.

Constraint: Each net avoids passing through the forbidden regions and must conform to the driving length constraint.

3. POWER DOMAIN-AWARE ROUTING (PDR)

In order to route each net under driver length constraint, this work proposes a point-to-point PDR (PPPDR) algorithm and a multi-source multi-target PDR (MMPDR) algorithm. Before detailing the proposed PPPDR and MMPDR algorithms, the notations used in Fig. 4(a) are defined as follows.

(a) $P_{d,u}$ is a routing path from driver d to node u;
(b) $M(u, n_i)$ is the minimum Euclidean distance between node u and the boundary of the nearest non-shutdown region of net n_i;
(c) $W(P_{d,u}, u, n_i)$ is the wirelength of wire segments of $P_{d,u}$ from u to the boundary of a non-shutdown region of n_i.

If u is in the non-shutdown region, $M(u, n_i)$ and $W(P_{d,u}, u, n_i)$ are both zero. In Fig. 4(a), $M(u, n_i)$ is *4* and $W(P_{d,u}, u, n_i)$ is *6* (from u to a).

3.1. PPPDR with Two Accelerated Methods

The point-to-point PDR (PPPDR) problem is formulated as follows. Given a 7-tuple (d, r, X, G, NSR, RFR, L_x), d and r denote the driver and the receiver of the routed net n_i, respectively. Net n_i belongs to the power domain X. G is the congestion graph and each grid edge in G has a specified congestion cost. NSR and RFR represent the sets of non-shutdown regions and repeater-free regions for n_i, respectively. L_x is the set of driving length constraints for n_i. For instance, if the route of n_i enters a repeater-free region from power domain Y (Y is a non-shutdown region), the continuous wirelength in the repeater-free region cannot exceed $L_x(Y)$ that can be identified by *Definition 2*. The objective of PPPDR is to identify a minimal-cost path from d to r on graph G where the continuous wirelength in repeater-free regions does not exceed $L_x(Y)$. If the continuous wirelength in repeater-free regions exceeds $L_x(Y)$, a driving length violation occurs.

The PPPDR regards d and r to the source and the target, respectively. The PPPDR stores all current paths in a heap. The heap is initialized to have only the driver d. In every step, the minimum-cost path is extracted from the heap for further expansion if the target has not been reached. The new path to adjacent G-cell u is stored if the new path conforms to the driving length constraint ($M(u, n_i)+W(P_{d,u}, u, n_i)\le L_x(Y)$) and

there is no existing path to reach u in the heap. For the case with another path to reach u in the heap, only the better path is reserved in the heap. How to reserve a better path in the heap is the main issue of PPPDR.

3.1.1. Look-Ahead Path Selection

Figures 4(b) and 4(c) show two routing paths $P_{d,u}$ ($W(P_{d,u}, u, n_i)$=4) and $P'_{d,u}$ ($W(P'_{d,u}, u, n_i)$=10), respectively, to tell the difference of selecting surviving path in the heap between maze routing and this study. The $P_{d,u}$ has higher routing cost than $P'_{d,u}$ does because $P_{d,u}$ passes through two congested regions (light gray G-cells). Traditional maze routing prefers to reserve $P'_{d,u}$. However, if $L_x(Y)$ is 12 and $P_{u,r}$ is the minimum-cost path from u to r, $P'_{d,u}$ cannot legally connect to r by $P_{u,r}$ since the driving length slack of $P'_{d,u}$ is insufficient. Instead, $P_{d,u}$ plus $P_{u,r}$ is a legal routing using less resources.

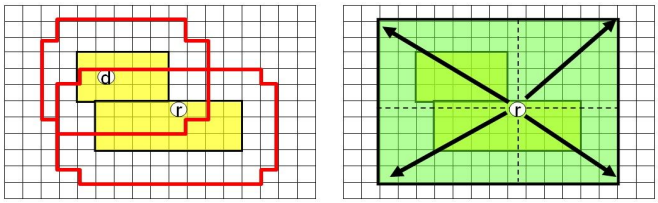
This study proposes a look-ahead path selection method for identifying a legal routing result with minimal routing cost. A path, say $P_{d,u}$, is considered *feasible* if the following equation is true:

$$W(P_{d,u}, u, n_i) + W(E_{u,r}, u, n_i) \le L_x(Y) \quad (1)$$

where $E_{u,r}$ is an u-to-r path that is estimated by identifying a least-cost monotonic path (MP). The least-cost MP detects the congestion information between u and r so that the router can determine whether a path has enough driving length slack to bypass the congested regions from u to r. The proposed look-ahead path selection uses the following scheme to choose between $P_{d,u}$ and $P'_{d,u}$. If only one path is feasible, the feasible path is selected. If both paths are feasible, the least-cost path is selected. If neither path is feasible, the path with shorter $W(P_{d,u}, u, n_i)$ is reserved because it is more likely to bypass the congested regions from u to r that have wirelength shorter than $L_x(Y)$. For $P_{d,u}$ and $P'_{d,u}$ in Figs. 4(b) and 4(c), the look-ahead scheme selects $P_{d,u}$. For $P'_{d,u}$, the feasible way to reach target is to return to the power domain X to satisfy the driving length constraint and then, if necessary, to re-enter to the repeater-free region. The dotted path in Fig. 4(c) shows a legal connection for $P'_{d,u}$.

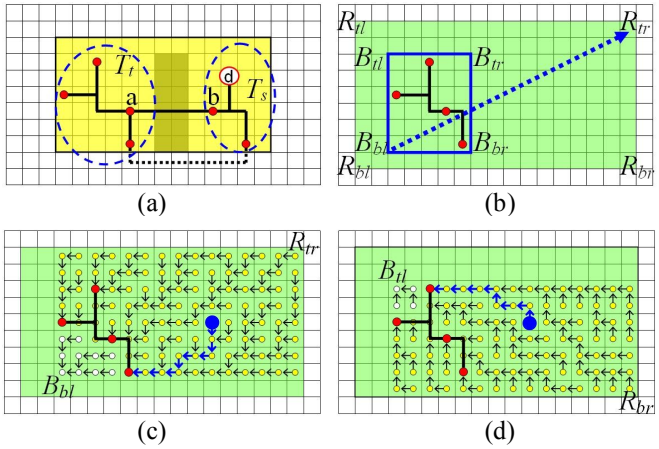
3.1.2. Accelerating Path Selection by Look-Up Table

Assume that V is the search region of PPPDR, the time complexity for look-ahead path selection is $O(|V|)$ because invoking monotonic routing requires the complexity of $O(|V|)$. To accelerate PPPDR, the proposed lookup table approach is used to perform look-ahead path selection in constant time. Before starting PPPDR for net n_i, the values for $W(E_{u,r}, u, n_i)$, where u is any node that can be explored in PPPDR, are stored in a lookup table. The PPPDR can access the table to look up $W(E_{u,r}, u, n_i)$ for each node in the selected path in constant time, which avoids invoking monotonic routing at every node and thus reduces the time complexity of $O(|V|)$ to $O(|1|)$. A node is considered useless if a path passing through this node always violates the driving length constraint. To prevent PPPDR from exploring the useless node and to minimize the size of the lookup table, the maximum search region (SR_{max}) of n_i is constructed by the following steps. Assume that $P(n_i)$ represents the set of power domains containing the pins of n_i. The *legitimate routing box* of



(a) (b)

Fig. 5. (a) Red boxes are the legitimate routing boxes of non-shutdown regions; (b) four monotonic routings from r to four corners of maximum searching region (SR_{max}) are performed.



(a) (b) (c) (d)

Fig. 6. (a) MMPDR result of a seven-pin net; (b) R and B are SR_{max} and B_{tar} of the routed net in (a); (c) the routing result of TPNMR from B_{bl} to R_{tr}; (d) the routing result of TPNMR from B_{tl} to R_{br}.

domain Y, $Y \in P(n_i)$, is obtained by enlarging the region of Y by $L_x(Y)/2$. Because node u in the legitimate routing boxes is potentially accessible by a legal routing path, the values of $W(E_{u,r}, u, n_i)$ must be pre-stored in the lookup table. The SR_{max} is the minimum area enclosing all legitimate routing boxes. In Fig. 5(a), the red boxes are the legitimate routing boxes for n_i. The green area in Fig. 5(b) is the SR_{max} of n_i. After SR_{max} is identified, four monotonic routings, from r to four corners of SR_{max}, are performed (Fig. 5(b)) to obtain the least-cost MPs from r to any node in the SR_{max}.

Lemma 1. *Four monotonic routings from r to four corners of SR_{max} determine the least-cost MPs from r to any node in SR_{max}.*

Proof. Since monotonic routing can be realized using dynamic programming algorithm, the monotonic routing from r to a corner also determines the least-cost MP of every node within the bounding box enclosing r and the corner. Accordingly, four monotonic routings indeed involve the least monotonic cost computation of each node in the SR_{max}, and each node u in SR_{max} with the exception of r has a predecessor node v. ■

Then the values of $W(E_{u,r}, u, n_i)$ can be calculated using the following equation before being stored in the lookup table.

$$W(E_{u,r}, u, n_i) = \begin{cases} W(E_{v,r}, u, n_i)+1 & \text{if } u \text{ is in the repeater-free region} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where node v is on the path r-u, it is the predecessor of node u.

3.2. Multi-Source Multi-Target PDR (MMPDR)

This study proposes multi-source multi-target PDR (MMPDR) to tackle multi-pin net routing. MMPDR is a tree-to-tree routing algorithm. Figure 6(a) shows a seven-pin net; a path $P_{a,b}$ (black bold line) passes through the overflowed G-cells. $P_{a,b}$ is ripped up and then two sub-trees T_s and T_t are obtained. Notably, T_s contains the driver. Next, MMPDR is used to identify a new routing path (dotted line in Fig. 6(a)) to connect T_s to T_t. At the beginning of MMPDR, the nodes on T_s are inserted into a heap; each node can be regarded as a zero-length path with zero routing cost. The subsequent propagation scheme of MMPDR is the same as that of PPPDR. However, the definition of feasible path in MMPDR differs from that of PPPDR. In MMPDR, a path $P_{d,u}$, is regarded as a feasible path if the following equation holds;

$$W(P_{d,u}, u, n_i) + W(E_{u,T_t}, u, n_i) \le L_x(Y) \qquad (3)$$

, where E_{u,T_t} represents the globally least-cost MP from node u to sub-tree T_t. The means of accelerating the operation of look-ahead path selection at every node is to build a lookup table to pre-store the values of $W(E_{u,T_t}, u, n_i)$ before MMPDR is performed. Initially, the searching region (SR_{max}) is determined, and the target box (B_{tar}) is set to be the minimum rectangle that encloses the target tree T_t. R_{bl}, R_{br}, R_{tr} and R_{tl} (B_{bl}, B_{br}, B_{tr} and B_{tl}) represent the four corners of SR_{max} (B_{tar}), respectively. Figure 6(b) displays the SR_{max} and B_{tar} of the net in Fig. 6(a). Next, the globally least-cost MPs from T_t to every node in SR_{max} (tree-to-all-nodes monotonic routing) are found by performing four modified point-to-point monotonic routings from B_{bl} to R_{tr}, B_{br} to R_{tl}, B_{tl} to R_{br} and B_{tr} to R_{bl}. Each modified point-to-point monotonic routing actually realizes a tree-to-partial-nodes monotonic routing (TPNMR), as shown in Fig. 7. Without loss of generality, the routing direction of TPNMR in Fig. 7 is from B_{bl} to R_{tr}. In Fig. 7, $d(u)$ represents the locally least MP cost from T_t to node u; $\pi(u)$ refers to the predecessor of u, and $cost(u, v)$ represents the congestion cost between u and its adjacent node v. Initially, the routing costs of the nodes on T_t are set to zero and the routing costs of the other nodes in SR_{max} are set to very large positive constants (line 1-3). Then, every node within the bounding box that encloses B_{bl} and R_{tr}, except for the nodes on T_t keeps its predecessor node and the locally least MP cost to T_t, by applying dynamic programming (line 4-17).

Figures 6(c) and 6(d) present the results of routing using TPNMRs from B_{bl} to R_{tr} and from B_{tl} to R_{br}, respectively. The yellow (white) circles represent the nodes that (do not) have a locally least-cost MP connection to T_t, and the arrow associated with each node points to its predecessor. Each TPNMR identifies for every node no more than three locally least-cost MPs that travel in one or two directions from node to T_t. Table 1 displays the routing directions of locally least-cost MPs that are identified by four TPNMRs. For instance, the routing from B_{bl} to R_{tr} identifies at most three locally least-cost MPs that travel leftwards, downwards, or leftwards and downwards from each node towards T_t. Thus, the globally least-cost MP that connects a node to T_t is the one with the least cost from the results obtained using four TPNMRs. In Figs. 6(c) and 6(d), the globally least-cost MP from the blue node to T_t is the one of the two blue locally least-cost MPs.

Algorithm Tree_to_Partial_Nodes_MN_Route
Input: source node $B_{bl}(x_1, y_1)$, target node $R_{tr}(x_2, y_2)$, tree T_t
1. **foreach** node u within the box enclosing B_{bl} and R_{tr}
2. **if**($u \in T_t$) $d(u)=0$, $\pi(u)$=nil;
3. **else** $d(u)=\infty$, $\pi(u)$=nil;
4. **for** $x= x_1+1$ **to** x_2
5. $u=(x, y_1)$, $v=(x\text{-}1, y_1)$;
6. **if**($u \notin T_t$) $d(u) = d(v) + cost(u, v)$, $\pi(u)=v$;
7. **for** $y= y_1+1$ **to** y_2
8. $u=(x_1, y)$, $v=(x_1, y\text{-}1)$
9. **if**($u \notin T_t$) $d(u) = d(v) + cost(u, v)$, $\pi(u)=v$
10. **for** $x= x_1+1$ **to** x_2
11. **for** $y= y_1+1$ **to** y_2
12. $u=(x, y)$, $v_1=(x\text{-}1, y)$, $v_2=(x, y\text{-}1)$;
13. **if**($u \notin T_t$)
14. **if** $d(v_1) + \text{cost}(u, v_1) < d(v_2) + cost(u, v_2)$
15. $d(u) = d(v_1) + cost(u, v_1)$, $\pi(u)= v_1$
16. **else**
17. $d(u) = d(v_2) + cost(u, v_2)$, $\pi(u)= v_2$
18. **end**

Fig. 7. The pseudo code of tree-to-partial-nodes monotonic routing.

Table 1. Routing directions of least-cost MPs from node to T_t.

	The direction of least-cost MP from node to T_t							
	L	R	T	D	TR	TL	DR	DL
B_{bl} to R_{tr}	√			√				√
B_{br} to R_{tl}		√		√			√	
B_{tl} to R_{br}	√		√			√		
B_{tr} to R_{bl}		√	√		√			

$W(E_{u,Tt}, w, n_i)$ is calculated using by Eq. (2) and then stored in the lookup table. The time complexity of TPNMR is $O(|V|)$, and the time complexity for building the lookup table for MMPDR is also $O(|V|)$. Finally, the worst time complexity of MMPDR is $O(|V|+|V|\log|V|)= O(|V|\log|V|)$.

4. POWER DOMAIN-AWARE MINIMUM SPANNING TREE

The initial tree topology of each net is a major factor in the quality of the final routing result. Traditional global routers usually focus on generating the initial tree topology with minimal wirelength and avoiding congestion. However, in MDSV designs, the distribution of power domains affects the final routing wirelength and the number of level shifters, so power domain distribution must be considered. For instance, the initial tree topologies in Figs. 8(a) and 8(b) have similar wire lengths. Figures 8(c) and 8(d) respectively show the final routing results of Figs. 8(a) and 8(b) when using the post-inserted level shifters and repeaters. The number of post-inserted level shifters in Fig. 8(c) is more than that of Fig. 8(d), and the wire length of the routing result in Fig. 8(c) is longer than that in 8(d) because the wire connecting the driver to the receiver r_3 (Fig. 8(c)) requires a detour into the non-shutdown region for inserting a repeater to avoid the driving length violation.

This work adopts power domain-aware minimum spanning tree (PDMST) instead of traditional minimum spanning tree

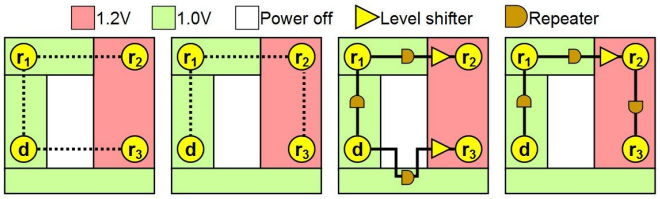


Fig. 8. (a) an initial tree topology; (b) another initial tree topology; (c) the final routing result of (a); (d) the final routing result of (b).

(MST) and Steiner minimum tree (SMT) for use as the initial tree topology for each net. The cost of a PDMST is sum of its wire length and the penalty for crossing power domains. An edge in a topology is defined as a crossing edge since its two terminals are located in different power domains. The penalty of crossing power domains for a tree t is defined as the number of crossing edges by t multiplied by the user-defined constant η. In the proposed implementation, η is set to 10. Because PDMST has fewer crossing edges compared to MST and SMT, the final routing results of PDMST are likely to have fewer level shifters and fewer detours compared to MST and SMT.

5. DESIGN FLOW of MDSV-BASED GLOBAL ROUTING

Figure 9 presents The design flow of the proposed MDSV-based global router. First, the algorithm decomposes each net into two-pin nets based on the topology of the PDMST (line 1), and then generates an initial congestion graph by monotonic routing (line 2). Next, the rip-up and rerouting stage iteratively reroutes the nets with overflows or driving length violation until an overflow-free routing result without driving length violation is obtained (line 3~line 13). In the rip-up and rerouting stage, maze routing is first adopted to reroute the net n_i (line 5~line 6). If the maze routing result of n_i violates the driving length constraint, then n_i will be ripped up and rerouted again using PPPDR (or MMPDR if n_i is a multi-pin net) (line 7~line 11). The routing result of n_i obtained by PPPDR or MMPDR always satisfy the driving length constraint.

6. EXPERIMENTAL RESULTS

The proposed algorithms were implemented in C/C++ language on a quad-core 3.0 GHz Intel Xeon-based PC with 32GB memory. Since no public MDSV global routing benchmarks are available, we modify ISPD'07 benchmarks to MDSV designs. Each ISPD'07 benchmark is randomly partitioned into several power domains and is set to several power modes. Then, the voltage level of each power domain in each power mode is randomly set between 0.8V to 1.2V. Note that, in some power modes such as sleep mode and waiting mode, some regions are set as shut-down power domains.

Columns 2-5 in Table 2 present the routing results obtained using a recently published method [13] that ignores the driving length constraint. In the table, #VN denotes the number of nets that violate the driving length constraint, and

Algorithm MDSV-based global router
Input: grid graph G, power domains PDS, power modes M, forbidden regions F, nets N
1. PDMST_Decomposition(N)
2. Monotonic_Routing(N, G)
3. **while (** G has overflows or driving length violation**) do**
4. **foreach** net n_i with overflows or driving length violation **do**
5. Rip_Up(n_i , G)
6. Maze_Routing(n_i , G, F)
7. **if** (n_i violates driving length constraint**)**
8. Rip_Up(n_i , G)
9. T←Building_Lookup_Table(n_i , PDS, M, N)
10. PPPDR(n_i , G, T, F**)** or MMPDR(n_i , G, T, F)
11. **end if**
12. **end foreach**
13. **end while**
14. **end**

Fig. 9. The design flow of the proposed MDSV-based global router.

TABLE 3 THE ROUTING RESULTS OF THE PROPOSED ROUTER WITH AND WITHOUT PDMST

	The proposed GR (w/o PDMST)		The proposed GR (w PDMST)	
	WL(10^5)	#LS	WL(10^5)	#LS
adaptec1*	52.61	30404	52.60	30155
adaptec2*	48.02	11695	48.01	11566
adaptec3*	122.33	44314	122.33	44053
adaptec4*	110.76	44294	110.76	43953
adaptec5*	149.08	66463	149.06	65925
newblue1*	45.38	10883	45.37	10659
newblue2*	74.52	42289	74.51	41514
Improve			0.012%	1.14%

OF, WL_1 and CPU_1 represent the total overflow, total wirelength and runtime of the results obtained using that method [13], respectively. Clearly, the conventional global router that does not consider the driving length constraint tends to cause numerous driving length violations.

Because no other study has addressed the MDSV global routing problem, we implement a global router RPD-GR to route MDSV designs for comparison with the proposed router. RPD-GR restricts the path of each net in non-shutdown regions to prevent the violation of driving length constraint. RPD-GR and the proposed router iteratively reroute the overflowed net until all overflows are eliminated or the overflows do not decline for five minutes. Columns 6-8 and 9-12 of Table 2 present the routing results obtained using RPD-GR and the proposed router, respectively. The #VN of both routers are not listed because both routers can produce the routing result without violating the driving length constraint. CPU_3 (CPU_4) represents the runtime of the proposed router without (with) look-up table acceleration. The routing results of adaptec4 and newblue1 produced by RPD-GR include many overflows, because the routing restriction increases the difficulty of identifying overflow-free routing paths, while the proposed router eliminates all overflows. Columns 13 and 14 compare the wirelengths and

runtimes of RPD-GR and the proposed router with look-up table acceleration. The proposed router with look-up table acceleration improves wirelength by an average of 1% and functions 1.92 times faster than PRD-GR. Notably, RPD-GR may generate the routing paths with many detours to bypass shutdown regions (Fig. 3(b)), while the proposed router can pass through shutdown regions under the driving length constraint to yield shorter routing paths (Fig. 3(c)). The final column reveals that the proposed look-up table acceleration approach increases the speed of the proposed router by a factor of 71.34.

Table 3 shows the effectiveness of each net adopting PDMST in the initial tree topology. The level shifters are post-inserted in the routing results of the proposed router with and without PDMST. In Table 3, the second and third (fourth and fifth) columns show the wire lengths and the numbers of level shifters of the proposed router without (with) PDMST, respectively. The proposed router with PDMST reduces the number of level shifters by 1.14% using almost the same wire length (less wire length by 0.012%) compared to the router without PDMST.

7. CONCLUSION

This paper discusses the global routing problem associated with MDSV designs and develops an MDSV-based global router that is based on the proposed PDR algorithm. Two novel enhancement methods, look-ahead path selection and a constant-time length slack query method using a look-up table, are proposed for PPPDR and MMPDR, and enhanced tree-to-all-nodes monotonic routing is utilized to pre-compute the least-cost MP of every node. Moreover, this work proposes a power domain-aware minimum spanning tree (PDMST) for use as the initial tree topology for each net, that can decrease the wirelength and the number of level shifters required of final routing result.The proposed router can yield legally high-quality results under the routing limitation of MDSV designs, with an average wirelength improvement of 1% and runtime speedup by a factor of 1.92 relative to PRD-GR.

REFERENCES

[1] H. Wu, I.-M. Liu, M. Wong, and Y. Wang, "Post-Placement Voltage Island Generation under Performance Requirement," *Proc. ICCAD*, pp. 309–316, 2005.

[2] R. Ching, E. Young, K. Leung, and C. Chu, "Post-Placement Voltage Island Generation," *Proc. ICCAD*, 2006.

[3] J. A. Roy and I. L. Markov, "High-performance routing at the nanometer scale,", *IEEE Trans. on Computer-Aided Design*, vol. 27, no. 6, pp. 1066-1077, 2008.

[4] M. D. Moffitt, "MaizeRouter: Engineering an effective global router," in *Proc. Asia and South Pacific Design Automation Conf.,* pages 232-237, 2008.

[5] M. Cho, K. Lu, K. Yuan, and D. Z. Pan, "BoxRouter 2.0: Architecture and implementation of a hybrid and robust global router," in *Proc. Int. Conf. Comput.-Aided Des.,* pages 503-508, 2007

[6] M. M. Ozdal and M. D.F. Wong, "ARCHER: A history-driven global routing algorithm," in *Proc. Int. Conf. Comput.-Aided Des.,* pages 488-495, 2007.

[7] Y.-J. Chang, Y.-T. Lee, and T.-C. Wang, "NTHU-Route 2.0: a fast and stable global router," in *Proc. Int. Conf. Comput.-Aided Des.,* pages 338-343, 2008.

[8] H.-Y. Chen, C.-H. Hsu, and Y.-W. Chang. "High-performance global routing with fast overflow reduction." in *Proc. Asia and South Pacific Design Automation Conf.,* pages 582–587, 2009.

[9] M. Pan and C. Chu, "FastRoute 2.0: A high-quality and efficient global router," In *Proc. Asia and South Pacific Design Automation Conf.*, pages 250-255, 2007.

[10] K.-R. Dai, W.-H. Liu, and Y.-L. Li, "Efficient Simulated Evolution Based Rerouting and Congestion-Relaxed Layer Assignment on 3-D Global Routing," in *Proc. Asia and South Pacific Design Automation Conf.,* pages 570-575, 2009.

[11] T.-H. Wu, A. Davoodi, and J. T. Linderoth, "A parallel integer programming approach to global routing" *in Proc. Des. Autom. Conf.*, pages. 194-199, June 2010.

[12] Tai-Hsuan Wu, A. Davoodi, and J. T. Linderoth, "Power-Driven Global Routing for Multi-Supply Voltage domains" *in Proc. DATE*, March 2011.

[13] W.-H. Liu, W.-C. Kao, Y.-L. Li and K.-Y. Chao, "Multi-Threaded Collision-Aware Global Routing with Bounded-Length Maze Routing" *in Proc. Des. Autom. Conf.*, pages. 200-205, June 2010.

[14] R. Lu, G. Zhong, C.-K. Koh, and K.-Y. Chao, "Flip-Flop and Repeater Insertion for Early Interconnect Planning," in *Proc. of 2002 Design, Automation and Test in Europe Conf.*, pp. 690-695, 2002

[15] W.-P. Lee, H.-Y. Liu, and Y.-W. Chang, "Voltage Island Aware Floorplanning for Power and Timing Optimization," *Proc. ICCAD*, pp. 389–394, 2006.

[16] D. Sengupta and R. Saleh, "Application-driven floorplan-aware voltage island design," in *Proc. Des. Autom. Conf.*, pages 155-160, 2008.

[17] W.-P. Lee, H.-Y. Liu, and Y.-W. Chang, "An ILP Algorithm for Post-Floorplanning Voltage-Island Generation Considering Power-Network Planning", in *Proc. ICCAD*, pp. 650–655, 2007.

[18] B. Liu, Y. Cai, Q. Zhou, and X. Hong, "Power Driven Placement with Layout Aware Supply Voltage Assignment for Voltage Island Generation in Dual-Vdd Designs," *Proc. ASPDAC*, 2006.

[19] H. Wu, M. Wong, and I.-M. Liu, "Timing-Constrained and Voltage-Island-Aware Voltage Assignment," *Proc. DAC*, pp. 429–432, 2006.

TABLE 2 COMPARISON BETWEEN AN INTUITIVE ROUTER AND THE PROPOSED ROUTER.

	[13]				RPD-GR			The Proposed Router				Comparison		
	#VN	OF	WL_1 (10^5)	CPU_1 (minute)	OF	WL_2 (10^5)	CPU_2 (minute)	OF	WL_3 (10^5)	CPU_3 (minute)	CPU_4 (minute)	$\frac{WL_2 - WL_3}{WL_2}$	$\frac{CPU_2}{CPU_4}$	$\frac{CPU_3}{CPU_4}$
adaptec1*	153	0	52.66	3.59	0	53.98	4.85	0	52.61	174.93	3.33	2.53%	1.45	52.53
adaptec2*	54	0	48.01	0.52	0	48.07	0.55	0	48.02	34.11	0.49	0.10%	1.14	70.18
adaptec3*	1481	0	121.36	2.28	0	122.67	2.62	0	122.33	148.70	2.18	0.27%	1.20	68.22
adaptec4*	622	0	109.83	0.81	212	111.39	6.02	0	110.76	60.72	1.04	0.57%	5.77	58.19
adaptec5*	1157	0	148.71	3.54	0	151.16	6.00	0	149.08	442.60	3.78	1.38%	1.59	117.23
newblue1*	92	0	45.36	47.41	396	46.31	50.60	0	45.38	3835.63	42.65	2.02%	1.19	89.93
newblue2*	84	0	74.51	0.52	0	74.62	0.59	0	74.52	22.86	0.53	0.13%	1.11	43.13
average												1%	1.92	71.34