PARR: Pin Access Planning and Regular Routing for Self-Aligned Double Patterning

Xiaoqing Xu, Bei Yu, Jhih-Rong Gao, Che-Lun Hsu, David Z. Pan
ECE Department, Univ. of Texas at Austin, Austin, TX, USA 78712
{xiaoqingxu, bei, jrgao, chsu1, dpan}@cerc.utexas.edu

ABSTRACT

Pin access has become one of the most difficult challenges for detailed routing in $14nm$ technology node and beyond, where double patterning lithography has to be used for manufacturing lower metal layers with tight pitches. Self-aligned double patterning (SADP) provides better control on the line edge roughness and overlay but it has very restrictive design constraints and prefers regular layout patterns. This paper presents a comprehensive pin access planning and regular routing framework (PARR) for SADP friendliness. Our key techniques include pre-computation of both intra-cell and inter-cell pin accessibility, as well as local and global pin access planning to enable the handshaking between standard cell level pin access and detailed routing under SADP constraints. Our experimental results demonstrate that PARR can achieve much better routability and overlay control compared with previous approaches.

Categories and Subject Descriptors

EDA8.1 [**Physical Design**]: Routing

General Terms

Algorithms, Design, Performance

Keywords

SADP, Regular Layout, Pin Access, Net Deferring

1. INTRODUCTION

In sub-$20nm$ technology node, pin access has become a critical challenge for detailed routing [1]. Due to the density increase or area reduction of the technology scaling, limited number of routing tracks are available for the standard cell (SC) design. It makes the local SC I/O pin access challenging because the access points of each pin available for the detailed router are limited and they interfere with each other [1, 2]. Furthermore, the continued technology scaling of the lower routing layers in $14nm$ node and beyond depends on the complex design-for-manufacturability (DFM) strategies. Extreme regular layout towards 1-D gridded design [3, 4] is a viable candidate for lower metal layers with increasing density. With 1-D gridded design, the line-space array decomposition can be applied to self-aligned double patterning (SADP) with tight control on overlay and wafer-print artifacts [5]. However, SADP-specific design rules and 1-D layout patterns impose even more complicated constraints on the SC I/O pin access for the detailed router [2].

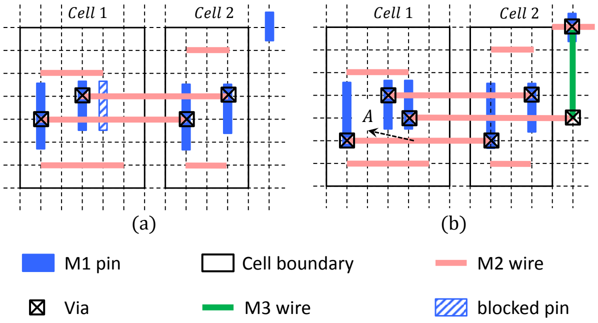

Figure 1: Pin access for detailed routing, (a) pin access failure, (b) pin access success.

Detailed routing aims at pin access and search for exact routes of each net. A typical detailed routing strategy performs pathfinding of the nets sequentially. For SC pin access, the access point selection of the local I/O pins of the net being routed could impact the routability of the remaining nets. A typical example is illustrated in Fig. 1. The pre-routed M2 wires in Fig. 1(a) blocks the I/O pin on the right side of *Cell* 1, which makes the remaining net unroutable. A different access point selection scheme is shown in Fig. 1(b), where the accessing points of net A are changed and all nets are routed. In addition, the routing order of nets is also critical for the routability when each I/O pin has limited number of access points interfering each other [6]. Thus, the pin access planning, including access point selection at the SC level and net order prediction, is very important for the detailed router to achieve better pin accessibility.

A wide range of academic researches across various design stages have been dedicated to the pin access issue in advanced technology nodes, including SC design [1, 2], placement mitigation [7], global routing [8] and detailed routing [6, 9]. Among them, SC I/O pin access and detailed routing play an important role due to their direct impact on the detailed routability. [2] addresses the I/O pin access issue for each cell in isolation under SADP-specific constraints but related detailed routing scheme is not explicitly presented. [6] proposes an escape routing strategy to improve the detailed routability for the dense pin clusters instead of

DAC'15, June 07-11, 2015, San Francisco, CA, USA.
Copyright 2015 ACM 978-1-4503-3520-1/15/06 ...$15.00
http://dx.doi.org/10.1145/2744769.2744890.

each SC within the design. [9] focuses on the gridless pin access in the detailed routing stage, which can not be directly applied in gridded based designs in advanced technology nodes [3].

SADP, in particular, imposes a new set of difficulties on the detailed routing stage. For example, in Fig. 1(b), the M2 extensions are needed to achieve SADP-legal routing results. Several detailed routing algorithms have been presented to deal with the SADP-aware routing [10–14]. All the previous works focus on the 2-D routing compatible with the SADP constraints. While 2-D SADP-friendly patterns are susceptible to overlay [14], regular routing with 1-D layout patterns provides tight control on the overlay of critical dimensions. All side boundaries of target patterns are protected by the spacer and the pattern distortions only occur on the line ends [5]. The overlay of the line ends is defined as *tip overlay* and the overlay of the side boundaries is defined as *side overlay* [14]. We can observe that successful line-space array decomposition induces zero side overlay. To apply the line-space array decomposition for SADP process, the SADP-aware regular routing with 1-D layout patterns becomes a competitive option for the detailed routing on the lower metal layers in future technology nodes. In addition, existing SADP-aware routing simply leaves the duty of pin access to the detailed router [13,14], which is challenging for the ultimate pin accessibility.

In this paper, we propose PARR, a comprehensive framework to explicitly address the SC I/O pin access and regular routing under SADP-specific layout constraints. Our main contributions are summarized as follows:

- The local pin access planning scheme is proposed to enable smart access point selection.
- We propose the global pin access planning strategy based on the concept of pin access graph to guide the regular routing for the ultimate routability.
- The experimental results show that regular routing with the overall pin access planning scheme can achieve zero side overlay and highest routability compared with the state-of-the-art 2-D SADP-aware router [14].
- To the best of our knowledge, this is the first work to systematically address the handshaking between SC level pin access planning and detailed routing stage with the SADP compliance.

The rest of this paper is organized as follows: Section 2 briefly introduces the relevant background and problem definition. Section 3 presents the pin accessibility studies. Section 4 discuss the details on the pin access planning strategies and overall routing flow. Section 5 demonstrates the effectiveness of the PARR framework. Section 6 concludes the paper.

2. PROBLEM FORMULATION

2.1 Related Design Rules

A set of design rules are needed by the detailed router to achieve legal routing results.

Min-area Rule: Min-area is an important rule specifying the minimum area required for the polygons on the routing layer. Due to the fixed width of the regular layout patterns, we convert the min-area rule into the minimum length rule for the metal wires.

Trim Mask Rules: Certain design rules should be assumed for SADP process in order to guarantee feasible line-space array decomposition and detailed discussions can be found in [2,5].

Via Rule: The minimum center-to-center spacing rule of the vias is considered for the completeness of our framework, which can be extended to other complex rules such as multiple patterning related constraints.

2.2 Problem definition

For practical designs, SCs are placed next to each other, which means both intra-cell and inter-cell pin access need to be addressed. Since the number of cells within a library is finite, the intra-cell and inter-cell pin access can be precomputed and stored in a look-up table (LUT). With the LUT for the intra-cell and inter-cell pin access, we define the pin access guided regular routing problem as follows.

Problem 1 (PARR) *Given a netlist, a grid routing plane, cell placement, pin access LUT of the library and a set of design rules, the pin access planning guided regular routing (PARR) is to perform the regular routing and design rule legalization simultaneously to achieve SADP-friendly routing results.*

3. PIN ACCESSIBILITY PREDICTION

3.1 Intra-Cell Pin Access

The SC I/O pin access problem has been explicitly addressed in [2], where the pin accessibility is determined while minimizing the total amount of line-end extensions. However, the detailed router can also perform the line-end extensions and dynamically choose the access direction of a specific accessing point [13,14]. Thus, a feasibility study at the SC level is enough to guide the detailed routing, which means extensions of pin access wires to cell boundary and differentiating accessing directions are not necessary here. We use the adapted pin access and standard cell layout co-optimization (PICO) method to determine the intra-cell pin accessibility. For convenience, we adopt the definitions of *Hit Point (HP)* and *Hit Point Combination (HPC)* from [2], examples of which are shown in Fig. 2(a). An HPC is considered to be a *Valid Hit Point Combination (VHPC)* if the legal pin access wires can be achieved with the adapted pin access optimization (PAO) from [2]. Otherwise, it is considered to be invalid. An HP is defined as *Valid Hit Point (VHP)* if it is accessible within some VHPC. Otherwise, it is considered to be invalid.

The intra-cell pin access computation yields a 2-D set of M2 wires for all VHPCs of each cell within the library, where pa_i^m denotes the m_{th} VHPC for i_{th} cell. In particular, from Fig. 2(a), we can see that the pin access boundary is extended beyond the cell boundary. For practical implementation, the left and right boundary of the cell could be extended by minimum M2 length, which preserves the validness of the HPs close to the cell boundary while satisfying the minLength rule for M2 wires.

3.2 Inter-Cell Pin Access

The pin accessibility may interfere and degrade when two cells are placed next to each other. A typical example of a

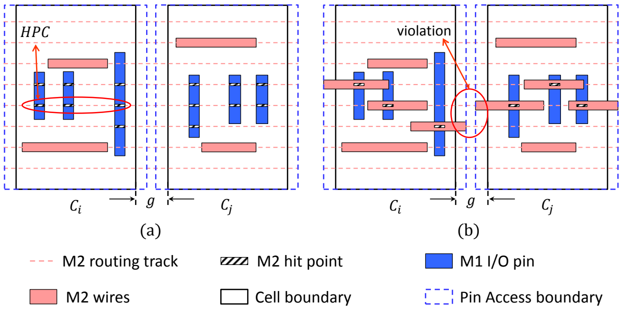

Figure 2: The intra-cell and inter-cell pin access, (a) M2 tracks and cell layout, (b) potential inter-cell pin access conflicts.

cell pair, denoted as $pair_{ij}$, is illustrated in Fig. 2, where c_i is placed to the left of c_j with the gap distance as g. For pragmatic placement, g is an integer multiple of the placement pitch. The pin access interference is expected from Fig. 2(a) because the pin access boundaries of the two cells are next to each other. To demonstrate the pin access interference, we choose m_{th} VHPC for c_i and n_{th} VHPC for c_j in Fig. 2(b). The M2 wires for intra-cell pin access associated with the selected VHPCs are also shown in Fig. 2(b), where pin access M2 wires introduce extra rule violations even with VHPCs for c_i and c_j in isolation. To further explore inter-cell pin accessibility, additional line-end extensions are required to fix the violations and make $pair_{ij}$ accessible in Fig. 2(b).

Actually, the additional line-end extensions based on the selected VHPCs for $pair_{ij}$ have the same formulation as the PAO in [2]. Specifically, if two cells c_i and c_j are assigned m_{th} and n_{th} VHPC and the gap distance is set to g in Fig. 2(b), the feasibility of fixing extra violations can be evaluated with PAO on the set of M2 wires, i.e. $pa_i^m \cup pa_j^n$.

3.3 Look-Up Table Construction

If two cells c_i and c_j are assigned m_{th} and n_{th} VHPC and the gap distance is set to g, the inter-cell pin accessibility can be evaluated on the set of M2 wires, i.e. $pa_i^m \cup pa_j^n$. Then $LUT(i, m, j, n, g)$ stores a boolean value denoting whether inter-cell pin access is feasible or not when c_i is to the left of c_j. Here, the gap distance g is also an index of the LUT because changing the gap distance between cells has potential impact on the inter-cell pin accessibility. For example, the violation in Fig. 2(b) can be fixed by additional line-end extensions. Thus, the item $LUT(i, m, j, n, g)$ within the LUT will store a true value. The cell flipping is also considered and related values are stored during LUT construction. Last but not least, our LUT is constructed only on critical pin-access cells, i.e., cells with small number of HPC (e.g., < 500) or some I/O pin of the cell has very small number of hit points (e.g., < 5). Suppose the number of pin-access critical cells is n, the maximum HPC per cell is m, and the maximum gap is g, then the LUT size is at most $n^2 * m^2 * g$.

4. PIN ACCESS PLANNING GUIDED REGULAR ROUTING

4.1 Single Row Pin Access Graph

In the row-structure for placement in Fig. 3(a), SCs are aligned horizontally and share the same height. The power and ground rails go from the very left to the very right of the die area. Given the position for each cell and a placement row, we build the single row pin access graph. As illustrated in Fig. 3(b), the single row PAG is a directed graph starting from the virtual source node (s) on the left to the virtual target node (t) on the right. For each cell placed within the row, we introduce a set of nodes into the PAG and each node corresponds to one of the VHPCs for that particular cell, where n_i^j denotes the node for the j_{th} VHPC for *Cell* i. We add edges between s and n_0^i, for each i. Similarly, edges will be introduced between n_5^i and t, for each i. No edges will be added between n_i^j and n_i^k, namely, nodes for the same cell. For neighboring cells, such as *Cell* 1 and *Cell* 2, an edge, denoted as blue arrow, will be added between n_1^i and n_2^m since the item $LUT(1, i, 2, m, g)$ is true, where g is the gap distance for $pair_{ij}$. In contrast, no edge is introduced between n_1^i and n_2^k since $LUT(1, i, 2, k, g)$ is false.

For the PAG, we have the following theorem.

Theorem 1 *The pin accessibility of the cells within the single row is equivalent to the existence of a path from s to t of the PAG associated with that particular row.*

PROOF. If there exists a path from s to t, the SCs within the row are accessible using the set of VHPCs on the path. An example is shown in Fig. 3(b). □

Moreover, routing wires on M2 layer will be created on top of the cells during the routing stage, as shown in Fig. 3(c). The routing wires over the cell create blockages, which block some specific HPs of the SCs. This means the associated VHPCs for the cell will also be blocked. As demonstrated in Fig. 3(d), some nodes will become invalid, indicated by the dashed pink nodes.

In addition, we can observe the graph partitioning from Fig. 3(b) to Fig. 3(d) since an edge exists between n_3^i and n_4^j, for any (i, j) pair. Then, pin accessibility of the cells with the row is equivalent to the existence of paths from s to t on two independent components in Fig. 3. However, no feasible path exists on the right component of the PAG after the creation of pre-routed wires in Fig. 3, which means pre-routed wires need to be ripped up to preserve the routability of the remaining nets. To achieve quick update on the PAGs, graph partitioning is applied to all the PAGs associated with the placement rows of the design. Furthermore, each component of the PAG is related to a set of cells in proximity, which is bounded by a determined bounding box. Since the search for impacted components of the PAG needs to be done whenever a net is routed, we adopt R-tree [15] to enable the fast indexing bounding box of each component of the PAG.

4.2 Local Pin Access Planning

Intra-cell pin accessibility study yields a set of VHPCs, denoted as $VHPC_k$ for each cell c_k within the library. A HP is invalid if there is no VHPCs associated with that HP. Therefore, invalid HPs are inaccessible and should be removed before the detailed routing stage, which helps to avoid unnecessary routing efforts.

For each I/O pin for the c_k, we propose a **Dynamic Hit Point Scoring** strategy to differentiate among various VHPs for that particular I/O pin. The basic idea is that, a higher score is assigned to a HP if that particular HP has larger number of VHPCs associated with it than other HPs

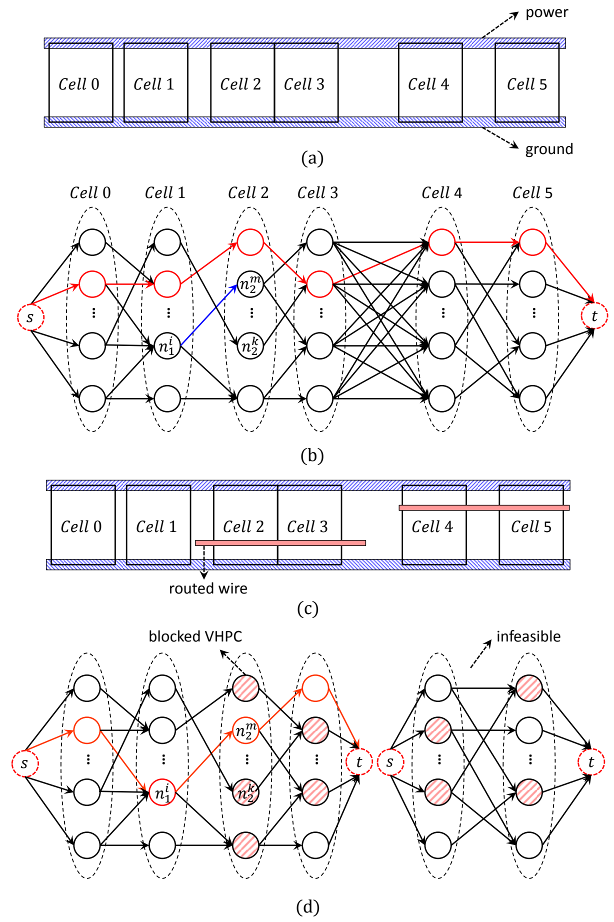

Figure 3: Single row pin access graph, (a) cell placement, (b) initial pin access graph, (c) cell placement with pre-routed wires, (d) simplified pin access graphs with blocked nodes.

of the same I/O pin. Hence, we calculate the score for the j_{th} HP of the i_{th} I/O pin for c_k, namely hp_k^{ij}, as:

$$score(hp_k^{ij}) = \frac{\text{number of VHPCs associated with } hp_k^{ij}}{\text{total number of VHPCs for } c_k} \quad (2)$$

In the sequential routing scheme, the M2 wires created for routed nets become blockages, an example of which is shown in Fig. 3(c). As discussed above, some VHPCs for blocked cells become invalid as illustrated in Fig. 3. Therefore, the score for each HP should be updated dynamically during the detailed routing stage. For the single-net routing, the router prefers selecting the HPs with higher scores for source and target pins of the net being routed.

4.3 Global Pin Access Planning

During sequential detailed routing, the routed wires block some VHPs of the I/O pins not yet routed, which degrades the routability of the remaining nets. Thus, this subsection addresses the global, rather than the local prediction of the pin accessibility.

For sequential detailed routing, the relative order of routing nets has potential impact on the routability as discussed in Section 1. Here, we introduce two techniques to enable **Net Deferring** to improve pin accessibility. First, the

```
Algorithm 1 Net Deferring Algorithm
Require: a set of nets ($Nets$), maximum deferring cost ($maxCost$), increasing unit for deferring cost ($unit$) and pin access graphs for placement rows ($PAGs$);
 1: Define $net\_heap$ as the minimum heap for $Nets$;
 2: Define $nets\_deferred$ as the set of nets with deferring cost exceeding the pre-set bound ($maxCost$);
 3: for each net $n_k$ in $Nets$ do;
 4:     Set $DCost(n_k) = 0$;
 5:     Compute $order(n_k)$ based on equation (3);
 6:     $insert\_heap(n_k, net\_heap)$;
 7: end for
 8: while $net\_heap$ is not empty do;
 9:     Define $net = extract\_min(net\_heap)$;
10:     Perform A* search for $net$;
11:     Update impacted components of $PAGs$;
12:     if $PAGs$ are infeasible for pin access then;
13:         $DCost(net) = DCost(net) + unit$;
14:         if $DCost(net) < maxCost$ then;
15:             Defer $net$ and update $PAGs$;
16:             $insert\_heap(net, net\_heap)$;
17:         else
18:             add $net$ to $nets\_deferred$;
19:         end if
20:     end if
21: end while
22: for each net $n_k$ in $nets\_deferred$ do;
23:     Perfore A* search for $n_k$;
24: end for
```

routability of a net relates to the accessibility of source or target pin. This motivates us to defer the routing of nets with robust source and target pins, which both have many VHPs available. Second, the PAG for each placement row helps determine the accessibility of the cells within the that row. To preserve the global pin accessibility of the remaining nets, we dynamically maintain the source-to-target path existence of each component of the PAGs. Therefore, the weight for the order of the net n_k is calculated as follows.

$$order(n_k) = HPWL(n_k) \cdot (1 + \alpha \cdot \min\{hp_s, hp_t\}) + DCost(n_k) \quad (3)$$

In Eqn. (3), $HPWL(n_k)$ denotes the half-perimeter wirelength of net n_k, α is a user-defined parameter, hp_s and hp_t denote the number of VHPs available for source and target pins, respectively. $DCost(n_k)$ is the deferring cost of n_k. With the net deferring scheme, one net may be deferred for several times due to its impact on the routability of the remaining nets. Then, we have the following definition.

Definition 1 (Deferring Cycle) *Deferring cycle is the maximum number of times that a net is deferred before reaching the cost upper bound.*

The overall net deferring scheme is shown in Algorithm.1. From line 1 to line 6, the minimum heap for routing nets is built based on the order of each net (3). In each loop, the *net* with minimum order is extracted from the *net_heap* and single-net routing is performed in line 8 and line 9. From the routing wires of the *net*, the impacted components of PAGs are updated on line 10. Impacted components are those components containing nodes blocked by the newly created M2 wires as shown in Fig. 3(c)-(d). As discussed in Section

4.1, R-tree enables quick search of the impacted components when new routing wires are created. If the routing results of the net break the pin accessibility of the PAG from line 11 to line 17, the deferring cost is increased and the net is deferred and pushed back to the net_heap when the accumulated deferring cost is within the maximum bound. Otherwise, the net is added to the $nets\_deferred$ in line 18. We perform the routing for the remaining nets in lines 22-24.

4.4 Routing with Design Rule Legalization

Our routing strategy adopts the grid-based routing model and targets at the 1-D layout patterns friendly to the design rules introduced in Section 2.1. The schemes related to design rule legalization are demonstrated in Fig. 4. The line-end extensions are performed to fix the violations relevant to minLength rule and trim mask rules for SADP in Fig. 4(a)-(c). Since both intra-cell and inter-cell pin accessibility are evaluated and pre-computed on M2 layer, all the M1 I/O pins are brought up to the M2 layer. Hence, the Via rule is imposed on the M2 and upper layers. Considering the spacing rule for via layer, we impose the forbidden grids once a legal via is inserted for the routed net. In Fig. 4(d), neighboring grids surrounding the via position are forbidden to be used for the remaining nets.

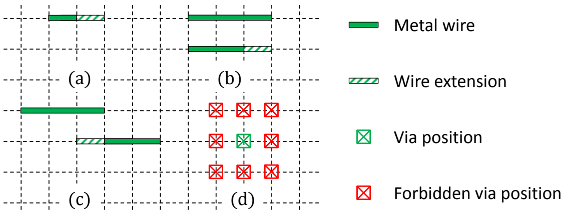

Figure 4: Legalizations, (a) minLength, (b) parallel line ends, (c) anti-parallel line ends, (d) Via rule.

Our detailed router follows the paradigm of A* search, which is guided by the local and global pin access planning strategies. The cost of the routing grid is calculated while performing A* search. If we consider a routing path from grid g_i to grid g_j, the cost of the grid g_j, denoted as $C(g_j)$, can be computed as follows:

$$C(g_j) = C(g_i) + \theta \cdot (1 - score(hp_j)) + \eta \cdot C(forbid(j)) \qquad (4)$$

$$+\beta \cdot C(WL_{ij}) + \gamma \cdot C(Via_{ij})$$

In Eqn. (4), $score(hp_j)$ is the dynamic hit point score for g_j if g_j is a source or target grid. Otherwise, $score(hp_j)$ is set to 1. In general, the A* search prefers routing grids with lower cost. Thus, the term $score(hp_j)$ enables the local pin access planning, which prefers selecting the HPs with higher scores for the source or target pins of the net being routed. $C(forbid(j))$ is the forbidden cost for the grids if the grid g_j is within the prohibited region of some pre-routed wires [5, 13]. This cost is set to help the design rule legalization for the trim mask rules. $C(WL_{ij})$ and $C(Via_{ij})$ are the amount of wirelength and vias for the routing path from g_i to g_j. $\beta, \gamma, \theta, \eta$ are user-defined parameters to adjust weights of the various cost. For each net being routed, A* search is performed and the routing wires are legalized for the given set of design rules. It shall be noted that the net is inserted back to the heap only if the deferring cost is within the preset bound. The overall routing ends with the routing for the remaining deferred nets, as discussed in Algorithm 1.

5. EXPERIMENTAL RESULTS

We have implemented PARR in C++ and all experiments are performed on a Linux machine with 3.4GHz Intel(R) Core and 32GB memory. With the help from the authors of [14], the 2-D SADP-aware routing results are generated on a Linux machine with 2.0GHz CPU and 72GB memory. We modify and scale the NanGate 45nm open cell library [16] to represent the pin access scenario in advanced technology nodes, where M2 wires may be introduced in the SC layout design [2]. For the LUT construction, we only store the false entries in our implementation since each entry is just true or false for the inter-cell pin accessibility checking. The number of entries in the LUT constructed is our implementation is around $3.8 * 10^6$. As illustrated in Table 1, modules from OpenSparc T1 are synthesized with Design Compiler [17]. The placement results are generated using Cadence SOC encounter [18] with utilization rate set to 0.7. All benchmarks are scaled and compacted to $10nm$-representative dimensions. Since our work targets at improving the pin accessibility in the detailed routing stage, we ignore the global nets for each benchmark before the detailed routing. The bounding box of a global net crosses more than M horizontal or vertical routing tracks and $M = 40$ in our implementation. We focus on the two-layer (M2 and M3) regular routing and the routing directions of M2 and M3 are horizontal and vertical, respectively. We adopt Gurobi [19] as our MILP solver. The upper bound on the gap distance of a cell pair is set to $g = 2$. The width and space of Metal-2 and Metal-3 wires are assumed to be $24nm$. The minimum length of the metal wires is set to $48nm$. The minimum center-to-center spacing of the vias is set to $96nm$. SADP related parameters are the same as those in [2, 20]. The user-defined parameters in equation (3) and (4) are set to $\alpha = 0.05, \theta = 4, \eta = 10, \beta = 1, \gamma = 5$.

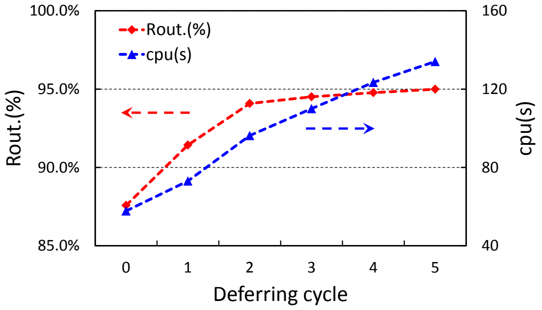

Figure 5: Routability and CPU vs deferring cycle.

The trade-off between routability, run time and deferring cycle is illustrated in Fig. 5 for the benchmark "alu". As the deferring cycle increases, both the routability and run time increase monotonically. The net deferring scheme improves the routability significantly during first few deferring cycles. After that, the run time increases quasi-linearly while the routability improvement degrades. Thus, the $deferring$ $cycle$ is set to 3 for the routability improvement exploration. The increasing $unit$ for deferring cost is set to 200 for better routability.

In Table 2, we compare PARR framework with [14] due to its best efficiency and routability for 2-D SADP-aware detailed routing. We average the number of vias from M3 to M2 over the number of routed nets, namely via number per routed net, which is listed as "V.p.n". The total wirelength,

Table 1: Benchmarks statistics

Ckt	ecc	efc	ctl	alu	div	top
Net#	1671	2219	2706	3108	5813	22201
Size(um^2)	21 x 21	20 x 19	24 x 24	20 x 19	31 x 31	57 x 56

Table 2: Comparison on detailed routability for regular routing with pin access planning

	[14]					Local Pin Access Planning					Local & Global Pin Access Planning				
Ckt	V.p.n	WL*	OLL	Rout.	CPU(s)	V.p.n	WL*	OLL	Rout.	CPU(s)	V.p.n	WL*	OLL	Rout.	CPU(s)
ecc	2.31	41497	2775	91.14%	16.77	2.51	45102	0	91.20%	14.21	2.66	46588	0	96.41%	19.98
efc	2.31	54459	4703	82.47%	100.5	2.25	56457	0	88.06%	22.86	2.40	57834	0	94.91%	34.52
ctl	2.24	67470	5255	87.25%	93.80	2.26	71643	0	88.29%	22.39	2.42	72388	0	95.27%	37.14
alu	2.26	68491	5713	79.44%	143.4	2.22	73430	0	87.48%	28.32	2.44	75679	0	95.17%	45.92
div	2.29	139309	11267	79.40%	253.5	2.34	150356	0	87.58%	57.79	2.51	155704	0	94.60%	106.0
top	N/A	N/A	N/A	N/A	N/A	2.32	496228	0	88.15%	253.7	2.47	513366	0	95.33%	763.2
Avg.	0.919	0.913		0.881	2.358	0.933	0.973		0.928	0.578	1.000	1.000		1.000	1.000

listed as "WL*", is the summation of the half perimeter wirelength for unrouted nets and actual wirelength for routed nets in terms of routing grid count. "OLL" denotes the total side overlay length [14]. "Rout." denotes the percentage of routed nets and "CPU" denotes the run time in seconds. Table 2 demonstrates the strength of the local and global pin access planning for the regular routing over the 2-D detailed routing from [14]. For the local pin access planning, the net order is computed with Eqn. (3) with the term $DCost(n_k)$ ignored. The zero side overlay originates from the line-space array decomposition method. Moreover, our router achieves faster run time by avoiding the extra efforts to maintain the overlay constraint graph from [14]. The overall pin access planning strategy achieves over 95% routability on average for all benchmarks, which is 7.8% increase from the local pin access planning scheme and over 10% increase from [14]. Meanwhile, we observe 6.7% increase in "V.p.n" and 2.7% increase in "WL*" from the local pin access planning strategy, which is treated as a reasonable trade-off for the routability improvement. It shall be noted that the "WL*" and "V.p.n" for [14] and local pin access planning scheme may also increase if similar routability is achieved. The run time overhead comes from the pin accessibility checking and net deferring, which is still two times faster compared with [14].

6. CONCLUSION

In this paper, we propose a comprehensive framework, including pin access LUT construction for a given library, local and global pin accesss planning to improve the pin accessibility during the SADP-aware regular routing. To the best of our knowledge, this is the first work to systematically enable the handshaking between standard cell level pin access and detailed routing stage. Compared to the 2-D SADP-aware detailed router, our approach can achieve significant improvement in terms of the overlay and routability.

7. ACKNOWLEDGMENT

This work is supported in part by NSF and SRC.

8. REFERENCES

[1] M.-K. Hsu, N. Katta, H. Y.-H. Lin, K. T.-H. Lin, K. H. Tam, and K. C.-H. Wang, "Design and manufacturing process co-optimization in nano-technology," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 574–581.

[2] X. Xu, B. Cline, G. Yeric, B. Yu, and D. Z. Pan, "Self-aligned double patterning aware pin access and standard cell layout co-optimization," in *ACM International Symposium on Physical Design (ISPD)*, 2014, pp. 101–108.

[3] M. C. Smayling, K. Tsujita, H. Yaegashi, V. Axelrad, T. Arai, K. Oyama, and A. Hara, "Sub-12nm optical lithography with 4x pitch division and SMO-lite," in *Proc. of SPIE*, 2013, pp. 868 305–868 305.

[4] Y. Zhang and C. Chu, "Regularroute: An efficient detailed router with regular routing patterns," in *ACM International Symposium on Physical Design (ISPD)*, 2011, pp. 45–52.

[5] G. Luk-Pat, B. Painter, A. Miloslavsky, P. De Bisschop, A. Beacham, and K. Lucas, "Avoiding wafer-print artifacts in spacer is dielectric (SID) patterning," in *Proc. of SPIE*, 2013, pp. 868 312–868 312.

[6] M. M. Ozdal, "Detailed-routing algorithms for dense pin clusters in integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 28, no. 3, pp. 340–349, 2009.

[7] T. Taghavi, C. Alpert, A. Huber, Z. Li, G.-J. Nam, and S. Ramji, "New placement prediction and mitigation techniques for local routing congestion," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2010, pp. 621–624.

[8] C. Alpert, Z. Li, C. Sze, and Y. Wei, "Consideration of local routing and pin access during vlsi global routing," Apr. 4 2013, uS Patent App. 13/252,067. [Online]. Available: http://www.google.com/patents/US20130086544

[9] T. Nieberg, "Gridless pin access in detailed routing," in *ACM/IEEE Design Automation Conference (DAC)*, 2011, pp. 170–175.

[10] M. Mirsaeedi, J. A. Torres, and M. Anis, "Self-aligned double-patterning (SADP) friendly detailed routing," in *Proc. of SPIE*, vol. 7974, 2011.

[11] J.-R. Gao and D. Z. Pan, "Flexible Self-aligned Double Patterning Aware Detailed Routing with Prescribed Layout Planning," in *ACM International Symposium on Physical Design (ISPD)*, 2012.

[12] C. Kodama, H. Ichikawa, K. Nakayama, T. Kotani, S. Nojima, S. Mimotogi, S. Miyamoto, and A. Takahashi, "Self-Aligned Double and Quadruple Patterning-Aware Grid Routing with Hotspots Control," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2013.

[13] Y. Du, Q. Ma, H. Song, J. Shiely, G. Luk-Pat, A. Miloslavsky, and M. D. Wong, "Spacer-is-dielectric-compliant detailed routing for self-aligned double patterning lithography," in *ACM/IEEE Design Automation Conference (DAC)*, 2013, p. 93.

[14] I.-J. Liu, S.-Y. Fang, and Y.-W. Chang, "Overlay-aware detailed routing for self-aligned double patterning lithography using the cut process," in *ACM/IEEE Design Automation Conference (DAC)*, 2014, pp. 1–6.

[15] A. Guttman, *R-trees: a dynamic index structure for spatial searching.* ACM, 1984, vol. 14, no. 2.

[16] "NanGate FreePDK45 Generic Open Cell Library," http://www.si2.org/openeda.si2.org/projects/nangatelib.

[17] "Synopsys Design Compiler," http://www.synopsys.com.

[18] "Cadence SOC Encounter," http://www.cadence.com/.

[19] "GUROBI," http://www.gurobi.com/html/academic.html.

[20] G. Luk-Pat, A. Miloslavsky, B. Painter, L. Lin, P. De Bisschop, and K. Lucas, "Design compliance for spacer is dielectric (SID) patterning," in *Proc. of SPIE*, 2012, pp. 83 260D–83 260D.