Closing the Gap between Global and Detailed Placement: Techniques for Improving Routability *

Chun-Kai Wang[1], Chuan-Chia Huang[2], Sean Shih-Ying Liu[1], Ching-Yu Chin[1],
Sheng-Te Hu[1], Wei-Chen Wu[1], Hung-Ming Chen[1]
[1] Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan
[2] NovaTek Inc., Hsinchu, Taiwan
{oldkai.ee90, chuan.ee96, sniealu, alwaysrain.gr, jim221122,
sherlock21315}@gmail.com, hmchen@mail.nctu.edu.tw

ABSTRACT

Improving routability during both global and detailed routing stage has become a critical problem in modern VLSI design. In this work, we propose a placement framework that offers a complete coverage solution in considering both global and detailed routing congestion. A placement migration strategy is proposed, which improves detailed routing congestion while preserving the placement integrity that is optimized for global routability. Using the benchmarks released from ISPD2014 Contest, practical design rules in advanced node design are considered in our placement framework. Evaluation on routability of our placement framework is conducted using commercial router provided by the 2014 ISPD Contest organizers. Experimental results show that the proposed methodologies can effectively improve placement solutions for both global and detailed router.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids—*Placement and routing*; J.6 [**COMPUTER-AIDED ENGINEERING**]: Computer-aided design (CAD)

Keywords

routability, placement migration

1. INTRODUCTION

Routability of the design is a critical factor to achieve design closure using minimal area. If the design is unroutable, designer is forced to increase chip area to obtain a routable design. Increase in chip area indicates increase in cost. Thus, improving routability is critical to achieve cost effective design.

While global routing congestion captures the general routability of the design, routing violations induced by non-default routes and routing blocked pins are unable to identify by global router. Previous works have shown that detailed routing congestion in advanced node design becomes more critical and can be inconsistent with congestion during global routing stage [18]. Additional effort is required during placement stage to consider the detailed routability and design rule constraints.

Prior arts proposed several methods on resolving global routing congestion. The work done in [2, 3, 6, 8, 15] resolves congestion by adjusting cell density through cell inflation. The work done in [6, 8] dynamically controls the target cell density during global placement iteration. In [3], a net based removal algorithm is proposed to reduce congestion. The work done in [4, 5] directly models routing overflow penalty in the nonlinear optimization. In [20], the placer adopted from the SimPL framework [9] expands look ahead region with precise amount of white space to meet routing demand. Based on empirical results from prior arts, cell inflation is effective in reducing local routing congestion and regional density control is effective in reducing global routing congestions.

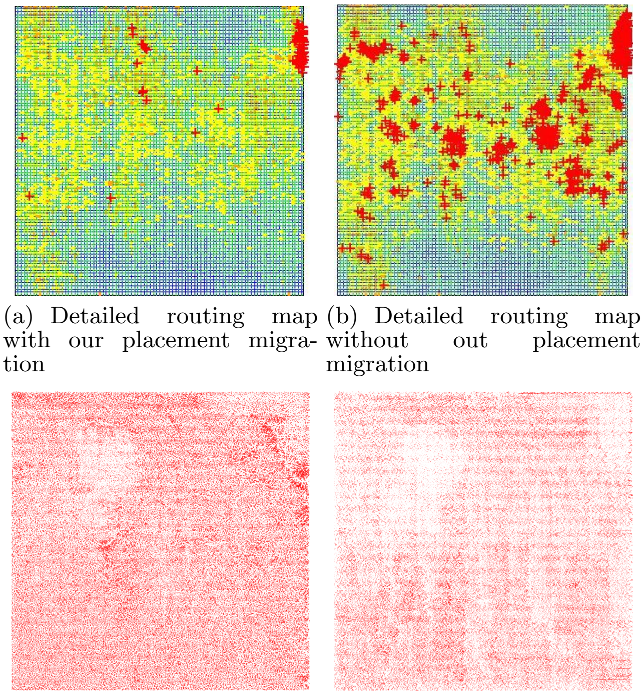
Several recent works [1, 7, 12] addressed their effort on congestion and design rule violations occurred during detailed routing stage. Conventional congestion-aware placer is unaware of the detailed routing congestion. In advanced node technology, design rule violations often occur around fixed macro blocks and between certain types of adjacent cells.

In terms of modeling routing congestion, recent work done in [6, 12] has proposed several methods to account local net congestion by adjusting global routing edge capacity. In 2014 ISPD Detailed Routing Driven Placement Contest [19], more realistic constraints from advanced node design are included in the contest, which include non-default routing (NDR), predefined power ground stripes, and edge type spacing rules. This requires delicate routing resource modeling to capture routing congestion at detailed routing stage.

In this work, we address global routing congestion using several global routing congestion optimization techniques including history based cell inflation and dynamic target density control in our global placer adopted from the SimPL framework [9]. To optimize placement for better detailed routability and to maintain the integrity of the original placement, we implement an incremental placer adopted from the

[1]This work was supported by the Ministry of Science and Technology of Taiwan by Grant No. MOST 103-2220-E-009-028.

ISPD'15, March 29–April 1, 2015, Monterey, CA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3399-3/15/04 . . . $15.00.
http://DOI .



(a) Detailed routing map with our placement migration

(b) Detailed routing map without out placement migration

(c) Cell displacement map with our placement migration

(d) Cell displacement map without our placement migration

Figure 1: The short number of detail routing is reduced after placement migration. (a) and (b) shows detail routing result; (c) and (d) shows cell movement from the end of global placement to the end of detail placement.

Kraftwerk2 framework [17]. Finally, a DRC-driven detailed placement is performed after incremental placement.

The purpose of our incremental placer follows the concept of placement migration proposed in [13, 14]. In [14], it is shown that placement migration can improve timing and total wirelegnth compared to conventional legalization. The concept of placement migration can also be observed in MAPLE [10] which applies progressive local refinement to mitigate local cell overlapping. This work on applying incremental placer demonstrates similar benefit compared to placement migration. However, the work done in [14] moves cells using diffusion-based method which does not include a wirelength model. On the contrary, our incremental placer is a three force balance system which include a hold force to maintain the integrity of original placement, a net force that models the wirelength and a move force to perturb the placement.

Fig. 1 shows the benefit on applying our incremental placer for placement migration. Fig. 1(a) is the detailed routing congestion map with our placement migration and Fig. 1(b) is the detailed routing congestion map without our placement migration. Regions colored in red indicate severe detailed routing violation, regions colored in yellow indicate moderate detailed routing violation and regions colored in blue indicate no detailed routing violation. It can be observed regions colored in red is greatly reduced with our placement migration.

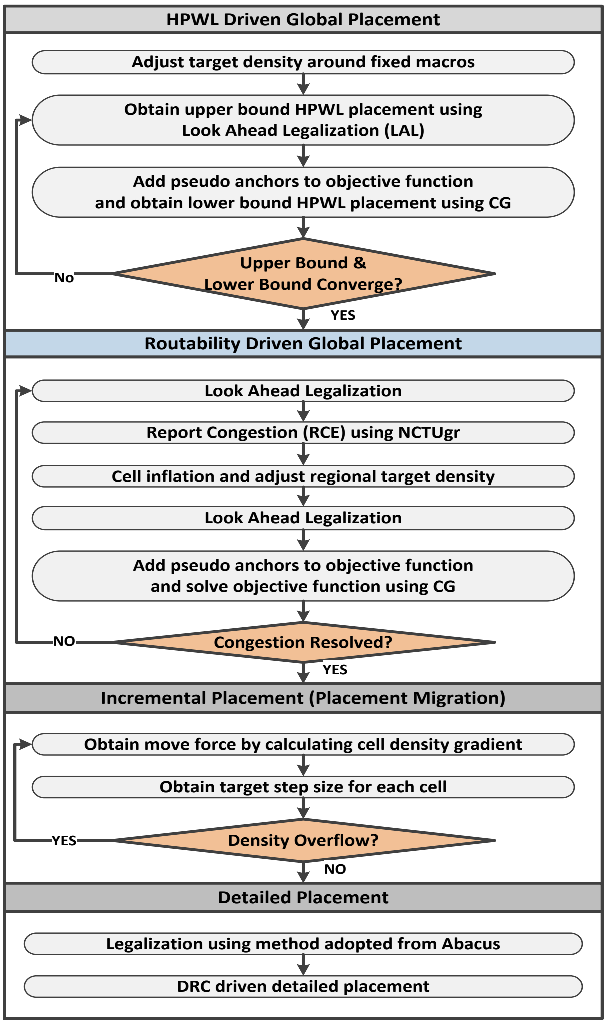

HPWL Driven Global Placement
Adjust target density around fixed macros
Obtain upper bound HPWL placement using Look Ahead Legalization (LAL)
Add pseudo anchors to objective function and obtain lower bound HPWL placement using CG
Upper Bound & Lower Bound Converge?
No
YES
Routability Driven Global Placement
Look Ahead Legalization
Report Congestion (RCE) using NCTUgr
Cell inflation and adjust regional target density
Look Ahead Legalization
Add pseudo anchors to objective function and solve objective function using CG
Congestion Resolved?
NO
YES
Incremental Placement (Placement Migration)
Obtain move force by calculating cell density gradient
Obtain target step size for each cell
Density Overflow?
YES
NO
Detailed Placement
Legalization using method adopted from Abacus
DRC driven detailed placement

Figure 2: Flow chart of the proposed placement framework

Fig. 1(c) is the cell displacement map with our placement migration and Fig. 1(d) is the cell displacement map without our placement migration. Fig. 1(d) finds nearest legalization spot for each cell. Thus, it generates less cell displacement. Although the placement migration generates larger cell displacement, it effectively reduces the disruption of routability induced by legalization.

Fig. 2 is the flow chart of the proposed placement framework. The placement framework consists of four stages. The first stage is a HPWL driven global placer. The second stage optimizes routability of the placed design using NCTUgr [11] for routing congestion estimation. The third stage performs placement migration to reduce the disruption during legalization and maintain the integrity from first stage global placement. The forth stage is a DRC-driven detailed placer that serves to reduce edge type spacing violations.

The contributions of this work are summarized as follows.

- An incremental placer for placement migration is proposed to improve cell relative order and maintain integrity of the original placement. Experimental result shows that with placement migration, detailed routing congestion can be greatly reduced.

- A design rule constraint (DRC) driven detailed placer is proposed to reduce the edge type minimum spacing violation for specific types of cell.
- Routing resource model for non-default routing and the power ground stripes.

The rest of this paper is organized as follows. Section 2 formulates the problem. Section 3 presents our routability driven global placer. Section 4 presents our incremental placer for placement migration. Section 5 presents our DRC-driven detailed placer. Section 6 shows the experimental result of our placement framework on 2014 ISPD Contest Benchmark [19] and Section 7 concludes this work.

2. PROBLEM FORMULATION

Given a netlist $N = (V, E)$ in which V is a set of modules and E is a set of connections of modules. The problem is to place all movable modules on placement site with the following objectives: 1) minimize total detailed routing violations, 2) minimize routed wire length, and 3) minimize total run time.

In 2014 ISPD Contest Placement benchmarks, a given design includes the following files.

- **A technology LEF** file that defines the placement site, available metal layers and routing rules. The routing rules include metal width, metal spacing, end of line rule, and non-default-routing (NDR).
- **A cell LEF** file that defines physical information of modules.
- **A floorplan DEF** file that defines the routing resource occupied power ground rails, fixed I/O ports, fixed modules, and unplaced modules.
- **A Verilog** file that defines the netlist in gate level.

3. ROUTABILITY DRIVEN GLOBAL PLACEMENT

Before global placer begins, the target density adjacent to fixed macros is scaled down using Gaussian Blurring. This prevents cell sliding to thin channels between fixed macro blocks. After target density around fixed macros is configured, a HPWL driven global placer is performed until the upper bound and lower bound of HPWL is converged. Routability optimization begins after HPWL driven global placement. Our routability optimization techniques consist of cell inflation which serves to reduce local routing congestion and regional target density control which serves to reduce global routing congestion.

During global routing stage, the design is partitioned into a 2D global routing tile array (g-cell). Each g-cell has a routing capacity in horizontal and vertical direction. Routing overflow occurs if routing demand of a g-cell exceeds its routing capacity.

Pin density of g-cell is monitored to determine on which circumstance to apply cell inflation and target density control. Congested g-cell with low pin density is considered as global congestion, and target density is adjusted for these g-cells. Algorithm 1 describes our procedure on inflating cells and adjusting target density. Table 1 defines the annotations used in Algorithm 1.

Table 1: Annotations used Algorithm 1

Variable	Definition
g_k	g-cell k
v_i	cell i
Dem(g_k)	Routing Demand of g_k
Cap(g_k)	Routing Capacity of g_k
width(v_i)	Current cell width of v_i
orig_width(v_i)	Original cell width of v_i
pin(g_k)	Pin density in g_k
H-Cap(g_k)	Horizontal routing capacity of g_k
V-Cap(g_k)	Vertical routing capacity of g_k
current_dens(g_k)	Current cell density in g_k
target_dens(g_k)	Target cell density in g_k

Algorithm 1 Cell Inflation and Target Density Control

```
for each g-cell $g_k$ do
  if Dem($g_k$)/Cap($g_k$) $>$ 90% then
    for each cell $v_i$ in $g_k$ do
      width($v_i$) = orig_width($v_i$) * (1 + 0.1$\beta$)
    end for
    if pin($g_k$)<0.3*(H-Cap($g_k$)+V-Cap($g_k$)) then
      target_dens($g_k$) = current_dens($g_k$)*0.9
    end if
  else
    target_dens($g_k$) += (1.0 - current_dens($g_k$)) * 0.6
  end if
end for
```

In Algorithm 1, a g-cell is defined as congested if routing demand of a g-cell exceeds 90% of its routing capacity. Each cell in a congested g-cell is inflated by 10% of its original width times β that is the number of times this cell has been inflated. A g-cell is defined as a globally congested g-cell if it is congested and its pin density is less than 30% g-cell capacity. The target densities for these global congested g-cell is scaled down by 0.9. For non-congested g-cell, the target density is increased by scaled up by 0.6 on its current white space.

Look ahead legalization is performed after cell inflation and target density in selected regions is adjusted. Routability of the placement is evaluated using the ACE metric [18] (weighted sum of top 0.5%, 1%, 2%, and 5% congested g-edges) and total routed wire length $rtWL$. When Eq. (1) is satisfied, the placement location will be updated.

$$ACE_{best} - ACE_{new} > \alpha \times (\frac{rtWL_{new}}{rtWL_{best}} - 1)) \qquad (1)$$

Initially, the ACE_{best} and $rtWL_{best}$ are assigned by routing congestion estimation of first iteration. When the condition described in Eq. (1) is satisfied, ACE_{best} and $rtWL_{best}$ are updated with the new placement result. In this work, α is set to 10.

NCTUgr [11] is applied to estimate routing congestion. NCTUgr provides APIs to define size of g-cell, pin blockage and routing mode. In this work, g-cell size is set to twice of the row height, pin blockage is set according to the number of available routing metal layers.

To account for NDR nets, the routing congestion estimation has two steps. In the first step, the routing resource is scaled down by the spacing and width of NDR and only

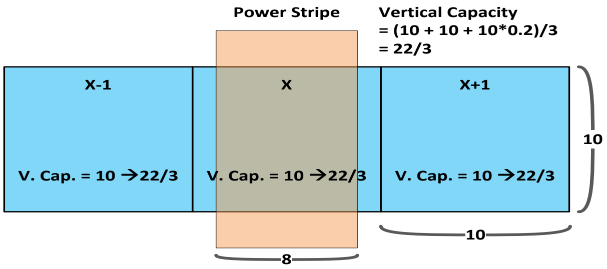


Figure 3: Routing resource model with power stripes

nets belong to NDR are routed. In the second step, the remaining nets are routed using the routing resource that is deducted by the NDR routing resource. In addition, to account for the NDR-pin-access, the cell with NDR pin are inflated by one site so that the spacing is preserved.

Predefined nets should be considered for routing resource during global placement. **However, the false alarm of routing congestion estimation would easily be generated if the following two conditions occurs: if 1) g-edge capacity is directly deducted by the overlap ratio between g-edge and predefined power ground stripes and 2) the g-cell size is equal to the power ground stripe width. Since any wire passing through these g-cells reports congestion even if it actually satisfies routing capacity constraint.** Our solution is to evenly deduct routing capacity from n g-cells adjacent to the g-cell overlapped with the power stripe. Fig. 3 illustrates an example of adjust routing capacity with power stripes. In Fig. 3, the original vertical capacity is set to 10 and window size is set to 3. 80% of the vertical routing capacity of the g-cell at the center is occupied by one power stripe. The adjusted vertical capacity takes total routing capacity to subtract occupied capacity by the power stripe and divide to three other g-cells.

4. PLACEMENT MIGRATION

The technique of placement migration is applied to migrate placement solution from global placement to legalization. In this work, we found that detailed routing congestion is improved when placement migration is applied. The Kraftwerk2 placement framework proposed in [17] is adopted as the incremental placer used in this work. The main difference between Kraftwerk2 and this work is how to derive the move force, which effectively smooths the transition from global placement to legalization.

Kraftwerk2's force system includes three forces: net force, hold force, and move force. The net force models the net connection and can be minimized using iterative method. Adding hold force to the force model cancels out the effect of net force so that cells remain in same location. Adding move force to the force model, the force equilibrium model moves cells to new position and placement is incrementally spread out. The force equilibrium system consisting of net force, hold force, and move force preserves the integrity of the original placement and incrementally adjust placement to migrate cell density.

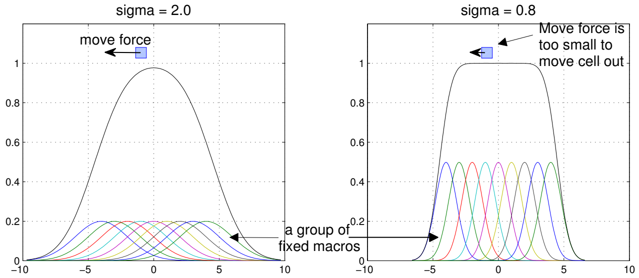
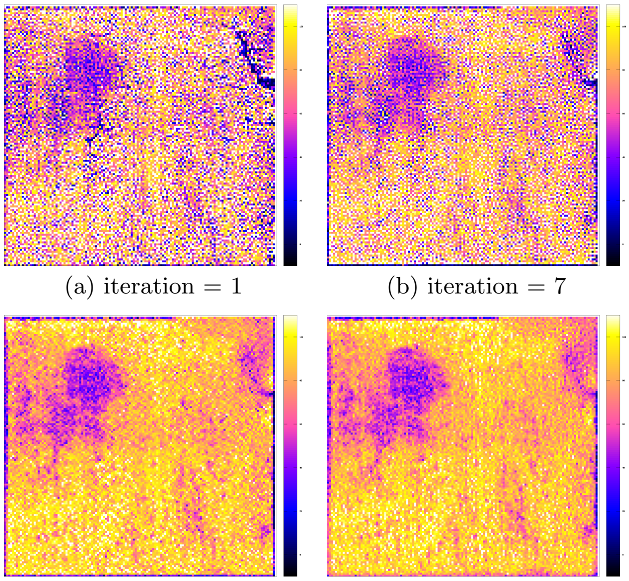


Figure 4: Gaussian superposition in response to different value of *sigma*.


(a) iteration = 1

(b) iteration = 7

(c) iteration = 14

(d) iteration = 20

Figure 5: Cell density of testcase mgc_matrix_multiply during incremental placement. Cell density is gradually smooth during placement migration.

In Kraftwerk2, each move force is modeled by an anchor $\mathring{x}$ as a pseudo net and a pseudo net weight $\mathring{w}$. The anchor position is derived by solving the Poisson equation describe in Eq. (2) in which cells are regarded as charges and the electric field implies the direction of cell spreading. However, since the primary objective is not the accuracy of the potential surface, any distribution function is suffice to meet the supply-demand constraint. In this work, the surface model is obtained using Gaussian Blurring described in Eq. (3)(4).

$$\mathring{x}(x,y) = -\nabla\Phi(x,y) \tag{2}$$

$$\begin{aligned}\Phi(x,y) = &\sum_{i\in standard} h(x,y) * g(x,y,\sigma_s) \\ &+ \sum_{i\in fixed} h(x,y) * g(x,y,\sigma_f)\end{aligned} \tag{3}$$

$$g(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\left(\frac{x^2}{2\sigma^2}+\frac{y^2}{2\sigma^2}\right)\right) \tag{4}$$

In terms of image processing, Gaussian blurring is equivalent to the convolution between a Gaussian function $g(x,y,\sigma)$ and a density function $h(x,y)$. The term σ defines the affected range for each unit cell area. In Eq. (3), the fixed macros and the standard cells are separated, and the different σ is applied to remove the overlap that standard cells are illegally placed on fixed macros. To remove such overlap, σ_f is larger than σ_s, or only the overlap near macros boundary will be removed. Fig. 4 illustrates difference in cell density surface model between a large σ_f value and a low σ_f values. After the overlap on fixed macros are removed, σ_f are reduced to be equal to σ_s so that the white space near macro boundary could be utilized.

In our implementation, $h[m,n]$ and $g[i,j]$ are the discrete representation for $h(x,y)$ and $g(x,y)$. The plane is discretized by slicing chip dimension into small bins and calculating the area overlap between bin and module and each unit cell area has an amplitude of 1 unit Gaussian distribution. In this work, the bin size is equal to the row height.

$$h[m,n] = \sum_{v \in V} P_v[m,n] \tag{5}$$

$$\Phi_x[m,n] = \sum_i \sum_j h[m-i,n-j] \frac{\partial}{\partial \boldsymbol{x}} g[i,j] \tag{6}$$

$$\Phi_y[m,n] = \sum_i \sum_j h[m-i,n-j] \frac{\partial}{\partial \boldsymbol{y}} g[i,j], \tag{7}$$

$P_v[m,n]$ is the discrete density function contributed by cell v. To derive Φ_x and Φ_y, the convolution between density matrix and Gaussian matrix is performed, and then the anchor of cell located at (x',y') is interpolated by Φ_x and Φ_y. Note that no target density is set in this stage, since the derived anchors are located within local bins, the global density were not changed in our experiment.

The execution time of two dimensional convolution in spatial domain increases quadratically with the size of Gaussian matrix g. The computation complexity is reduced, when convolution is performed in frequency domain. Suppose a $M \times M$ Gaussian matrix and a $N \times N$ density matrix, the complexity is reduced from $O(M^2N^2)$ to $O(N^2 logN^2)$ in frequency domain.

In Kraftwerk2, the pseudo net weight is iteratively adjusted using Eq. (8) in which μ_T is the target step size to control the trade-off between convergence rate and placement quality. However, the target step size is the average movement of all cells that lacks of local view and does not effectively remove local overlaps.

$$\mathring{w}_i^{k+1} = \mathring{w}_i^k \cdot (1 + \tanh(\ln(\mu_T/\mu))) \tag{8}$$

Thus, Eq. (9) is applied to configure the pseudo net weight $\mathring{w}_i^k$ in this work. To prevent large perturbation, the pseudo net weight is set to half of the Bound2Bound (B2B) wirelength model. Compared to the pseudo net weight in Eq. (10) which is applied in the SimPL framework, the net weight is equivalent to the 50th global placement iteration.

$$\mathring{w} = 0.5 \times \frac{1}{|x' - \mathring{x}|} \tag{9}$$

$$\mathring{w}^{k+1} = \frac{0.01 \cdot (1 + \text{iterationNumber})}{|x' - \mathring{x}|}, \tag{10}$$

The incremental placer iteratively spreads out placement density until the maximum iteration is reached. As the iteration number increases, σ_s is gradually decreased so that cells are separated from each other. Larger value of σ_s implies that each cell affects larger region of placement which can drive small cell out of fixed macro. A smaller value of σ_s implies that each cell affects smaller region of placement which let cell rearrange its relative order and find nearest white space.

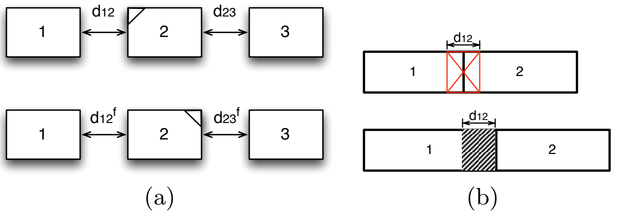


Figure 6: Cells are flipped if the spacing between adjacent cells is reduced after flipping. (a) Flipping cell 2. (b) Inflating cell 1 to resolve spacing rule violation.

In this work, the maximum incremental placement iteration is set to 20 and the value of σ_s is halved in every 7 iterations. The initial value of σ_s is set to four times of row height. Figure 5 shows the change in cell density for testcase mgc_matrix_multiply during incremental placement. It can be observed that cell density is gradually decreased on each iteration of incremental placement.

5. DRC DRIVEN DETAILED PLACEMENT

After placement migration, cells are legalized using technique adopted from Abacus [16]. A DRC driven detailed placer is initiated to reduce the edge type spacing violations. Cells are examined in pairs on each row from left to right. There are two rounds of examination. In the first round of examination, if a pair of cells violates edge type spacing rule, the width of the left cell is inflated to the minimum width such that spacing rule is satisfied. In the second round of examination, cell is flipped and checked whether the spacing between adjacent cells can be reduced. After examining all adjacent pairs, cells are legalized again. Figure 6 illustrates the cell flipping and cell inflation to resolve edge type spacing rule. In Figure 6(a), cell 2 is flipped if $d_{12} + d_{23} > d_{12}^f + d_{23}^f$. In Figure 6(b), cell 1 and cell 2 have an edge type spacing violation, the width of cell 1 is inflated with the required amount of spacing. Note that the pin blockage due to power stripes was not be dealt in this work.

6. EXPERIMENTAL RESULTS

In this section, experimental results of our implementations are presented. All of our implementations are implemented using standard C++ language and performed on an Intel Xeon E5620 machine running at 2.4G Hz. Table 2 lists the statistics of the benchmarks used in 2014 ISPD Contest.

The parameters used in the HPWL driven placement are described as follows. For the Gaussian Blurring fixed macros before global placement, the grid size is four times of the row height, and the σ term of Gaussian function is 8. The look ahead legalization has the finest bin area being 64 row height square and the bin aspect-ratio being the average cell width to the row height. Table 3 lists the density limit (d.l.) for look ahead legalization in HPWL driven placement

Table 4: Comparison on placement result of our placement framework with top 3 teams in 2014 ISPD Placement Contest. (Normalized value excludes the unaccepted results)

Benchmark	UW&UC	CUHK	NCTU	[7]	Our	
DP						**run time (sec)**
mgc_des_perf_1	3.05	3.43	4.08	2.12	4.26	233.99
mgc_des_perf_2	2.85	2.27	1.75	1.55	2.47	228.72
mgc_edit_dist_1	0.99	0.91	0.84	0.03	1.14	363.93
mgc_edit_dist_2	1.06	1.15	1.48	0.03	1.15	365.31
mgc_pci_bridge_1	1.46	1.58	1.51	0.26	1.45	60.41
mgc_pci_bridge_2	1.44	1.62	1.44	0.44	1.62	60.73
mgc_fft	2.04	2.09	2.48	0.96	2.88	72.21
mgc_matrix_mult	1.27	1.22	1.30	0.32	1.43	345.09
mgc_superblue_11	0.55	0.00	0.01	-	0.00	4217.31
mgc_superblue_12	-	-	-	-	0.00	7638.05
mgc_superblue_16	0.04	0.00	0.00	-	0.00	2821.62
Norm.	0.90	0.87	0.91	0.35	1.00	
DR						
mgc_des_perf_1	998.7	421.0	1411.9	296.0	1536.75	
mgc_des_perf_2	19.1	8.5	507.2	3.8	3.75	
mgc_edit_dist_1	3.0	265.5	0.0	23.9	0.0	
mgc_edit_dist_2	120.3	151.0	50.3	209.8	0.0	
mgc_pci_bridge_1	0.0	0.0	0.0	3.9	0.0	
mgc_pci_bridge_2	0.0	0.3	1.5	0.7	0.5	
mgc_fft	105.2	266.6	228.5	331.7	165.2	
mgc_matrix_mult	158.1	1033.4	311.4	182.1	30.95	
mgc_superblue_11	349.8	99.6	349.8	-	232.7	
mgc_superblue_12	-	-	-	-	178.05	
mgc_superblue_16	449.1	21.0	298.4	-	122.25	
Norm.	1.05	1.08	1.51	0.53	1.00	
rtWL (m)						
mgc_des_perf_1	1.838	1.995	2.026	1.820	2.016	
mgc_des_perf_2	1.920	1.972	1.893	1.863	1.952	
mgc_edit_dist_1	4.977	5.417	na	4.854	4.968	
mgc_edit_dist_2	4.695	5.138	5.048	4.858	4.876	
mgc_pci_bridge_1	0.3653	0.3602	0.3688	0.361	0.3709	
mgc_pci_bridge_2	0.3719	0.3704	0.3660	0.352	0.3718	
mgc_fft	0.6516	0.6796	0.6600	0.646	0.6734	
mgc_matrix_mult	3.085	3.133	3.085	3.045	3.073	
mgc_superblue_11	44.54	44.09	45.24	-	44.17	
mgc_superblue_12	-	-	-	-	61.59	
mgc_superblue_16	35.29	38.25	35.89	-	32.88	
Norm.	1.02	1.06	1.10	0.97	1.00	

Table 2: Statistics on ISPD 2014 Detailed Routing-Driven Placement Contest Benchmarks

Benchmarks	Util.	#Cell	#nets	#I/O
Suite A				
mgc_des_perf_1	90%	112644	112878	374
mgc_des_perf_2	85%	112644	112878	374
mgc_edit_dist_1	40%	130661	133223	2574
mgc_edit_dist_2	43%	130661	133223	2574
mgc_pci_bridge_1	84%	30675	30835	361
mgc_pci_bridge_2	85%	30675	30835	361
mgc_fft	83%	32281	33307	3010
mgc_matrix_mult	80%	155325	158527	4802
Suite B				
mgc_superblue_11	44%	925616	935731	27371
mgc_superblue_12	48%	1286948	1293436	5908
mgc_superblue_16	49%	680450	697458	17498

Table 3: Parameters in the implemented global placement

Benchmarks	p.f.	d.l.	NDR Util.
mgc_des_perf_1	0.15	96%	95%
mgc_des_perf_2	0.15	91%	89%
mgc_edit_dist_1	0.15	51%	42%
mgc_edit_dist_2	0.175	55%	46%
mgc_pci_bridge_1	0.15	96%	95%
mgc_pci_bridge_2	0.15	98%	97%
mgc_fft	0.15	92%	91%
mgc_matrix_mult	0.175	87%	84%
mgc_superblue_11	0.15	70%	53%
mgc_superblue_12	0.15	62%	47%
mgc_superblue_16	0.15	70%	57%

and the pin factor (p.f.) for routing congestion estimation. Note that the density limit considers the inflated cells due to NDR-pin-access, and the utilization after inflating cells due to NDR-pin-access is listed in the last column. The HPWL driven placement is terminated after 45th iteration.

Table 4 compares our results with the top three teams in 2014 ISPD Contest and [7]. Score for each team is collected from the contest website. Every placement result of our placement framework is evaluated by Mentor Graphics OLYMPUS-SOC placement and route tool by submitting the result to the contest server. The contest evaluates placement quality using three metrics. The first metric is the detailed placement (DP) score, which evaluates the displacement of cell from its original input location to its final location, determined by the detailed placer of OLYMPUS-SOC. The second metric is the detailed routing (DR) score, which evaluates the total number of detailed routing violations. Open and short routing violation is penalized by a factor of 1.0, while other routing violations are penalized by a factor of 0.2. The third metrics is the routed wirelength ($rtWL$) from the detailed router.

Table 5: Comparison on placement quality without placement migration, force directed placement migration and diffusion-based placement migration

	No PM		Force Directed PM		Diffusion-Based PM	
Bench	**rtWL**	**DR**	**rtWL**	**DR**	**rtWL**	**DR**
mgc_des_perf_1	-	-	2.02	1536.75	2.05	1744.80
mgc_des_perf_2	-	-	1.95	3.75	1.99	13.20
mgc_edit_dist_1	5.00	0.0	4.97	0.0	4.99	0.0
mgc_edit_dist_2	4.92	0.0	4.88	0.0	4.93	0.0
mgc_pci_bridge_1	0.412	0.0	0.371	0.0	0.382	0.0
mgc_pci_bridge_2	0.382	0.5	0.372	0.5	0.378	1.75
mgc_fft	0.680	225.95	0.673	165.2	0.683	250.10
mgc_matrix_mult	3.18	262.05	3.07	30.95	3.15	127.0
Norm.	-	-	1.000	1.000	1.013	1.230

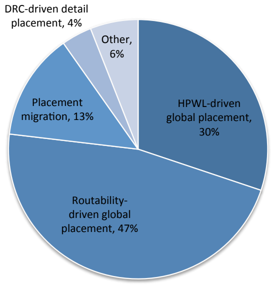


Figure 7: Run time breakdown

Table 6: Comparison on placement quality with and without DRC-driven detailed placer

	rtWL		DR	
Benchmark	**w/o DRC**	**w/DRC**	**w/o DRC**	**w/DRC**
mgc_des_perf_1	2.04	2.02	1377.15	1536.75
mgc_des_perf_2	1.94	1.95	7.4	3.75
mgc_edit_dist_1	4.99	4.97	0.0	0.0
mgc_edit_dist_2	4.90	4.88	0.0	0.0
mgc_pci_bridge_1	0.373	0.371	0.0	0.0
mgc_pci_bridge_2	0.375	0.372	0.0	0.5
mgc_fft	0.676	0.673	208.15	165.2
mgc_matrix_mult	3.09	3.07	31.55	30.95

The contest script first global routes the placement solution. If global routing congestion is higher than a certain threshold, detailed routing will not be performed on the given placement input. The threshold of suite A is that 1.75% g-edge overflow, and the threshold of suite B is that 0.25% g-edge overflow. Lower DP score implies less displacement, lower DR score implies less detailed routing violations and lower $rtWL$ means lower total routed wirelength. The goal is to minimize all three scores.

From Table 4, [7] makes the lowest score among the three metrics but does not report on the case suite of superblue. For all other teams, their DP scores are approximately at the same range. The DP score in mgc_des_perf_1 is slightly higher from our placer. We suspect it is because that this testcase has higher utilization rate for which our placement framework does not perform well. In DR score comparison, we score the least score in 6 out of 11 testcase. Our average routed wire length is similar to UW&UC and is slightly lower than CUHK and NCTU. The normalization value takes the arithmetic mean and normalizes other team's result to our work. The case suite of superblue for [7] and superblue_12 for the top three teams are excluded, since only our placer generates acceptable solution to detailed routing. Figure 7 shows our run time breakdown.

6.1 Placement Migration

Table 5 shows the effectiveness of our incremental placer. Placement solutions without placement migration are compared with two placement migration strategies. The first placement migration strategy (force directed PM) in Table 5 considers HPWL within the force model. The second placement strategy (diffusion-based PM) in Table 5 does not consider HPWL and moves cell based on the gradient of cell density.

Without placement migration, testcase *mgc_des_perf_1* and *mgc_des_perf_2* do not pass the global routing check, which do not proceed to the detailed routing stage. Using both placement migration strategies, both testcases passed the global routing check. However, considering HPWL within the force model produces placement solution with 1.7% less routing wirelength and 74% less DR score.

6.2 DRC Driven Detailed Placement

Table 6 compares placement solutions with and without DRC-driven detailed placement. DRC-driven detailed placement is more effective on testcases with lower utilization rate. From Table 6, DRC-driven detailed placement improves DR score on 3 out of 8 testcases and 2 out of 8 testcases have worse DR score. The two testcases with worse DR score have utilization rate above 84%.

7. CONCLUSION

In this work, we propose a comprehensive placement solution that resolves routing congestion from global routing stage to detailed routing stage. Optimization techniques for routability across each stage bf are integrated to one framework. Our placement framework includes a global router for routing congestion estimation, a global placer for routability optimization, an incremental placer to reduce cell density overlaps with better cell relative order, and a detailed placer to reduce design rule violations. Our placement framework considers design constraints in advanced node technology and placement quality is evaluated using commercial place and route tool. Experimental results show that placement migration is effective in reducing detailed routing violation while preserving the integrity of the original placement.

In the future, we are going to investigate the difference between using Gaussian Blurring or electrostatic model to derive the move force in Kraftwerk2's three force system. How to model the complex detailed routing rule, for example of the end-of-line rule, is still a research problem.

References

[1] X. He, W.-K. Chow, and E. F. Young. Srp: simultaneous routing and placement for congestion refinement. In *Proceedings of the International Symposium on Physical Design*, pages 108–113, 2013.

[2] X. He, T. Huang, W.-K. Chow, J. Kuang, K.-C. Lam, W. Cai, and E. F. Young. Ripple 2.0: High quality routability-driven placement via global router integration. In *Design Automation Conference*, pages 1–6. IEEE, 2013.

[3] X. He, T. Huang, L. Xiao, H. Tian, G. Cui, and E. F. Young. Ripple: an effective routability-driven placer by iterative cell movement. In *Proceedings of the International Conference on Computer-Aided Design*, pages 74–79, 2011.

[4] M.-K. Hsu, Y.-F. Chen, C.-C. Huang, T.-C. Chen, and Y.-W. Chang. Routability-driven placement for hierarchical mixed-size circuit designs. In *Proceedings of the Design Automation Conference*, page 151, 2013.

[5] M.-K. Hsu, S. Chou, T.-H. Lin, and Y.-W. Chang. Routability-driven analytical placement for mixed-size circuit designs. In *Proceedings of the International Conference on Computer-Aided Design*, pages 80–84, 2011.

[6] J. Hu, M.-C. Kim, and I. L. Markov. Taming the complexity of coordinated place and route. In *Design Automation Conference*, page 150, 2013.

[7] A. Kennings, N. K. Darav, and L. Behjat. Detailed placement accounting for technology constraints. In *International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 1–6, 2014.

[8] M.-C. Kim, J. Hu, D.-J. Lee, and I. L. Markov. A SimPLR method for routability-driven placement. In *Proceedings of the International Conference on Computer-Aided Design*, pages 67–73, 2011.

[9] M.-C. Kim, D.-J. Lee, and I. L. Markov. SimPL: an algorithm for placing VLSI circuits. *Communication of the ACM*, 56(6):105–113, June 2013.

[10] M.-C. Kim, N. Viswanathan, C. J. Alpert, I. L. Markov, and S. Ramji. MAPLE: Multilevel Adaptive Placement for Mixed-Size Designs. In *Proceedings of the International Symposium on Physical Design*, pages 193–200, 2012.

[11] 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 *Proceedings of the Design Automation Conference*, pages 200–205, 2010.

[12] W.-H. Liu, Y.-L. Li, and C.-K. Koh. A fast maze-free routing congestion estimator with hybrid unilateral monotonic routing. In *Proceedings of the International Conference on Computer-Aided Design*, pages 713–719, 2012.

[13] T. Luo, H. Ren, C. J. Alpert, and D. Z. Pan. Computational geometry based placement migration. In *Proceedings of the International Conference on Computer-Aided Design*, pages 41–47, 2005.

[14] H. Ren, D. Z. Pan, C. J. Alpert, and P. Villarrubia. Diffusion-Based Placement Migration. In *Proceedings of the Design Automation Conference*, pages 515–520, 2005.

[15] J. A. Roy, N. Viswanathan, G.-J. Nam, C. J. Alpert, and I. L. Markov. Crisp: congestion reduction by iterated spreading during placement. In *Proceedings of the International Conference on Computer-Aided Design*, pages 357–362, 2009.

[16] P. Spindler, U. Schlichtmann, and F. M. Johannes. Abacus: fast legalization of standard cell circuits with minimal movement. In *Proceedings of the International Symposium on Physical Design*, pages 47–53, 2008.

[17] P. Spindler, U. Schlichtmann, and F. M. Johannes. Kraftwerk2 - A Fast Force-Directed Quadratic Placement Approach Using an Accurate Net Model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(8):1398–1411, Aug. 2008.

[18] Y. Wei, C. Sze, N. Viswanathan, Z. Li, C. J. Alpert, L. Reddy, A. D. Huber, G. E. Tellez, D. Keller, and S. S. Sapatnekar. Glare: global and local wiring aware routability evaluation. In *Design Automation Conference*, pages 768–773, 2012.

[19] V. Yutsis, I. S. Bustany, D. Chinnery, J. R. Shinnerl, and W.-H. Liu. Ispd 2014 benchmarks with sub-45nm technology rules for detailed-routing-driven placement. In *Proceedings of the International Symposium on Physical Design*, pages 161–168, 2014.

[20] Y. Zhang and C. Chu. Crop: fast and effective congestion refinement of placement. In *Proceedings of the International Conference on Computer-Aided Design*, pages 344–350, 2009.