Modeling Repeaters Explicitly Within Analytical Placement

Prashant Saxena
Intel Labs (CAD Research)
2111 NE 25th Ave, Hillsboro OR 97124 USA
prashant.saxena@intel.com

Bill Halpin*
Synplicity, Inc.
600 W California Ave, Sunnyvale, CA 94086 USA
bhalpin@synplicity.com

ABSTRACT
Recent works have shown that scaling causes the number of repeaters to grow rapidly. We demonstrate that this growth leads to massive placement perturbations that break the convergence of today's interleaved placement and repeater insertion flows. We then present two new force models for repeaters targeted towards analytical placement algorithms. Our experiments demonstrate the effectiveness of our repeater modeling technique in preserving placement convergence (often also accompanied by wirelength improvement) at the 45 and 32 nm technology nodes.

Categories and Subject Descriptors
B.7.2 [**Integrated Circuits**]: Design Aids – *layout, placement and routing.*

General Terms
Algorithms, Design, Experimentation, Theory.

Keywords
Analytical placement, Buffering, Force-directed placement, Interconnect, Placement, Repeater insertion, Scaling.

1. INTRODUCTION
Wires scale much worse than devices, becoming increasingly resistive as process technology scales to yet smaller feature sizes. As a consequence, when an optimal interconnect is shrunk to the next technology node, it often requires additional repeaters. Indeed, recent technology studies [8] have demonstrated that the number of repeaters is growing rapidly as a consequence of the relatively poorer scaling of wires as compared to devices, and predicted that this will have a significant impact on various physical synthesis algorithms, including placement.

Previous works have often treated the repeater insertion and placement problems separately. The number of repeaters required by a wire is not known prior to the placement of the netlist. This poses the challenge of ensuring that the repeaters created during the placement process itself are also placeable, without disrupting the cells that have already been placed. Several researchers have proposed repeater blocks [3] or opportunistic repeater placement [1] to plan for this area impact of repeaters. However, with rapidly decreasing inter-repeater distances [8], the utility of pre-determined or opportunistic repeater banks decreases as the detours required by nets in order to access these banks become an increasingly large fraction of the desired inter-repeater distance itself. Furthermore, repeater banks also tend to become thermal hot spots. Consequently, finer-grained repeater placement approaches such as incremental repeater insertion within placement become more appealing for upcoming process nodes. The idea of incremental repeater insertion within placement was briefly discussed in [7]. Today's commercial physical synthesis tools also deal with this problem by invoking the placer in incremental mode in order to absorb the netlist changes created by repeater insertion (as well as limited re-synthesis and sizing) [2].

In this paper, we demonstrate experimentally that the current paradigms for handling repeaters by interleaving their insertion with incremental placement are not scalable. We use the force-directed quadratic placement approach [5] (a paradigm that is widely recognized as among the most stable and robust of today's competitive placement approaches [11]) to demonstrate that proposed flows such as [7][2][10] will no longer be able to converge at future technology nodes. We then describe a theoretical modification to this formulation that preserves convergence even in the presence of a large number of repeaters. This is achieved through new force models for the repeaters; we present two such models that allow us to capture the semantics of repeater insertion within the placement engine. Finally, we demonstrate experimentally that, unlike interleaved flows, our repeater modeling technique indeed allows us to preserve placement convergence at the 45 and 32 nm technology nodes, and does so without impacting the placement quality.

2. CURRENT APPROACHES
We define the placement and repeater insertion problem as follows. *Given a circuit netlist, a cell library, and process technology parameters, construct a nearly legal placement and modified netlist such that the repeaters required for all long wires have also been inserted and placed.*

Two typical state-of-the-art repeater insertion flows based on [5] are presented in Algorithm 1. The first, ***one-pass-repeater-flow*****,** works reasonably well with today's designs. It uses a traditional repeater insertion approach in which global placement is followed by repeater insertion and legalization. However, as the repeater count increases rapidly with process scaling, this approach breaks down, with the newly introduced repeaters creating too many placement overlaps to permit legalization without significant design perturbation. The second approach, ***incremental-repeater-flow***, is more sophisticated and seeks to interleave placement spreading with the netlist changes required for repeater insertion. However, there are several intuitive reasons why even this flow is not scalable, even though it works very well today (when the number of repeaters involved is small). Firstly, in this flow, the repeaters introduced within each iteration are treated as first-order objects, just like the other cells required for actual logic computation. However, these repeaters should exist only if their location on the net helps alleviate RC concerns. But traditional flows provide no means of retaining a repeater in its desired location in the middle of its net. If subsequent placement

* This work was carried out while B.Halpin was with Intel Corporation.

DAC 2004, June 7-11, 2004, San Diego, California, USA

procedure one-pass-repeater-flow
input: netlist
output:: placement, netlist with repeaters
generate spread global placement
generate required repeaters for placement
patch new repeaters into netlist
legalize // (if global placement converged)

procedure incremental-repeater-flow
input: netlist,, # of repeater insertion breakpoints b
output:: placement, netlist with repeaters
estimate spreading force f required for fully spread global placement
for i = 1 to b
apply f/b additional spreading force to cells
generate new placement
generate required repeaters for new placement
patch new repeaters (with locations) into netlist
restore placement state and cell locations
legalize // (if global placement converged)

Algorithm 1. State-of-the-art flows integrating repeater insertion with analytical placement.

iterations push it away from its desired location, it no longer serves any useful purpose (in fact, its existence actually restricts the placement flexibility of the remaining cells). Such repeaters should be removed as soon as possible; yet these flows provide no way of removing them until the next breakpoint is reached. Similarly, the opportunity for inserting new repeaters is also limited to the granularity of the interleaving frequency, thus leaving large portions of the solution space unexplored. Furthermore, incremental placement works well only when the percentage of modified nets is small; as more and more nets require repeaters [8], the consequent perturbation makes incremental placement infeasible. Finally, the splitting of a repeated net into subnets can cause poor routes for the net, even as each subnet is optimized locally by the placement engine.

A more flexible approach is required in which the repeater insertion problem is modeled directly within the placement formulation. This approach should retain the original netlist topology while also accounting for the actual area of the proposed repeaters at every stage of the placement. The existence of these repeaters should be contingent only on their effectiveness in wire RC alleviation, with all unneeded repeaters removed. Finally, the repeater insertion solution space should not be artificially restricted by some interleaving granularity. Rather, repeater requirements should be analyzed continuously, enabling on-the-fly repeater insertion or deletion as the placement evolves.

3. A KRAFTWERK PRIMER

The net modeling framework used in Kraftwerk [5] builds on Gordian's analytical model [6]. In this model, the objective function Φ seeks to minimize the weighted sum of squared Euclidean distances of connected cells and can be expressed in matrix notation as $\Phi = \frac{1}{2}\vec{p}^T C \vec{p} + \vec{d}^T \vec{p} + const$, where C is a *2n x 2n* symmetric matrix, $\vec{d}$ is a *2n*-dimensional vector, and $\vec{p} = (x_1,...,x_n, y_1,...,y_n)^T$ is a *2n*-dimensional vector representing the placement of the circuit (*n* being the number of cells in the circuit, with cell *i* located at (x_i, y_i)). Φ can be minimized by solving the linear system $C\vec{p} + \vec{d} = 0$. This formulation is equivalent to modeling nets as springs and calculating the state of equilibrium. While quadratic formulations for placement had been proposed earlier, Gordian's contribution was in using partitioning to spread cells rather than to reduce the size of the mathematical program. This analytical approach was extended in GordianL [9] which improved the wirelength further by optimizing for the (non-differentiable) "linear" sum-of-wirelengths objective function. This was accomplished through the use of an iterated weight function for each net that varied inversely with $|l|$, l being the current length of that net. The significant contribution of Kraftwerk [5] was to push the spring analogy further by iteratively creating a spreading force field $\vec{e}$, consisting of additional springs required to effectively spread the cells to cover the placement area. This formulation relied on certain mathematical conditions to ensure that the forces were easily computable. These conditions were: (1) The spreading force acting on a cell is a function of the cell location only; (2) Regions of high (low) density are sources (sinks) of this force field; (3) There are no loops in the force field; and, (4) The force equals zero at infinity. These conditions yield Poisson's equation, and the existence of a closed form solution to this equation allows the fast computation of the spreading forces.

The remainder of this paper addresses the question: "*Can we capture repeater semantics during placement without breaking the 'nice' properties of force computation?*"

4. PROPOSED APPROACH

We describe two different force models for repeaters. In both these models, the repeater forces are inserted *in addition to* the regular connectivity forces between the driver(s) and the receiver(s) of the net. Furthermore, whenever spreading forces are used, repeaters contribute to spreading forces on other cells (thus helping avoid overlaps), but are not themselves acted upon by the spreading forces (thus tending to remain close to their desired locations on the nets). The first model introduces a new decaying repulsive force that, when combined with the attractive connectivity forces within the quadratic placement model, allows the separation between the repeaters to be tuned to the desired inter-repeater length. The second model relies on quadratic attractive forces on each segment of a repeated line, causing the repeaters to be approximately equi-spaced along the line. Note that unlike the traditional quadratic formulation in which the forces modeling the original net's connectivity are discarded after repeater insertion, our formulations retain a "spring" corresponding to the original net in addition to the new quadratic wirelength terms for the subnets connected to the repeaters. This retention not only avoids the problems of net fragmentation (as discussed in Section 2), but also tends to reduce the wirelength for the repeated nets, thus reducing the overall repeater requirement for the entire design block.

4.1 Modeling Repeaters with Repulsive Forces

Of the four spreading force requirements derived in [5], the first two (viz., dependence only on the cell co-ordinates, and direction from high to lower density regions) serve solely to capture the semantics of overlap removal among cells. However, the third and fourth requirements (viz., loop-free force field, and zero forces at infinity) contribute directly towards the existence of a unique and easily computable solution. Therefore, we would like our new repeater forces to create a force field such that when it is superimposed on the spreading force field, these two mathematical requirements are still satisfied in the resultant field.

A combination of quadratic [6] or linear [9] connectivity forces and inversely varying repulsive forces on a repeater captures the semantics of preferred inter-repeater distances. If the repeaters are too close to each other (or to the driver(s) and/or receiver(s)), the repulsive forces dominate, pushing them apart. On the other hand, if they are too far apart, the connectivity forces overwhelm the repulsive forces and bring the repeaters together[1]. The higher the degree of these repulsive forces, the sharper is the resultant inter-repeater distance tuning (as these forces grow/die more rapidly as one moves away from the desired inter-repeater separation). Cubic and higher powers make the system hard to solve analytically, but inversely varying quadratic forces still result in an easily computable solution to the system as discussed next.

An inversely varying quadratic repulsive force on a repeater is of the form $\vec{f}_{rep} = \frac{A}{|\vec{r}|^2}\hat{i} + \frac{B}{|\vec{r}|^2}\hat{j}$, where $\vec{f}_{rep}$ is the repulsive repeater force corresponding to a separation $\vec{r} = (x, y)^T$, and A and B are some coefficients. It is easy to see that this template (and therefore its sum with the spreading force field [5]) satisfies the "zero force at infinity" requirement. The "loop-free" requirement is equivalent to showing that $\vec{f}_{rep}$ is conservative, i.e., there exists a scalar flux function $\phi_{rep}(x, y)$ such that $\vec{\nabla}\phi_{rep}(x, y) = \vec{f}_{rep}(x, y)$. Integrating and solving for ϕ_{rep} (after choosing to set the constants of integration to 0, since we need to show only the *existence* of some valid ϕ_{rep}), we obtain $\frac{B}{A} = \left[\frac{\pi}{Tan^{-1}\frac{y}{x}} - 1\right]\frac{x}{y}$ as a sufficient condition for the existence of ϕ_{rep}. This implies that the force field due to $\vec{f}_{rep}$ is conservative. Furthermore, observe that the gradient operation is linear, allowing the application of superposition. As a result, there exists a scalar flux function $\phi_{total} = \phi_{rep} + \phi_{overlap}$ such that its gradient equals the combined force field resulting from the spreading forces and the repulsive repeater forces. In other words, the combined force field is also "loop-free". Furthermore, the equation relating A and B derived above allows us to compute the repulsive force $\vec{f}_{rep}$ for any repeater efficiently.

4.2 Modeling Repeaters with Attractive Forces

Although the attractive force model does not enable us to tune the separation between repeaters, it allows for a simpler formulation that equi-spaces the repeaters along a net. It relies on the observation that, for any $l_1, l_2, l_3 ...$ such that Σl_i is constant, Σl_i^2 is minimized when all the l_i are equal to one another. While this goal can be automatically achieved in a pure quadratic formulation [6] (although at a large wirelength cost [9]), our experiments show that it is still not sufficient to ensure convergence. This is because repeaters fragment a long net into smaller subnets (of lengths $l_1, l_2, l_3 ...$, say). Then, observe that since $\Sigma l_i^2 < (\Sigma l_i)^2$, the total force shrinking the length of the original net is much smaller than in the unrepeated case. Consequently, the fragmentation of the long nets causes them to become even longer after repeater insertion, requiring yet more repeaters in the process. This is in addition to the large overall wirelength deterioration caused due to the use of the non-linearized objective function.

In the more practical formulations of force-directed placement that linearize the wirelength objective function [9], one must explicitly retain the quadratic terms corresponding to subnets of repeated nets. Thus, while the net weights for a regular net with length currently equal to λ (say) are successively divided by $|\lambda|$ in a nested iteration until they converge within each iteration of the placement engine [9], the weights for the nets attached to the repeaters remain unchanged within the nested iterations. This mechanism also allows us to leverage the theory underlying the quadratic formulation by reducing the repeated net problem to a version of the standard problem, albeit with different net weights.

Since equi-spacing of the repeaters on a net is achieved only if the quadratic forces on either side of a repeater are balanced, multi-terminal nets pose a problem for this simple formulation. Given any net topology model for such nets, one must ensure that this balancing is done across each repeater having more than one fanin or fanout net. Recall that the separability of the x and y terms in the objective function allows each iteration of quadratic placement to operate on only one of the two (i.e. x and y) co-ordinates at a time. Let the current iteration be focusing on the w co-ordinate (where $w = x$ or y). Then, if the relevant co-ordinate of a repeater's location is w_{rep} and those for the far ends of the different subnets connected to it are represented by w_i (*i=1,2,..*), then a normalization factor that we empirically found to be effective for equi-spacing within the linearized objective function framework of [9] is $\sum_{w_{rep}>w_i}(w_{rep} - w_i) \Big/ \sum_{w_{rep}<w_i}(w_i - w_{rep})$ for nets with $w_i > w_{rep}$, and its reciprocal for nets with $w_i < w_{rep}$.

4.3 Repeater Insertion and Deletion Granularity

The work described in this paper is orthogonal to the specific choice of the repeater insertion heuristic; instead, it focuses on ensuring that the repeater locations determined by any heuristic are kept physically feasible (and updated as the net's terminal locations evolve) during the placement process.

We update the repeater requirements of the netlist being placed, during each KraftWerk iteration. For each net with too few or too many repeaters, we call a repeater insertion heuristic that determines the ideal locations for all the repeaters required on that net. We do this after the overlap forces have been computed within each KraftWerk iteration as in [5]. For the repulsive force model, we then compute the repulsive repeater forces, scale them and add them to the spreading forces. Next, we create transient quadratic or linear nets (as needed) to model the connectivity of the subnets attached to the repeaters. This modified netlist and the current net weights and cell forces and locations are then used to generate the coefficients in the linear matrix system solved within

[1] In practice, one inter-repeater stage along the net should have comparatively weaker forces to handle the corner-case when the net length changes appreciably, causing the preferred number of repeaters on the net to also change. This "slack" prevents equi-spaced repeaters with large separation (if the net has lengthened) or detoured repeaters (if the net has shrunk), and the change is localized to the "weak" stage.

the current KraftWerk iteration. Subsequent to this solution, the cell locations are updated and the system checked for convergence. If convergence has not yet been achieved, the next placement iteration is initiated.

5. EXPERIMENTAL RESULTS

We implemented the attractive force model for repeaters as a system called MorePlace ("*Modeling repeaters explicitly during Placement*") on top of an industrial placement engine Gplace that implements KraftWerk [5] along with objective function linearization [9]. We tested MorePlace on circuits from a recent microprocessor[2], with the desired inter-repeater distances corresponding to the 45 and 32 nm nodes as in [8]. The circuits range from 3978 to 42127 cells (Table 1). All experiments were run on a 2.8 GHz Intel® Xeon™ server with 4 GB of memory.

Table 1. Testcase details.

Testcase	Ckt_A	Ckt_B	Ckt_C	Ckt_D	Ckt_E
# Cells	3978	4014	12312	13343	42127
# Nets	4268	4384	13073	17685	42247

We conservatively predict the repeater requirements of a net based on its minimum length route, since any detours will only worsen the situation). If l_{rep}^{M} is the optimal inter-repeater distance on metal layer *M*, we insert a new repeater on a net routed on *M* only if its length is greater than $1.4 * l_{rep}^{M}$; similarly, we do not delete an existing repeater from a net unless the inter-repeater distance has shrunk to less than $0.7 * l_{rep}^{M}$. As in [8], we assume that since short nets will be routed on the lower metal layers, we can use l_{rep}^{M3} to determine their repeater needs. Similarly, we determine the repeater needs for the longer nets (with unrepeated length greater than $4 * l_{rep}^{M3}$) using l_{rep}^{M6}. We have validated this model as being conservative against the tape-out data for a recent 90 nm microprocessor, confirming that it consistently under-counts the repeater needs across all wire length ranges.

Global placement terminates when KraftWerk claims convergence (i.e., stabilized cell locations, net weights and forces); at that stage, we would like the number of repeaters needed for electrical convergence to be relatively small. We report data after global placement rather than after legalization. This is because placements produced by interleaved flows usually can not be legalized, owing to a large number of repeaters required even after the placement has converged physically. The perturbation caused by patching all these repeaters into the placed netlist destroys the "converged" placement produced by the interleaved flows.

We first study the behavior of MorePlace, the one-pass flow and various interleaved flows in depth on one of our testcases (viz., the median-sized *Ckt_C*), in order to justify the choice of the 3-breakpoint interleaved flow as the best of the interleaved and one-pass flows. We then compare this flow against MorePlace on the remaining testcases also.

5.1 Placement by MorePlace

We first demonstrate that MorePlace produces placements that are of a high quality, and that repeater semantics are indeed respected by MorePlace. The placement for *Ckt_C* produced by MorePlace at the conclusion of the KraftWerk iterations is shown in Figure 1. Regular movable cells are represented in white, while the newly inserted repeaters are in red (gray in B&W). The block also has four pre-placed macros. It can be seen from the layout that the distribution of the repeaters is non-uniform, but still fine-grained. This suggests that statistical approaches like pre-planned repeater banks or the *a priori* reservation of some fraction of each row for repeaters will not be very effective in this scenario.

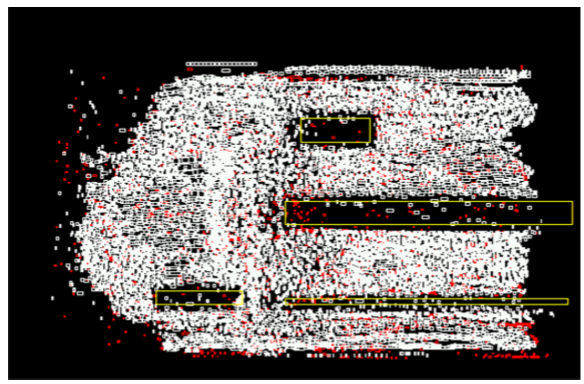

Figure 1. Placement for *Ckt_C* produced by MorePlace.

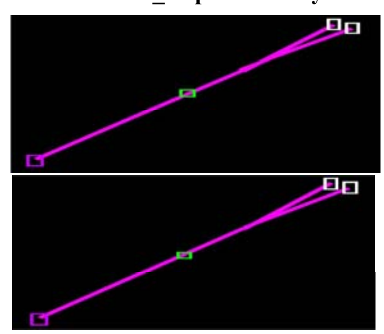

Figure 2. "Before" and "After" MorePlace screenshots of a typical repeated net.

The effectiveness of our repeater force modeling technique is brought out by the screenshots of a typical repeated net in *Ckt_C*, presented in Figure 2. The "before" image shows the desired position of the repeater (prior to invoking the solver), while the "after" image shows that the repeater indeed stays in this location even after all the cells have been updated to their new locations. Furthermore, the length of the net does not change appreciably in the process. This behavior is in sharp contrast to the repeater collapse or severe wirelength deterioration that occurs with traditional KraftWerk formulations (See Figure 5 and accompanying text). Furthermore, the desired behavior is

[2] In order to model wiring and placement complexity at future process technology nodes, we used unshrunk cells along with shrunk inter-repeater distances. This model was validated via experiments in which we shrank all the relevant geometries (including the cell sizes), and then studied the convergence obtained by traditional flows and MorePlace. Compared to our model of choice, this "shrink-all" mode worsened the convergence problems and wirelength deterioration with the traditional flows vis-a-vis MorePlace even more. Furthermore, unlike our model, the "shrink-all" mode does not capture the synthesizable block size increase (and consequent increase in wiring complexity and repeater demands) that occurs with scaling. Thus, our model lower-bounds the actual convergence problems that can be expected at future nodes.

achieved without disproportionately large balancing forces keeping the repeaters in their place – as is evidenced by the fact that the overall wirelength of this layout is comparable to that obtained by traditional flows (See Tables 2 and 3).

5.2 Convergence Comparisons

Gplace includes a very sophisticated incremental mode for its KraftWerk implementation. This mode is used extensively for incremental sizing, buffering and local resynthesis interleaved with the global cell placement. In this mode, described in the *incremental-repeater-flow* procedure in Algorithm 1, the user first specifies the desired number of break-points. Then, Gplace does a complete KraftWerk run without any breakpoints to estimate the spreading forces and wirelengths upon convergence. It then re-starts the KraftWerk iterations, using these estimates and the instantaneous convergence rates to determine the breakpoint iterations dynamically. At each breakpoint, the netlist is modified as needed by other synthesis or circuit optimization tools. This interleaved flow has proven very successful in supporting physical synthesis at the current (90 nm) technology node. We have empirically observed that the use of three breakpoints seems to yield the best results in this flow at this node. Figures 3 and 4 below indicate that the 3-breakpoint variant is the best-performing choice from among these traditional flows at the 32 nm node also.

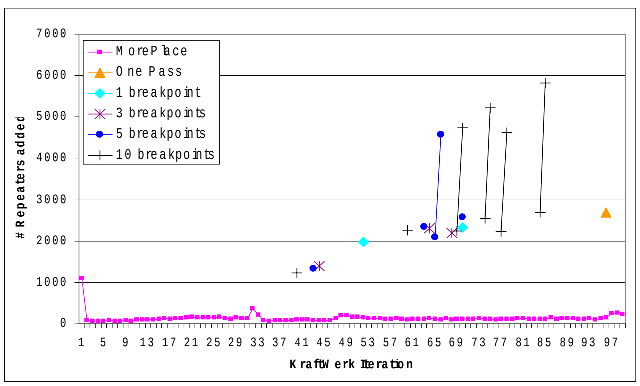

Figure 3. Number of repeaters inserted per iteration in *Ckt_C* in MorePlace, one-pass and various interleaved schemes.

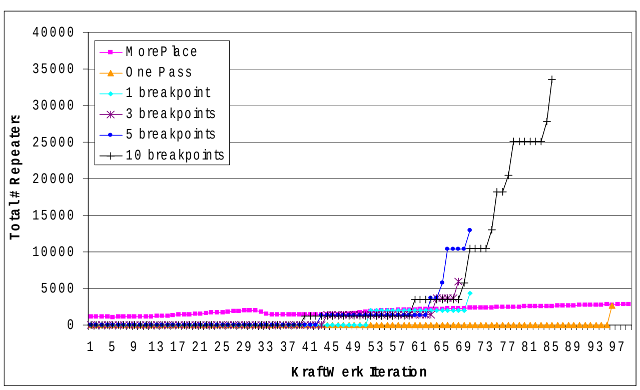

Figure 4. Cumulative number of repeaters inserted in *Ckt_C* in MorePlace, one-pass and various interleaved schemes.

Figures 3 and 4 compare the performance of the one-pass flow (described in the *one-pass-repeater-flow* procedure in Algorithm 1), interleaved flows with 1, 3, 5 and 10 breakpoints, and MorePlace on *Ckt_C* at the 32 nm node. Figure 3 shows the number of repeaters added after each iteration of KraftWerk (note that the number of repeaters deleted in each iteration by MorePlace is not included in this chart, since deletions usually do not cause as much of a perturbation as insertions), while Figure 4 shows the cumulative number of new repeaters in the design after each iteration of KraftWerk. Figure 3 shows that, among the one-pass and interleaved flows, it is the 3-breakpoint flow that requires the fewest repeaters (2195) when KraftWerk terminates. It also shows that, while MorePlace requires 1101 repeaters in the first iteration, the number of new repeaters quickly stabilizes and never rises above a couple of hundred repeaters for any subsequent iteration. In contrast, the one-pass and each of the interleaved flows require >1000 repeaters at *each* breakpoint, and >2000 at the time the global placement "converges". Furthermore, when a netlist is patched with the repeaters required by it at any breakpoint, the next iteration shows the repeater demand increasing even more (because of the perturbation introduced by the newly added repeaters) before it stabilizes again; this phenomenon can be seen in the 5-breakpoint and 10-breakpoint cases, where some breakpoints happen to fall in consecutive iterations (and are depicted linked by vertical lines).

Since the netlist requires a huge number of repeaters at *each* breakpoint in the interleaved flows, the cumulative number of repeaters (shown in Figure 4) grows very rapidly with the number of breakpoints, making using a large number of breakpoints impractical. This cumulative number is 4313, 5897, 12931 and 33603 for the 1, 3, 5, and 10 breakpoint flows respectively, in contrast to 2858 for MorePlace and 2686 for the one-pass flow. However, unlike MorePlace that requires the addition and deletion of only 231 and 185 repeaters respectively in its final iteration, the one-pass flow layout is unrealizable because it still requires all 2686 repeaters to be added to its "converged" netlist.

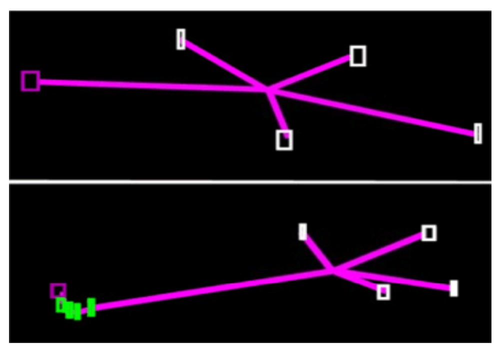

Figure 5. Screenshots of a typical repeated net at the first breakpoint and at termination, using the 3-breakpoint interleaved flow with a linearized objective function.

It is interesting to study why the repeater demand of the netlist does not decrease in spite of the repeaters added to it at each breakpoint in the interleaved flows. Since a quadratic objective function grows more rapidly than a linear one, the approximation of the latter by iterating using the former while adjusting the net weights [9] tends to emphasize the net weights for shorter nets (for which the quadratic function is smaller than the linear one) at the cost of those for the longer ones, thus pushing the quadratic function closer to the linear one for all the nets. However, as a result, even as the overall wirelength decreases, short nets tend to become even shorter. Therefore, without explicit repeater modeling, a repeater inserted at the mid-point of a net is in a state of unstable equilibrium. If a slight perturbation from its ideal location causes the subnet on one side of the repeater to become slightly shorter than that on the other side, the system tends to push the repeater even further towards the shortened subnet. The final result is the collapse of the all repeaters on a net towards one end of that net, as can be seen for a representative net from *Ckt_C* in Figure 5. These collapsed repeaters no longer serve any useful purpose and can be deleted. But, even if we were to augment the interleaved flow by deleting unneeded repeaters at each breakpoint, the collapsing behavior suggests that it would not help the convergence (because repeaters inserted on long nets would collapse and be deleted, still leaving the nets unrepeated).

Since this collapse is caused by the linearization of the objective function, one could hope that retaining the quadratic objective function of [6] would make the flow convergent, even if at the cost of increased wirelength. However, with a (non-linearized) quadratic objective function, the interleaved flow with 3 breakpoints on *Ckt_C* at 32 nm requires 1129, 1834 and 1469 repeaters to be patched into the netlist at the 3 breakpoints respectively. Furthermore, the total wirelength worsens by 23.2%. Although the repeaters don't collapse into the net terminals, repeated nets become much longer compared to when they are left unrepeated, thus requiring yet more repeaters (as discussed in Section 4.2). Consequently, the placement still fails to converge.

Table 2. Residual repeater requirements upon KraftWerk convergence at the 32 nm node.

Testcase	*MorePlace*		*Interleaved*		
	Repeater Delta	CPU (sec)	**Repeater Delta**	CPU (sec)	Wirelength change (%)
Ckt_A	**+15 -10**	129	**+224**	170	-14.28
Ckt_B	**+12 -9**	124	**+336**	166	0.30
Ckt_C	**+112 -155**	663	**+2196**	721	-3.69
Ckt_D	**+268 -297**	829	**+5286**	1025	13.96
Ckt_E	**+666 -548**	1993	**+13237**	3851	19.51

Table 3. Residual repeater requirements upon KraftWerk convergence at the 45 nm node.

Testcase	*MorePlace*		*Interleaved*		
	Repeater Delta	CPU (sec)	**Repeater Delta**	CPU (sec)	Wirelength change (%)
Ckt_A	**+2 -1**	89	**+46**	229	-11.81
Ckt_B	**+2 -1**	95	**+43**	178	-10.09
Ckt_C	**+53 -79**	465	**+1021**	486	-7.61
Ckt_D	**+148 -178**	630	**+2675**	856	5.49
Ckt_E	**+414 -406**	2273	**+6843**	3329	16.18

We now compare the performance of the 3-point interleaved flow (empirically determined to be the best among the interleaved and one-pass flows) to that of MorePlace on our entire test suite. Tables 2 and 3 present the number of repeaters still required by the netlist when the global cell placement has "converged", at the 32 nm and 45 nm nodes respectively. The data in the Column titled *Repeater Delta* is a measure of the extent of perturbation caused by the subsequent electrical convergence of the placed netlist; positive (negative) numbers indicate the number of repeaters that still need to be added (deleted). *On all the testcases, this number is more than an order of magnitude higher for the interleaved flow than for MorePlace*, showing that the layouts produced by the interleaved flow are not realizable. In contrast, for the layouts from MorePlace, this number is well within 2% of all the cells and can be easily handled by today's legalizers without much design perturbation. Furthermore, on the larger examples (especially at the 32 nm node), this improved convergence is accompanied by improved wirelength also, as can be seen from the column titled *Wirelength Change.* Positive numbers in this column correspond to cases where the interleaved flow results in a longer total wirelength (even though the layout is not electrically converged). The data shows that *as the wiring complexity (and consequently, the repeater demand) of the design increases (either due to increasing design size or due to process scaling), MorePlace not only yields more convergent designs but also does so with an improved wirelength*. Furthermore, it does not require any extra runtime for these improvements.

Note that, as mentioned in Footnote 2, the wiring complexity of typical design blocks at future technology nodes will usually be higher than those of our testcases [4]. Consequently, interleaved flows will perform even more poorly on them as compared to MorePlace than is observed in our experiments.

6. CONCLUSION

We have demonstrated the non-scalability of today's incremental placement paradigms due to repeaters. Instead, repeaters need to be modeled natively within the placement engine. We present two such models for analytical placement. Our experiments highlight the effectiveness of our repeater modeling technique in preserving placement convergence even at the 45 and 32 nm technology nodes without sacrificing placement quality.

7. REFERENCES

[1] Alpert, C.J., Gandham, G., Hrkic, M., Hu, J., and Quay, S.T., Porosity aware buffered steiner tree construction. *Int. Symp. Physical Design*, 2003, 158-165.

[2] Changfan, C., Hsu, Y.-C., and Tsai, F.-S. Timing Optimization on Routed Designs with Incremental Placement and Routing Characterization. *IEEE Trans. CAD*, *19, 2* (Feb. 2000), 188-196.

[3] Cong, J., Kong, T., and Pan, D.Z. Buffer block planning for interconnect-driven floorplanning. *Int. Conf. Computer-aided Design*, 1999, 358-363.

[4] Davis, J.A., De, V.K., and Meindl, J.D. A Stochastic Wire-length Distribution for Gigascale Integration (GSI). *IEEE Trans. Electron Devices*, *45, 3* (Mar. 1998), 580-597.

[5] Eisenmann, H., and Johannes, F.M. Generic Global Placement and Floorplanning. *Design Automation Conf.*, 1998, 269-274.

[6] Kleinhans, J.M., Sigl, G., Johannes, F.M., and Antreich, K. GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization. *IEEE Trans. CAD, 10, 3* (Mar. 1991), 356-365.

[7] Rajagopal, K., Shaked, T., Parasuram, Y., Cao, T., Chowdhary, A., and Halpin, B. Timing Driven Force Directed Placement with Physical Net Constraints. *Int. Symp. Physical Design*, 2003, 60-66.

[8] Saxena, P., Menezes, N., Cocchini, P., and Kirkpatrick, D.A. Repeater Scaling and its Impact on CAD. *IEEE Trans. CAD, 23, 4* (Apr. 2004).

[9] Sigl, G., Doll, K., and Johannes, F.M. Analytical Placement: A Linear or a Quadratic Objective Function? *Design Automation Conf.*, 1991, 427-432.

[10] Stenz, G., Riess, B.M., Rohfleisch, B., and Johannes, F.M. Performance Optimization by Interacting Netlist Transformations and Placement. *IEEE Trans. CAD, 19, 3* (Mar. 2000), 350-358.

[11] Villarrubia, P. Important Placement Considerations for Modern VLSI Chips. Invited talk, *Int. Symp. Physical Design*, 2003.