Congestion Driven Quadratic Placement

Phiroze N. Parakh, Richard B. Brown, Karem A. Sakallah
University Of Michigan
1301 Beal Ave, EECS - ACAL
Ann Arbor, MI. 48109-2122
1-(734)-647-0301
{phiroze, brown, karem}@umich.edu

1. ABSTRACT

This paper introduces and demonstrates an extension to quadratic placement that accounts for wiring congestion. The algorithm uses an A* router and line-probe heuristics on region-based routing graphs to compute routing cost. The interplay between routing analysis and quadratic placement using a growth matrix permits global treatment of congestion. Further reduction in congestion is obtained by the relaxation of pin constraints. Experiments show improvements in wireability.

1.1 Keywords

Congestion, Quadratic placement, Relaxed pins, Global routing, Routing models, Supply-demand.

2. INTRODUCTION[1]

The drive toward higher levels of integration has invalidated classical assumptions in VLSI CAD. Previous abstractions have proven to be inadequate, thus changing the intent of CAD from optimization of gates to the management of routing resources. A design that effectively uses its resources requires smaller area because inefficient utilization leads to more wire crossings, increased wire length and poor channeling.

With the advent of overcell routing, the goal of every place and route methodology has been to utilize all available active area to prevent spilling of routes into channels. It is this overflow of routes that accounts for an increase in area. “Congestion aware” global routers avoid this by constructing a set of routing trees for all nets in the design to match the global supply. Further heuristics are then applied to manage local congestion to enhance and improve the final route quality. The concept of supply and demand to drive global routers is favorably demonstrated in [13]. The objective of this global router is defined in terms of route uniformity and route density. However, this scheme and others like it are unlikely to achieve optimality because the quality of a routing solution is largely determined by the input placement, with congestion occurring where demand exceeds supply. Thus the need rises to model congestion within placement.

Typical placement objectives involve reducing net-cut costs or minimizing wire length. Due to their constructive nature, min-cut based strategies minimize the number of net crossings but fail to uniformly distribute them [14]. Quad-partitioning schemes, first demonstrated in [8] did account for some global routing resources, but did not address supply nor account for internal routing. Congestion-driven placement based on multi-partitioning was proposed in [10], but was limited in the number of partitions due to the use of pre-computed Steiner trees. To our knowledge, congestion driven placement has not used congestion at a global level to update global placement. Alternatives to min-cut based schemes attempt to minimize wire length. The use of minimal wire length as a metric to guide placement has been successful in achieving good placement. However, it only *indirectly* models congestion and the behavior of the router. Reducing the global wire length helps reduce the *wiring demand,* but does not affect the *wiring supply.* It is entirely feasible for a minimum wire length solution to require more tracks for routing through a region, than are available. Therefore, schemes such as quadratic placement which are based solely on wire length minimization cannot adequately account for congestion. A simulated annealer in [1] used wire distribution functions and net bounding boxes for routing-resource demand modeling. However, the complexity of its wire model precluded a global view and forced the analysis to be performed in smaller blocks.

The quadratic fomulation for squared wirelength can be solved for a global minimum. However the solution contains much cell overlap which must be resolved by partitioning with appropriate constraints. This methodology is incapable of producing *sufficient non-overlap* to account for routing resources. This paper develops a framework that drives quadratic placement to relieve congestion while simultaneously solving for minimal wire length. Supply-demand weights are computed through appropriate modeling of routing resources on a region-based routing graph. These weights influence the quadratic placer into growing (or shrinking) regions based on resource demand. Repeated partitioning during quadratic placement allows the analysis to converge upon a dense routing grid. The dynamic behavior of this graph supports increasingly accurate supply-demand modeling at each iteration of the algorithm. The current implementation of the supply-demand algorithm creates global routes between regions using an A***** search [2] algorithm and computes internal costs using a line-probe [3] heuristic. However, this component of the algorithm can be replaced with any appropriate route model. Further reductions in congestion, wire-length and area are obtained by the inclusion of relaxed pin constraints into the formulation.

Section 3 of the paper presents the key concepts in quadratic placement. Section 4 describes region-updating during quadratic placement along with the necessary framework for congestion analysis on a region-based grid. Results of this new methodology are presented in Section 5, followed by conclusions in Section 6.

[1] This work was supported by ARPA/ARO. Contract: DAAH04-94-G-0327

35th Design Automation Conference ®

1-58113-049-x-98/0006/$3.50
DAC98 - 06/98 San Francisco, CA USA

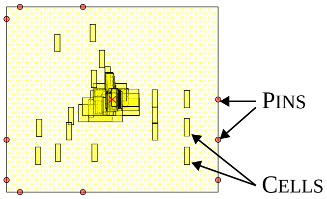

Figure 1. First iteration of Quadratic placement

3. QUADRATIC PLACEMENT

Quadratic placement engines [7, 6, 9] solve an unconstrained minimization problem, the objective function $\Phi_q(\boldsymbol{x}, \boldsymbol{y})$ of which is the squared wirelength

$$\Phi_q(x, y) = \sum_{i,\,j} a_{ij}(x_i - x_j)^2 + \sum_{i,\,j} a_{ij}(y_i - y_j)^2 \qquad (1)$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are coordinate vectors for the n cells and $a_{ij} \in \{0, 1\}$ represents connectivity between cells i, j. This can be represented in two systems of linear equations, separable in x and y. By expansion and proper assimilation of common terms, (1) can be rewritten in the more familiar form[1] of

$$\Phi_q(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}_x^T\boldsymbol{x} \qquad (2)$$

and optimally minimized by solving $\nabla\Phi_q(\boldsymbol{x}, \boldsymbol{y}) = 0$, where $\boldsymbol{Q}$ is the $n \times n$ Laplacian matrix for the netlist. The vector $\boldsymbol{d}_x$ originates from the locations of the pins and can be adjusted to include port locations on the cells. The solution for Φ_q tends to group around the origin [Figure 1]. In order to disperse the cell locations and resolve overlaps, the solutions are partitioned and iterated upon.

The quadratic placer used in this work makes use of the GORDIAN algorithm by Klienhans et. al [7]. Initially, Φ_q is constructed and minimized to obtain the $\boldsymbol{x}$ and $\boldsymbol{y}$ coordinates. These vectors are used to aid the partitioner in generating new regions. New center of gravity constraints are then imposed upon each region. The process is repeated, alternating in horizontal and vertical directions until each region represents an individual cell. Generating center-of-gravity constraints for each region has the effect of distributing the cells by forcing the area-weighted mean location of cells within a region to correspond to the region's center of gravity. These constraints are embedded into (2) through $\boldsymbol{Z}$, yielding the unconstrained linear system

$$\psi(\boldsymbol{x}_I) = \frac{1}{2}\boldsymbol{x}_I{}^T\boldsymbol{Z}^T\boldsymbol{C}\boldsymbol{Z}\boldsymbol{x}_I + c^T\boldsymbol{x}_I \qquad (3)$$

on the $\boldsymbol{x}_I$ independent cells. Which can be minimized by setting $\nabla\psi(\boldsymbol{x}_I) = 0$, and solving

$$\boldsymbol{Z}^T\boldsymbol{C}\boldsymbol{Z}\boldsymbol{x}_I = -\boldsymbol{c} \qquad (4)$$

Quadratic placement algorithms require pin locations, else they return a trivial solution with all cells at the origin. Graphically, the pins stretch the network of cells to induce a valid placement solution, and therefore, they have a strong impact on the solution. While it may be prudent to assign pin locations based on floorplanning, or through other top-down pin-placement approaches, the location (or order) of these pins may aggravate the placement. Consider the fact that incorrect pin ordering could twist the design and thus create congestion. In order to reduce congestion, the placement should be able to influence the pin locations.

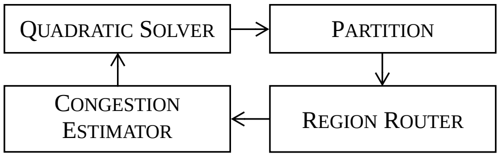

Figure 2. Congestion driven Placement methodology

4. CONGESTION DRIVEN PLACEMENT

Figure 2 illustrates the congestion-driven placement methodology. Fundamental to this technique is the need to model congestion and to influence the quadratic placer with a congestion metric. As in the GORDIAN algorithm, placement produced by minimizing the wire length metric is iteratively partitioned into regions, and placed with new center of gravity constraints. Before each successive placement, internal route estimation and a region-based global route are performed on each region to estimate supply-demand ratios. These ratios are used to influence the quadratic placer into growing (or shrinking) regions based on resource demand. Furthermore, congestion induced by incorrect pin ordering is relieved by relaxing the pin constraints to a single dimension. Experimental observations show that computing the internal and external routes takes much less time than a single minimization of Φ_q.

4.1 Growing and Shrinking Regions

For q regions, we define $\boldsymbol{G}$ as an $\langle n - q \times n - q\rangle$ diagonal *growth* matrix with entry g_{ii} equal to the region weight for the independent cell $\boldsymbol{x}_i$. If we let $\boldsymbol{x}_i' = \boldsymbol{G}\boldsymbol{x}_i$ and substitute for $\boldsymbol{x}_i$ in (4), we can obtain a global minimum of Φ_q by solving:

$$(\boldsymbol{G}^{-1}\boldsymbol{Z})^T\boldsymbol{C}(\boldsymbol{Z}\boldsymbol{G}^{-1})\boldsymbol{x}_i = -c' \qquad (5)$$

with $c'^T = (\boldsymbol{C}\boldsymbol{x}_0 + \boldsymbol{d}_x)^T\boldsymbol{G}^{-1}\boldsymbol{Z}$. We know that $\boldsymbol{Z}$ forms a basis for the positive definite $\boldsymbol{C}$ [7], and since $\boldsymbol{G}$ is a diagonal matrix with positive diagonal entries, $\boldsymbol{G}^{-1}\boldsymbol{Z}$ will have the same form as $\boldsymbol{Z}$ and thus (5) is a positive definite system. Standard SOR or Krylov subspace solvers [12, 5] can be used to solve the linear system.

The use of $\boldsymbol{G}$, causes cell positions to reflect supply-demand ratios. Resource limited regions are expanded (or reduced) to account for the wiring demand imposed upon them. The horizontal ratio is used to stretch the region in $\boldsymbol{y}$, while the vertical ratio stretches the region in $\boldsymbol{x}$. The growth factors disperse modules within a region and influence other cell positions while permitting the solver to minimize Φ_q. Reduction in congestion occurs due to the ability of the model to transform horizontal routes into vertical and vice-versa. In Figure 3, if a region is deemed vertically congested, then a vertical expansion is performed. At this point, internal horizontal nets could become vertical, thus relieving congestion. Alternatively, if the region is shrunk horizontally, during the next iteration more nets from the previous iteration could become vertical, further relieving congestion.

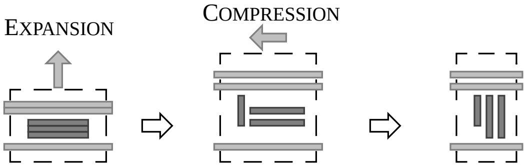

Figure 3. Example of region growth relieving congestion

[1] Shown for the x-dimension only.

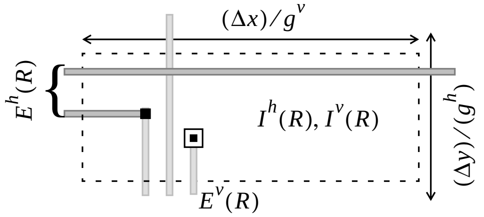

Figure 4. Region supply and demand components

4.2 Computing the Growth Matrix

After each minimization of ψ, the current regions are partitioned to generate a new set of regions. For each congestion analysis, a new routing graph is constructed using these regions. An important characteristic of this routing graph is its *dynamic* behavior. With each successive placement and partitioning, the granularity of the routing graph increases, thus increasing the quality of the analysis, until a detailed routing graph is achieved. Furthermore, since the regions reflect well-connected cells, they form a more appropriate routing graph than one produced for an arbitrary pitch. Together, these features permit the appropriate amount of supply and demand analysis at each iteration.

The supply created by each region R (Figure 4) is separated into horizontal and vertical components: R_s^h and R_s^v, and cannot exceed the capacity for the region, which is determined by the horizontal and vertical metal pitch (g^v, g^h). Internal blockages, reflecting pre-allocated routing channels and port locations on cells diminish the supply. Each cell $\mu \in R$ contributes μ_f to the total number of vertical and horizontal feed-thrus for the region. This amount is normalized to the width R_w (or height) of the region, thus producing the incident supply of a region

$$R_s^v = \frac{\sum_{\mu \in R} \mu_f^v}{(\sum_{\mu \in R} \mu_w)/R_w} \qquad (6)$$

where μ_w is the cell width. The horizontal component is expressed identically. Demand on each region is comprised of external, $E(R)$, and internal, $I(R)$, components

$$R_d^v = E^v(R) + I^v(R) \qquad (7)$$

The external cost originates from nets that span multiple regions and is computed by the region router. It is the sum of all region-routes that intersect it and includes routes that terminate within it. In order to compute the internal-route cost, a heuristic that mimics a line-probe router is used. Figure 5 illustrates the line-probe internal [3] cost for horizontal and vertical tracks. Once the supply and demand values for a region are determined, their difference determines the number of extra routing tracks required and therefore, the amount of region growth required to satisfy demand.

$$w_R^v = 1 - \frac{(R_s^v - R_d^v) \cdot g^v}{\Delta x_R} \qquad (8)$$

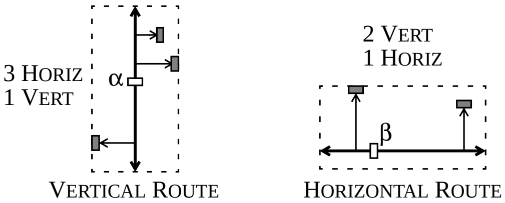

Figure 5. Internal routing model

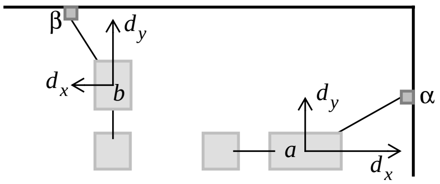

Figure 6. Effect of fixed pin locations

4.3 Relaxed Pins

The task of optimally assigning pin locations to a large block of standard cells is NP-complete. Consider the fact that the optimal pin location permits an optimal placement which is a known NP-complete problem. While classical top-down floorplanning does aid in establishing the locations of pins on a block, it does not consider the resultant effect on the block. A poor ordering of pins could, therefore, unduly “torque” the block, creating congested regions. This effect can be very pronounced in quadratic placers, where the placement solution, ultimately, is derived from the locations of the pins. From (2), we know that $\Phi(x)$ and $\Phi(y)$, are functions of d_x and d_y respectively, with each fixed pin imparting a d_x and d_y component to the layout. Thus pin α in Figure 6 pulls module a up as well to the right. It is obvious that module a will be unduly moved to minimize the vertical component on it. This effect can be nullified by setting the d_y component due to α, and likewise the d_x component for β, to zero. This has the effect of allowing α to track a vertically and β for b horizontally. In our experiments this approach produced wirelength reductions of as much as 7% for quadratic placement and 10% for linearized quadratic placement. Similar reductions in area were achieved, thus showing a reduction in congestion.

5. IMPLEMENTATION AND RESULTS

5.1 The Region Router

Routing algorithms are numerous. The algorithm for this application should accurately estimate supply and demand while maintaining computational efficiency. It is critical that the region router used in this methodology match the characteristics of the detailed router. Discrepancies between the characteristics of the two could lead to incorrect assumptions, causing the placer to create inappropriate placement for the final route. In this preliminary implementation, routing on the region-based graph was performed by an A***** algorithm [2]. The algorithm initially sorts the sinks in order of criticality (manhattan distance to source). The most critical sink is selected and routed to the nearest edge on the current tree. The tree is then updated with the new edges and the criticality of the remaining nodes are redetermined. The process terminates when all nodes are connected. In our methodology, it is not essential for the router to be congestion aware, as the placer will yield solutions that permit direct routes.

Two implementations of the sink lists can be considered: region-

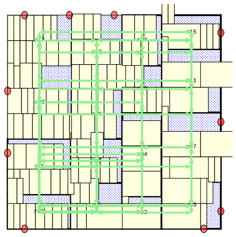

Figure 7. Global router on s298 in region center mode

Table 1: Benchmark Information

Module	Cells	Edges	Module	Cells	Edges
s298	133	404	s1494	653	2057
s444	202	595	s5378	2958	7527
s641	398	974	s9234	5825	14251
s1238	526	1602	s13207	8620	21122
s1423	731	2042	s15850	10369	25207
s1488	659	2057	s35932	17793	49517

based (Figure 7) and cell based. In the former, region centers are the sink locations for inter-region nets, while, in the latter, cell locations are used. An advantage of using region centers is the potential reduction in the number of sinks. However, this is accompanied by a loss of detail, as certain regions may not get intersected. This effect gets reduced at finer levels of partitioning until the region centers correspond very closely to the cell locations. In the current implementation routes are initially performed to cell locations. At later stages, a switch to region mode occurs, thus maintaining accuracy and efficiency.

5.2 Results

The congestion-driven algorithm, implemented in C, was incorporated into an existing quadratic placer provided by Cascade Design Automation. The numerical routines for sparse matrix operations [12] were obtained through use of the Meshach library [4]. A subset of netlists from the ISCAS-89 benchmark (Table 1), were placed and subsequently analyzed.

We express global flow uniformity as the average supply-demand excess over all edges, and local flow uniformity using standard deviation. When comparing two placements, a lower average reflects lower global congestion and a tighter standard deviation is representative of greater uniformity. The goal of a routing resource driven placement algorithm should be to minimize the global, local and maximum flow excess. The final analysis is performed on a uniform grid. The demand-supply excess, Δ_{DS}, is computed for each grid node and is processed to determine the global maximum, $max\{\Delta_{DS}\}$, average, $\overline{\Delta_{DS}}$ and standard deviation, $\sigma(\Delta_{DS})$. Table 2 lists the improvements in congestion, as measured by $max\{\Delta_{DS}\}$, $\overline{\Delta_{DS}}$ and $\sigma(\Delta_{DS})$ over the benchmark set. With larger designs such as s15850 and s35932, the local reductions in congestion tend to have less effect on the global average and uniformity. $wlen$ is the total route length as determined by the A* global router which also creates the internal routes during the final analysis.

Table 2: Improvement over quadratic placement

Benchmark	$wlen$	$max\{\Delta_{DS}\}$	$\overline{\Delta_{DS}}$	$\sigma(\Delta_{DS})$
s298	0%	0.5%	2.2%	3.7%
s444	4%	14.9%	26.5%	17%
s641	6%	5.0%	21.8%	15.6%
s1238	2%	9.1%	18.7%	15.6%
s1423	2%	15.7%	6.6%	12.4%
s1488	2%	8.5%	11.1%	-1.9%
s1494	2%	7.8%	-6.0%	6.8%
s5378	5%	8.1%	8.3%	3.0%
s9234	2%	2.4%	2.5%	4.1%
s13207	4%	-8%	7%	3.0%
s15850	-1%	5.8%	0.4%	-0.8%
s35932	0%	12.8%	-0.1%	4.4%

Table 3: Improvement due to relaxed pins

Module	Quadratic placement		Linear Placement	
	$wlen_q(\mu)$	$area_q(mils)$	$wlen_l(\mu)$	$area_l(mils)$
s298	0.8%	-8.4%	9.3%	6.5%
s641	2.2%	-0.2%	10.4%	3.1%
s1238	7.1%	5.7%	-3.7%	-8.8%
s1488	3.6%	2.5%	1.9%	2.3%
s1494	2.1%	2.8%	1.9%	2.4%
s5378	7.2%	4.2%	3.2%	-1.1%
s9238	7.0%	4.5%	3.9%	-2.4%

Table 3 shows the effectiveness of relaxed pins in wire length reduction for a sample set of small to medium sized netlists. The effect of the enhancement on both quadratic and linearized quadratic placement is shown. The wirelength and area reductions were obtained after detailed routing using a router provided by Cascade Design Automation.

6. CONCLUSION

This paper presents a methodology for integrating congestion into quadratic placement. The inclusion of a growth matrix into the quadratic placement formulation permits the use of a routing model to resolve congestion while concurrently minimizing wirelength. Further reduction in congestion, reflected in wirelength and area reduction, is obtained by the relaxation of pin constraints. This implementation with an A* search router and line-probe heuristics showed up to 20% reductions in average demand-supply excess. Ongoing experiments have shown it capable of improving linearized quadratic placement, but a more accurate model of routing resources is required. The methodology holds promise for greater improvements in congestion, with the possibility of including crosstalk into the routing cost.

7. REFERENCES

[1] C. L. E. Cheng, "RISA: accurate and efficient placement routability modeling". *1994 IEEE/ACM ICCAD*, pp. 690-5.

[2] D. K. Boese et. al, "High-performance routing trees with identified critical sinks". *30th DAC, 1993*, pp 182-7.

[3] D. W. Hightower, "A solution to Line Routing Problems on the Continuous Plane", *6th Design Automation Workshop* 1969, pp. 172-174.

[4] E. S. David et. al, "Meshach: Matrix Computations in C". *Proceedings of the Center for Mathematics and its Applications*. Australian National University, vol. 32, 1994.

[5] G. H. Golub et. al, "Matrix computations", The Johns Hopkins University Press, 2nd ed.

[6] G. Sigl et. al, "Analytical placement: a linear or a quadratic objective function?". *28th ACM/IEEE DAC*, pp. 427-432.

[7] J. M. Kleinhans et. al, "GORDIAN: VLSI placement by quadratic programming and slicing optimization". *1991 IEEE TCADICS*, vol.10, no.3, pp. 356-5.

[8] P. R. Suaris et. al, "A quadrisection-based combined place and route scheme for standard cells". *1989 IEEE TCADICS*, vol.8, no.3, pp. 234-244.

[9] R. S. Tsay et. al, "PROUD: A Fast sea-of-gates placement algorithm". *25th DAC 1988*, pp 318-323.

[10] S. Mayrhofer et. al, "Congestion-driven placement using a new multi-partitioning heuristic". *IEEE ICCAD* 90, pp. 332-5.

[11] S. Prasitjutrakul et. al, "A timing-driven global router for custom chip design". *1990 IEEE ICCAD*, pp. 48-51.

[12] S. Yousef, "Iterative methods for sparse linear systems". PWS Publishing Company.

[13] W. Dongsheng et. al, "Performance-driven interconnect global routing". *Proc. 6th Great Lakes Symp. on VLSI*, pp. 132-6.

[14] Y. Saab, "A fast clustering-based min-cut placement algorithm with simulated-annealing performance". *1996 VLSI Design*, vol.5, no.1, pp. 37-48.