Force Directed Mongrel with Physical Net Constraints

Sung-Woo Hur
Donga University

Tung Cao Karthik Rajagopal Yegna Parasuram
Amit Chowdhary Vladimir Tiourin
Intel Corporation

Bill Halpin
Syracuse Univ.
and
Intel Corporation

ABSTRACT

This paper describes a new force directed global placement algorithm that exploits and extends techniques from two leading placers, Force-directed [12] [26] and Mongrel [22]. It combines the strengths of force directed global placement with Mongrel's cell congestion removal to significantly improve the quality of placement during the difficult overlap removal stage of global placement. This is accomplished by using the spreading force in [12] to direct and control Mongrel's ripple move optimization. This new placer is called Force Directed Mongrel (FD-Mongrel). FD-Mongrel also incorporates physical net constraints [26], and improves the congestion model for sparse placements. We propose a new placement flow that uses a limited number of the spreading iterations of [12] to form a preliminary global placement. We then use the new FD-Mongrel described in this paper to remove cell overlaps, while meeting net constraints and optimizing wirelength. We present results on wirelength as well as timing driven placement flows.

Categories and Subject Descriptors

B.7.2 [**Hardware, Integrated Circuits, Design Aids**]: placement and routing

General Terms

Algorithms, Design.

Keywords

Timing Driven Placement, Force Directed Placement, Net Constraints, Mongrel

1. INTRODUCTION

Automated cell placement has always been an important step in the fast and efficient design of VLSI circuits. Cell placement has a big impact on the key design parameters – wirelength, performance and routability of the design. With ongoing advances in the semiconductor process, circuit performance is becoming more dependent on the wire delays since wire delays do not reduce as rapidly as gate delays [4] [17]. Thus, there is a greater need to design efficient timing-driven placement algorithms for high speed interconnect dominated designs. Furthermore, the scaling down of semiconductor process allows larger designs and the importance of the placement step grows with design size [25].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC '03, June 2-6, 2003, Anaheim, California,USA
Copyright 2003 ACM 1-58113-688-9/03/0006…$5.00.

Automated cell placement has been the subject of much research [1][2][7-12][14-16][19-26]. One of the most powerful techniques for cell placement is the force-directed method for global placement by Eisenmann [12], called Kraftwerk. Kraftwerk uses forces derived from the cell congestion to remove cell overlaps during placement. The main advantage of the force-directed method is that it is an iterative approach that models the wirelength, cell congestion and timing in the same mathematical formulation. This allows smooth and simultaneous optimization of design in terms of these three parameters. Wire length and cell congestion are modeled as forces, while timing is modeled as higher net weights on timing-critical nets [12]. The forces for wirelength and cell congestion should be precisely weighted to result in an optimized design with little or no congestion. The weight on the spreading force which models cell congestion is initially small compared to the wirelength force, but is increased with every iteration to spread out the cells.

Timing driven Kraftwerk has been recently improved by a more precise modeling of timing in KraftwerkNC [26]. KraftwerkNC models timing in terms of net constraints. A net constraint is an upper bound on the half-perimeter of the smallest rectangle that encloses all the nets' connections. Net constraints are set on the critical nets by analyzing the timing of the design. KraftwerkNC generates good global placements with optimized wirelength and timing, but it does have a few drawbacks. KraftwerkNC is very sensitive to the rate of increase of weight on the spreading force. If the weight on spreading force is increased rapidly, then the placement converges faster at the expense of wire length and timing. Therefore, the weight on spreading forces is increased slowly with every iteration. After the initial spreading of KraftwerkNC, there is little cell movement due to the equilibrium between the weight on spreading force and the wirelength force, even though there is localized congestion across the design. KraftwerkNC will ultimately remove most of the congestion by increasing the contribution of the spreading force at the potential cost of wirelength, timing and long runtimes. Also, the modeling of spreading forces on the cells on the boundary of the design is not accurate, which might lead to congestion on the boundary. The mathematical formulation in KraftwerkNC represents a net as a clique, which is a collection of edges connecting every cell pair in the net. Thus, the net length is not modeled accurately in terms of the half-perimeter of the bounding box of all cells connected to the net. This deficiency has more impact during the later Kraftwerk iterations.

Mongrel [22], in contrast, is a set of hybrid techniques for cell placement. It starts by assigning cells to global bins in a grid imposed over the placement area. It then extracts a sub-circuit from the circuit, assigns new positions to the cells in the sub-circuit regardless of overlaps and then removes the overlaps using a novel scheme based on ripple moves. The ripple move takes cells from the most congested bin to the least utilized bin along the monotone path that results in the lowest wirelength. Mongrel then uses optimal interleaving, as a detailed placement step, to improve the linear ordering of cells in each row. Strengths of Mongrel are the efficient techniques of ripple moves to remove overlaps and optimal interleaving within each row, and the accurate modeling of the net length as the half-perimeter of the net's bounding box.

2. MOTIVATION

The key strengths in Mongrel precisely complement the drawback of resolving fine-grained congestion in the later stages of KraftwerkNC. Local congestion can be efficiently removed by the Mongrel technique based on ripple moves. Modeling of the half perimeter bounding box of a net is more accurate in Mongrel, compared to Kraftwerk, which would result in improved final wirelength of the placed design. We propose a placement approach based on timing-driven KraftwerkNC (based on net constraints) followed by improvement techniques from Mongrel. The idea is to use Kraftwerk to generate a global placement that has only some fine-grained congestion and then use new techniques motivated by Mongrel to resolve the fine-grained overlaps with minimum perturbation of cells and minimum impact on wirelength and timing.

There are two key problems with the placements generated by KraftwerkNC followed by Mongrel (with no modifications to Mongrel). First, running Mongrel on KraftwerkNC-generated placements would result in significant perturbation of the KraftwerkNC placement, because Kraftwerk and Mongrel use different schemes to resolve congestion. We propose a force-directed approach to Mongrel, called FD-Mongrel, that determines the forces based on cell congestion (like Kraftwerk) and uses these spreading forces to direct the ripple-movement for removing congestion. We modify this technique to flexibly choose the target bin in order to minimize the cell perturbation. Secondly, running Mongrel after KraftwerkNC would degrade the timing because Mongrel ignores net constraints. We extend FD-Mongrel to meet the net constraints, so that it obeys the same net constraints as KrafwerkNC resulting in improved final timing.

We propose a force-directed Mongrel approach that uses two key ideas from timing-driven KraftwerkNC – spreading forces for resolving congestion and net constraints to meet timing. Since KraftwerkNC and FD-Mongrel are both based on spreading forces and net constraints, our approach of KraftwerkNC followed by FD-Mongrel would result in legalized placements with optimized wirelength and timing.

3. OVERVIEW OF KRAFTWERK

Kraftwerk models the circuit as a graph with cells as vertices and nets as sets of edges. A net connecting *k* cells is modeled as a clique of size *k*. In the quadratic placement problem, the cost of a net is the sum of the cost of all its edges, where the cost of an edge is modeled as the squared distance between the two vertices (cells) of the edge. The overall objective function is to minimize the sum of the cost of all nets. In matrix notation, the objective function is given below in terms of a *2n*-dimensional placement vector *p*, where *n* is the number of vertices.

$$Objective = \frac{1}{2}\vec{p}^{\,T}.C.\vec{p} + \vec{d}^{\,T}.\vec{p} + const$$

$$where, \vec{p} = (x_1,\ldots,x_n,y_1,\ldots,y_n)^T$$

Here, the x and y locations of vertex i are denoted by x_i and y_i, respectively. The cost of an edge between two movable cells i and j is given by $(x_i-x_j)^2 = x_i^2 - 2.x_i.x_j + x_j^2$. The quadratic objective function is optimized by solving the following system of linear equations.

$$C.\vec{p} + \vec{d} = 0$$

Kraftwerk modifies the above formulation by including an additional force vector $\vec{e}$ that is derived from the cell density distribution in the placement area. The force vector e is used to remove cell overlaps by moving cells from areas of high cell density to areas of low cell density.

$$C.\vec{p} + \vec{d} + \vec{e} = 0$$

During early spreading iterations, the weight given to spreading forces is relatively small as compared to wirelength. The forces computed during early iterations are very useful, as they provide a global picture of how the design should spread to achieve general overlap removal. However, as the iterations proceed, the weights on spreading forces grow to converge the solution and to avoid oscillations. The drawback of having stronger spreading forces during later iterations is that the objective of wirelength minimization is compromised.

4. OVERVIEW OF MONGREL

Mongrel is a collection of optimization techniques, which includes Relaxation Based Local Search (RBLS), FM partitioning and optimal interleaving. It is divided into two phases: global placement and detailed placement. In global placement phase, the layout area is divided into grids. Each grid has the height of one row site and the width such that it can accommodate a few standard cells. In each iteration of RBLS, a random sub circuit is extracted and solved for its optimal placement. A legalized placement is maintained in RBLS through ripple movement of cells from the most over-congested bin to the least congested bin. In the final part of each RBLS iteration, FM partitioning is applied on adjacent bins. In the detailed placement stage, optimal interleaving is applied to further improve the wire length

5. PROBLEM STATEMENT

The problem of refining a global placement falls in the class of detail placement and legalization. Given a netlist, an initial global placement and a fixed placement area, the goal is to generate a legalized placement. Sub-goals of detail placement are to meet net constraints on critical nets, minimize the total wire length and perturb the initial placement as little as possible.

6. PROPOSED APPROACH

6.1 Notation

The following notation is used in the algorithm description:

$B_{r,c}$: bin (r, c) in the fine grid indexed by row and column.

d(S) : density of bin S, i.e., sum of cell area assigned to S divided by the capacity of S,

D_{th}: density threshold target. The goal of the FD-Mongrel is to have the density of every bin less than D_{th}

6.2 Force Creation

For the force at any point in the layout area, we adopt the four requirements for the additional forces in [12]. They are restated here for reference purposes:

1. The force on any cell depends only on the coordinates of that cell.
2. Regions with higher density are the sources of the forces and region with lower density are the sinks.
3. The forces do not form circles.
4. The force should be zero in infinity.

The mathematical formulation can be found in [12]. The force at any point as derived in [12] is:

$$\vec{f}(x,y) = \frac{k}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} D(x',y')\frac{\vec{r}-\vec{r}'}{|\vec{r}-\vec{r}'|^2}dx'dy'$$

As it will be discussed in the next section, we use a coarse force grid in FD-Mongrel. The force vector is computed at the center of each coarse grid bin.

6.3 Description of FD-Mongrel

1. *procedure* **FD-Mongrel**
2. *input:* global placement P, density threshold value D_{th}
3. *output:* new global placement P'
4. *begin*
5. *while* (there is a bin B such that $d(B) > D_{th}$ w.r.t. P) {
6. Determine force for each bin in the coarse grid
7. P ← **resolve-congestion**(P, D_{th})
8. }
9. *for* (each bin S that has an over-congested fine bin) {
10. **move-cells**(S, S) // move cells within the bin S
11. }
12. *return* new placement
13. *end*

The goal of FD-Mongrel is to direct the powerful optimization global search method from Mongrel with the force concept from Kraftwerk. In order to combine these two approaches, a coarse force grid is created using the force method described in the previous section. A second fine grid, similar to the grid in the original Mongrel, is used in the ripple movement. Each bin of the force grid corresponds to several bins of the fine grid. The coarse grid is used to direct and control optimizations on the fine grid. Similar to Kraftwerk, FD-Mongrel uses an iterative force approach. In each iteration, the forces are updated based on the current cell congestion. Each bin in the force grid is assigned a force vector based on the force at the center of the bin.

In **resolve-congestion** the force bins that exceed D_{th} are sorted according to their density. Of the unlocked bins, the algorithm selects the most congested bin, S_{force}, for overlap removal. To

1. *procedure* **resolve-congestion** (P, D_{th})
2. *input:* global placement P, density threshold value D_{th}
3. *output:* new global placement P'
4. *begin*
5. Unlock every bin B in the force grid
6. *while* (there is a bin B such that $d(B) > D_{th}$ and B is unlocked) {
7. S_{force} ← the most congested bin (source) among unlocked bins
8. T_{force} ← target (neighbor) bin based on the force vector of S_{force}
9. *while* ($d(S_{force}) > d(T_{force})$ && $d(S_{force})>D_{th}$) {
10. **move-cells**(S_{force}, T_{force}) // move some cells from source to target bin
11. }
12. Lock the source bin S_{force}
13. }
14. *return* new placement
15. *end*

avoid infinite loops, once a source bin is considered, it is locked for the remainder of the call to **resolve-congestion**. Based on the force direction at S_{force}, an adjacent bin, T_{force} is selected. Given, $S_{force} = b_{r,c}$, T_{force} is determined based on the direction of the force vector among 8 neighbor bins, $b_{r+1,c}$, $b_{r+1,c+1}$, $b_{r,c+1}$, $b_{r-1,c+1}$, $b_{r-1,c}$, $b_{r-1,c-1}$, $b_{r,c-1}$, and $b_{r+1,c-1}$. For example, if there is an upper right force at bin $b_{r,c}$, then T_{force} will be $b_{r-1,c+1}$ (In Mongrel, $b_{0,0}$ is the upperleft most bin). The choice of T_{force} is limited to an adjacent bin to limit the maximum ripple move distance. In each FD-Mongrel iteration, cells can move only to a neighboring bin on the force grid. Since there can be a sharp density gradient between S_{force} and T_{force}, the number of cells that can move from S_{force} to T_{force} is limited to 10% of the cells in S_{force} in step 9 of **resolve-congestion**. Step 9 also ensures that movement from S_{force} to T_{force} stops when $d(S_{force}) \leq D_{th}$ and that T_{force} does not become more dense than S_{force}. Together, these limits on cell movement within each iteration of FD-Mongrel reduce the inaccuracy of the force information and provide an incremental smooth spreading.

In the original Mongrel, cells are forced to move from the most congested fine bin to the least congested fine bin. These bins are often very far apart from each other. Since the fine bins are extremely small, with on average only 2-3 cells, the congestion gradient does not generally follow the path picked by the original Mongrel. Hence, the perturbation from the original placement can be significant. This spreading is not smooth as the monotone paths may cross each other depending on which source and destination bins are chosen. If the input global placement is already obtained considering timing critical nets, the perturbation can severely degrade the timing.

```
1.  procedure move-cells (S_force, T_force)
2.  input: source and destination bins S_force, T_force in the force grid
3.  begin
4.      S_fine ← fine bin with max density in S_force
5.      T_fine ← fine bin with min density in T_force
6.      G ← construct-gain-graph(S_fine, T_fine)
7.      Since G is DAG apply topological sort to determine max-gain monotone path P
8.      for (each edge E(B_origin, B_end) of P in order from S_fine to T_fine) {
9.          C ← cell in B_origin with gain g(C) = g(B_origin)
10.         move cell C from B_origin to B_end
11.         if (d(B_end) < D_th) break
12.     }
13. end
```

The **move-cells** procedure performs the ripple move along the maximum gain monotone path. As with Mongrel, **move-cells** uses a fine grid. The large number of cells in each coarse grid bin makes it unsuitable for Mongrel's ripple move method. Also, without the fine grid, more legalization work would be required at the end of FD-Mongrel. **move-cells** selects the most congested fine bin within S_{force}. This bin is called S_{fine}. Similarly, the least congested bin in T_{force} is selected. This bin is called T_{fine}.

```
procedure construct-gain-graph (S, T)
input: source and destination bins S, T in the fine grid
output: gain graph G as DAG
begin
    G ← Ø
    // bins in bounding box of S, T are nodes
    for (each fine bin B between S and T) {
        add node B to gain graph G
        for (each candidate cell C in bin B) {
            gain g(C, X/Y) ← wire-length reduction, when C moves towards T along X/Y
        }
        gain g(B, X/Y) ← MAX(g(C,X/Y) | all C in B)
        //bin B has at most 2 neighboring bins in X/Y direction towards T
        add edge to gain graph G as an out arc along X/Y with gain cost g(B, X/Y)
    }
    return G
end
```

The monotone path from S_{fine} to T_{fine} is computed by **construct-gain-graph** [22] with some enhancements. Since the goal in FD-Mongrel is to achieve a legal placement with minimum perturbation of the original placement, the ripple-move operation from S_{fine} to T_{fine} is modified. While ripple-moving a cell from S_{fine} to T_{fine} along the max-gain path, the rippling process stops if any fine bin's density (after a cell moves into it) is less than the threshold value D_{th}. Stopping as early as possible in the middle of rippling process significantly reduces perturbation of the original placement and reduces the total wire length since it does not artificially enforce an even cell spreading.

In step 9 of **move-cells**, we pick the cell with maximum gain from the source bin B_{origin} to move to the target bin B_{end}. If the cell with maximum gain is attached to a net with net constraints and moving this cell would degrade the quality, the algorithm skips this cell and picks the next cell with maximum gain. By not degrading net constraints, FD-Mongrel ensures that the design's timing does not deteriorate during global placement refinement.

Each iteration of FD-Mongrel smoothly reduces over-congestion. Iterations repeat as long as there is a force bin, B, with $d(B) > D_{th}$. After FD-Mongrel iterations are completed, cells are well distributed with respect to the force grid. However, there may be some over-congested fine bins within a force bin. In steps 9 and 10 of FD-Mongrel, any intra force bin over-congestion is removed using the idea of ripple-move along the max-gain path within the same force bin.

7. EXPERIMENTAL RESULTS

We have implemented FD-Mongrel algorithm in C++ on LINUX. Instead of using MCNC benchmarks [27], we used circuits from a recent microprocessor, since the effect of net constraints on meeting timing is more accurately studied by using data from a recent manufacturing process using state of the art engines for parasitic estimation and timing analysis. For our experiments, we used a set of 6 circuits from a 1.5 GHz microprocessor designed on 0.18 micron process. The circuit sizes range from 3,374 to 6,223 cells. We used a static timing analysis engine that accurately estimates the delay and transition times across the cells and nets of the circuit. The values of resistance and capacitance per unit length for the interconnect were obtained from the 0.18 micron process.

The circuits testcase1 through testcase6 are listed in Table 1 with number of cells and nets. For every circuit, we set up two experiments: (a) KraftwerkNC for global placement as well as refinement, and (b) KraftwerkNC for global placement and FD-Mongrel for refinement. Both global placements were legalized using the same end case placer. In our experiments, timing was analyzed every ten iterations of KraftwerkNC and refined net constraints were applied.

Table 2 compares the final wirelength for the two runs. We got an average 7.88% improvement by using FD-Mongrel for global placement refinement. The reason for better wirelength is improved net modeling and more controlled final spreading in FD-Mongrel compared to KraftwerkNC.

Table 3 compares the slack of the most critical path WNS (worst negative slack) in the final placements from the two runs. We got an average 14.2% improvement in WNS, which is attributed to meeting and improving net constraints in FD-Mongrel and better wire length modeling.

Table 4 illustrates improvements in TNS. TNS is the sum of the slacks of all paths with negative margin. The TNS improvement from using FD-Mongrel is 21.25%. This can be attributed to reduction in total wirelength as well as improved meeting of net constraints by FD-Mongrel.

As shown in Table 5, we found that KraftwerkNC followed by FD-Mongrel improved the runtime by 24% on average. This is to be expected since KraftwerkNC spends a lot of time in resolving cell congestion in the final stages. FD-Mongrel achieves the same effect in a more efficient manner.

Table 1: Number of cells and nets in the designs

Design	Number of cells	Number of nets
testcase1	6223	7296
testcase2	6039	7081
testcase3	5010	5855
testcase4	4905	5735
testcase5	3399	4150
testcase6	3374	4122

Table 2: Comparison of final wirelength in microns

Design	KraftwerkNC	KraftwerkNC, FD-Mongrel	% Improvement
testcase1	850144	752203	11.52%
testcase2	841097	791976	5.84%
testcase3	666798	653781	1.95%
testcase4	593115	508404	14.28%
testcase5	423055	394214	6.82%
testcase6	527916	491762	6.85%
Average			7.88%

Table 3: Comparison of WNS (worst negative slack among all timing endpoints) in nanoseconds

Design	KraftwerkNC	KraftwerkNC, FD-Mongrel	% Improvement
testcase1	-0.255	-0.193	24.31%
testcase2	-0.165	-0.144	12.73%
testcase3	-0.267	-0.26	2.62%
testcase4	-1.434	-1.405	2.02%
testcase5	-0.314	-0.246	21.66%
testcase6	-0.119	-0.093	21.85%
Average			14.20%

Table 4: Comparison of TNS (total negative slack of all timing endpoints) in nanoseconds

Design	KraftwerkNC	KraftwerkNC, FD-Mongrel	% Improvement
testcase1	-36.546	-27.652	24.34%
testcase2	-16.453	-12.574	23.58%
testcase3	-71.939	-62.859	12.62%
testcase4	-101.981	-80.643	20.92%
testcase5	-28.224	-24.755	12.29%
testcase6	-6.982	-4.626	33.74%
Average			21.25

Table 5: Runtime Comparison (minutes)

Design	KraftwerkNC	KraftwerkNC, FD-Mongrel	% Improvement
testcase1	35	17	51.43%
testcase2	27	24	11.11%
testcase3	26	25	3.85%
testcase4	24	18	25.00%
testcase5	13	8	38.46%
testcase6	13	11	15.38%
Average			24.21%

8. CONCLUSIONS AND FUTURE WORK

We have presented a new placement approach, FD-Mongrel that improves on the techniques in Mongrel and KraftwerkNC. We propose a force-directed placement flow based on global placement by KraftwerkNC and refinement by FD-Mongrel. Our results show good improvements in important metrics measuring placement quality. Further work includes improved modeling of net constraints and researching FD-Mongrel's application for ECO placement after design changes such as buffering, resynthesis and sizing.

9. REFERENCES

[1] Michael A. B. Jackson, Arvind Srinivasan and E. S. Kuh, "A Fast Algorithm for Performance-Driven Placement", Digest of Technical Papers, ICCAD, pp. 328-331, Nov. 1990.

[2] Bernhard M. Riess and Gisela G. Ellelt, "SPEED: Fast and Efficient Timing Driven Placement", Proc. Intl Symp. of Circuits and Systems, pp. 377-380, 1995.

[3] J. Rubenstein , J. Paul Penfield and M. A. Horowitz, "Signal Delay in RC Tree Networks", IEEE Transactions on Computer-Aided Design, vol. 39, no. 11, pp. 825-840, 1992.

[4] ITRS Roadmap for Interconnect, Semiconductor Industry Association, http://public.itrs.net/Files/2001ITRS/Interconnect.pdf

[5] W. C. Elmore, “The transient response of Damped Linear network with particular regard to wideband amplifier”, Journal of Applied Physics, pp.55-63, 1948.

[6] Lawrence T Pillage and Ronald A Rohrer, “Asymptotic Waveform Evaluation for Timing Analysis”, IEEE Transactions on Computer-Aided Design, pp. 352-366, 1990.

[7] William Swartz and Carl Sechen, “Timing Driven Placement for Large Standard Cell Circuits”, DAC, pp. 211-215, 1995.

[8] Wern-Jieh and Carl Sechen, “Efficient and Effective Placement for Very Large Circuits”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp. 349-359, 1995.

[9] A. Srinivasan, A K. Chaudhary, E. S. Kuh, “RITUAL: Performance Driven Placement Algorithm for Small Cell ICs”, ICCAD, pp. 48-51, Nov. 1991.

[10] Jurgen M. Kleinhans, Georg Sigl, Frank M. Johannes, and Kurt Antreich, “GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization”, IEEE Transactions on CAD, Volume 10, No. 3, pp. 356-365, 1991.

[11] K. Doll, F. M. Johannes, and K.J. Antreich, “Iterative placement improvement by network flow methods”, IEEE Transactions on CAD, vol. 13, pp.1190-1200, Oct. 1994.

[12] H. Eisenmann and F. M. Johannes, “Generic Global Placement and Floorplanning”, ACM/IEEE DAC, 1998.

[13] J. Cong, “Timing models for Interconnects and Devices”, DAC, 1997.

[14] R.S. Tsay, “Timing-Driven Placement”, DAC, 1997.

[15] Shih-Lian Ou and Massoud Pedram, “Timing-driven Placement Based on Partitioning with Dynamic Cut-net Control”, DAC, 2000.

[16] Habib Youssef and Eugene Shragowitz, “Timing Constraints for Correct Performance”, ICCAD, 1990

[17] D. Sylvester and K. Keutzer, “Getting to the Bottom of Deep Submicron”, ICCAD, pp. 203-211, 1998.

[18] Jorge Nocedal and Stephen J. Wright, “Numerical Optimization”, Springer-Verlag, 1999.

[19] Bill Halpin, C.Y. Roger Chen, and Naresh Sehgal, “A Sensitivity Based Placer for Standard Cells”, GLS-VLSI, 1999.

[20] Ren-Song Tsay and Juergen Koehl, “An Analytic Net Weighting Approach for Performance Optimization in Circuit Placement”, DAC, pp. 620-624, 1991.

[21] X. Yang, B-K. Choi, and M. Sarrafzadeh, "A Standard-Cell Placement Tool for Designs with High Row Utilization", in International Conference on Computer Design, Sep. 2002.

[22] S. Hur and J. Lillis. ``Mongrel: Hybrid Techniques for Standard Cell Placement'', in International Conference on Computer-Aided Design, pp. 165-170, 2000.

[23] A. B. Kahng, S. Mantik and I. L. Markov, “Min-max Placement For Large-scale Timing Optimization”, in Proc. ACM/IEEE Intl. Symp. on Physical Design, pp. 143-148, Apr. 2002.

[24] Bill Halpin, C.Y. Roger Chen, and Naresh Sehgal, “Timing Driven Placement using Physical Net Constraints”, Design Automation Conference, pp. 780-783. IEEE/ACM, Jun. 2001.

[25] Yih-Chih Chou, Youn-Long Lin, “A performance-driven standard-cell placer based on a modified force-directed algorithm, ISPD, pp. 24-29, 2001.

[26] Tung Cao, Amit Chowdhary, Bill Halpin, Yegna Parasuram, Karthik Rajagopal, Tal Shaked, “Timing Driven Force Directed Placement with Physical Net Constraints”, to appear in ISPD, 2003.

[27] “www.cbl.ncsu.edu/benchmarks/layoutsynth92/”

10. AUTHOR CONTACT INFORMATION

Address for Intel Corporation authors (except Prof. Hur)
Intel Corporation MS SC12-606
2200 Mission College Blvd.
Santa Clara, CA 95052
USA

Karthik Rajagopal - Email: karthik.rajagopal@intel.com

Yegna Parasuram - Email: yegnashankar.parasuram@intel.com

Tung Cao - Email: tung.d.cao@intel.com

Amit Chowdhary - Email: amit.chowdhary@intel.com

Bill Halpin - Email: william.halpin@intel.com

Vladimir Tiourin - Email: vladimir.tiourin@intel.com

Prof. Sung-Woo Hur – Email: shur@ce.donga.ac.kr

Address: Donga University, Pusan, Korea