Congestion Estimation During Top-down Placement [*]

Xiaojian Yang   Ryan Kastner   Majid Sarrafzadeh

Computer Science Department
University of California at Los Angeles
Los Angeles, CA 90095

xjyang,kastner,majid@cs.ucla.edu

ABSTRACT

Congestion is one of the fundamental issues in VLSI physical design. In this paper, we propose two congestion estimation approaches for early placement stages. First, we theoretically analyze the peak congestion value of the design and experimentally validate the estimation approach. Second, we estimate regional congestion in the early top-down placement. This is done by combining the wirelength distribution model and inter-region wire estimation. Both approaches are based on the well known Rent's rule, which is previously used for wirelength estimation. This is the first attempt to predict congestion using Rent's rule. The estimation results are compared with the layout after placement and global routing. Experiments on large industry circuits show that the early congestion estimation based on Rent's rule is a promising approach.

1. INTRODUCTION

Minimizing the total routed wirelength is one of the fundamental goals in VLSI placement stage. In order to achieve such a challenging objective, a number of heuristics and objective functions were proposed in the past couple of decades. Half-perimeter wirelength has emerged as the most typical objective in placement because it adequately models the routed wirelength, especially for two-terminal and three-terminal nets. In general, it is believed that there is a positive correlation between half-perimeter wirelength and routed wirelength. Many successful placement tools are based on half-perimeter wirelength minimization [1, 2].

As VLSI circuits are growing in both size and complexity, not only the half-perimeter wirelength but also *congestion* need to be emphsized at the placement stage. Most global routers use congestion as one of the main objectives. However, the optimization performance is constrained because the cells are already fixed at this stage. A highly congested region in the placement often leads to routing detours around the region, in turn results in a larger routed wirelength. Congested areas can also downgrade the performance of global router and, in the worst case, create an unroutable placement in the *fix-die* regime [3].

Congestion can be modeled as the summation of linear [4] or quadratic [5] function of difference between *routing demand* and *routing resource*. Existing congestion reduction techniques include incorporating congestion into cost function of simulated annealing [5], combining a regional router into placement tool [6] and performing a post placement processing step [7]. While congestion reduction at late or post placement stage is empirically effective, congestion estimate achieved in early placement stage would be equally valuable. First, a congestion driven placement tool guided by early congestion information might be more powerful. Such a tool could use techniques like white space allocation to relieve layout congestion. Second, early congestion estimates could be utilized by combined logic and layout optimization to improve design convergence. For example, when logic designers are given a number of different netlists, they can estimate the congestion by running only several steps of placement. The netlists with bad estimated congestion will be discarded much earlier.

The main contribution of this paper is to estimate both peak congestion and congestion distribution at the early top-down placement stage. Specifically, we quantitatively estimate the maximum congestion prior to placement stage. Also we give a congestion distribution picture of the chip layout at coarse levels of hierarchical placement flow. Both estimates are made based on Rent's rule — a well known stochastic model for "real" circuits. While Rent's rule has been used for wirelength estimation for a long time, to the best of our knowledge, there is no published work on congestion estimation problem using the same basis. To evaluate our estimation approaches, the estimation results are compared with the real congestion which is extracted from the design after placement and global routing. It is generally believed that the global routing output correlates well with the final routing in industrial circuits [8]. Therefore we target estimates that match the real congestion map after global routing.

Congestion is a function of routing demand and routing resource. Once the technology feature and chip characteristics (die size, number of layers, position of pre-placed macros) are fixed, the routing resource is roughly determined[1]. Con-

[*]This work was partially supported by NSF under Grant #CCR-0090203

ISPD'01, April 1-4, 2001, Sonoma, California, USA.
Copyright 2001 ACM 1-58113-374-2/01/0004 ...$5.00.

[1] Accurate available routing resources can only be obtained

gestion and routing demand are so closely related that it is straightforward to convert one to the other. In this work, we will focus on the estimation for routing demand.

The rest of this paper is organized as follows: Section 2 briefly introduces the terms and definitions used in this paper. It also reviews Rent's rule, the fundamental theory upon which this work is based. Section 3 analyzes the peak congestion problem and gives a good estimate which is validated by experiments. Section 4 models the regional routing demand in a top-down placement context. Experiments in this section show the effectiveness of the proposed method. Section 5 gives the conclusion and the future work.

2. PRELIMINARIES

2.1 Placement, Routing and Congestion

For consistency we will use the following terms throughout the paper. A *circuit* is a hypergraph $G(V, E)$, where V is a set of cells and E is a set of hyperedges. A *hyperedge* $e \in E$ is a subset of V which contains two or more cells. A *placement* is a set of locations for all cells on a rectangular chip area.

A common top-down placement flow is based on min-cut placement, in which the circuit is recursively bi-partitioned into subcircuits. Meanwhile the layout area is also partitioned into *placement regions*, each of which contains a corresponding subcircuit.

During global routing, the chip is divided into an array of uniform rectangular *tiles*. The tile is small enough that each placement region covers an integral number of tiles. All the nets will be routed by connecting the cells of each net using grid wires. For each *boundary* of the tiles b, the *routing demand* $d(b)$ is the number of wires that cross this boundary; the *routing supply* $s(b)$ is the number of wires that are allowed to cross the boundary. The *overflow* of a boundary $c(b)$ is $max(d(b) - s(b), 0)$. The congestion of a placement region is the summation of the overflow over all the boundaries within this placement region. The *peak congestion* of a placement is the maximum overflow over all the tile boundaries.

2.2 Rent's rule

Rent's rule is an empirical observation first described by Landman and Russo [9]. It states the relationship between the number of elementary blocks B in a subcircuit of a partitioned design, and the number of external connections P of the subcircuit. Specifically,

$$P = T_b B^r \tag{1}$$

where T_b is the average number of interconnections per block, and r is the Rent exponent ($0.4 < r < 0.8$ in real circuits). The Rent exponent r can be computed by plotting the P versus B relation in a log-log diagram for every value of B in a top-down partitioning process, and then fitting a line on the plotted points. The slope r of this line represents the Rent exponent.

Rent's rule has been widely used to estimate interconnect wirelength [10, 11, 12]. In general, a higher Rent exponent will result in a longer average wirelength, which in turn implies a larger wiring area and more congested layout.

3. PEAK CONGESTION ANALYSIS

3.1 Cut Ratio in Recursive Bi-partitioning

In order to analyze peak congestion over all the tile boundaries of the layout, we assume that the circuit is an *ideal* circuit which is *self-similar* [2] and strictly obeys the Rent's rule. This ideal circuit is placed using a hierarchical placement flow which is based on recursively bipartitioning. On every hierarchical level of the top-down placement, each subcircuit is quadrisectioned into four smaller subcircuits. A quadrisection step consists of a vertical bi-partitioning followed by a horizontal one. The net cut result of the first bi-partitioning by a vertical cut line is C_1. The net cuts of the second horizontal bi-partitioning are net cut $C_{2,1}$ and $C_{2,2}$, and so on(Figure 1(a)). Since the circuit is self-similar,

$$C_{i,1} = C_{i,2} = \ldots = C_i \qquad i = 1, \ldots 2H$$

where H is the number of hierarchical levels in the top-down recursive bi-partitioning placement.

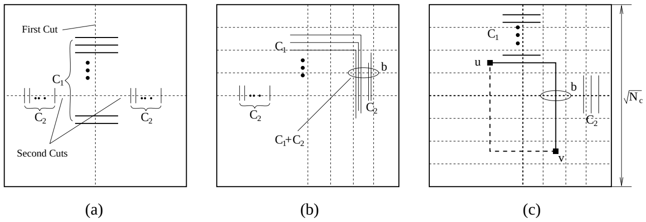

Figure 1: Relationship between net cut and congestion. (a) recursive bi-partitioning and cut. (b) worst case routing demand analysis. (c) average case routing demand analysis.

THEOREM 1. *In a recursive bi-partitioning approach on an ideal circuit, the ratio between the net cut of the (i+1)th bi-partitioning C_{i+1} and the net cut of the ith bi-partitioning C_i is 2^{-r}.*

Proof: Consider the subcircuits to be bi-partitioned at each hierarchical level. If we denote the size of subcircuits in the *ith* bi-partitioning B_i, then the size of subcircuit in the *(i+1)th* bi-partitioning is $B_{i+1} = B_i/2$. Since the circuit is self-similar, all the subcircuits have the same Rent exponent r and Rent coefficient T_b. According to equation 1, the bi-partitioning results of the *ith* and *(i+1)th*, which are the external number of interconnects for the partitioned subcircuits, can be represented as $T_b(B_i/2)^r$ and $T_b(B_{i+1}/2)^r$. We obtain

$$\frac{C_{i+1}}{C_i} = \frac{T_b(B_i/2)^r}{T_b(B_{i+1}/2)^r} = \frac{T_b(B_i/2)^r}{T_b(B_i/4)^r} = 2^{-r} \qquad \blacksquare$$

3.2 Worst Case Analysis

The top-down placement terminates at the $H = log_4 N_c$ level where N_c is the number of cells of the circuit. In the final placement each cell occupies one tile, which has unit width and height. The global routing uses L-shape routing model, in which a net is routed using either the upper or the lower part of the bounding box of this net. This is not

after placement and global routing with the consideration of the layer area occupied by placed cells and the number of routing layers. However a main portion of routing resources could be predicted at this point.

[2] A circuit is self-similar if its subcircuits at any hierarchical level present similar characteristics.

a good routing method but it gives a general picture of net distribution.

Now we want to find out the maximum routing demand over the tile boundaries without placing the circuit. First we discuss the worst case. Let us denote the maximum routing demand of a tile boundary as C_{max}.

THEOREM 2. *In a recursive bi-partitioning approach on an ideal circuit, the maximum routing demand over all the tile boundaries* $C_{max} < C_1\frac{1-\alpha^{2H}}{1-\alpha}$, *where* C_1 *is the net cut of the first bi-partitioning and* $\alpha = C_{i+1}/C_i$ *is the ratio between net cuts of two consecutive partitioning operations.*

Proof: In figure 1(b), the circuit is partitioned into two parts with a net cut C_1. This means that there are C_1 nets crossing from left to right, or vice versa. Let us look at a tile boundary located at the right half. In the worst case, all these C_1 nets pass this specific boundary. Hence the first bi-partitioning contributes C_1 to the routing demand of this boundary. Similarly, the i_{th} bi-partitioning contributes C_i to the routing demand. Thus, for any boundary, the upper bound of the maximum routing demand is

$$\sum_{i=1}^{2H} C_i = C_1 \sum_{i=0}^{2H-1} \alpha^i = C_1\frac{1-\alpha^{2H}}{1-\alpha} \qquad \blacksquare$$

3.3 Uniform Distribution of Cut Nets

In the previous discussion we assume that all the nets which are cut in a bi-partitioning cross a particular tile boundary. That is, obviously, not the general case. However, once we construct a framework like the model in the previous subsection, we can study the congestion behavior using different cut net distribution models.

We continue the analysis using a *uniform distribution model*, in which the cut nets of a bi-partitioning are uniformly distributed over all the subcircuit area. In other words, the cells in the partitioned subcircuit have equal probabilities to be connected to a cut net.

THEOREM 3. *In a recursive bi-partitioning approach on an ideal circuit, assuming cut nets are uniformly distributed, the maximum routing demand over all the tile boundaries*

$$C_{max} < \frac{C_1}{\sqrt{N_c}}(\frac{1}{2}+2\alpha)\frac{\sqrt{N_c}\alpha^{2H}-1}{2\alpha^2-1}$$

where C_1 *is the net cut of the first bi-partitioning and* $\alpha = C_{i+1}/C_i$ *is the ratio between net cuts of two consecutive partitioning operations.*

Proof: The proof is similar to that of Theorem 2, and is omitted due to space constraints.

3.4 Experimental Validation

Theorem 3 gives a much tighter upper bound of peak routing demand for a circuit. If the Rent exponent of a circuit is known, we can estimate the peak routing demand prior to placement stage. The following experiments evaluate the effectiveness of the estimation method.

Given a circuit, we first compute the Rent exponent using the partition-tree method proposed in [13]. Specifically, we record the number of cells and number of external nets for each partition while recursively partitioning the circuit. Then we do a linear regression on these data points plotted on a log-log diagram. The slope of the linear regression result is Rent exponent r. By partitioning we also obtain the net cuts of each partitioning $C_1, C_2, \ldots$,etc. Because of the existence of Region II of Rent's curve, the first several net cuts do not correlate well with Theorem 1. Therefore we use C_h/α^{h-1} instead of C_1 in experiments ($h = 8$ in our work). Now we can compute estimated maximum routing demand using Theorem 3.

The peak routing demand of the design is obtained by placing and global routing the circuit. The placement algorithm used in the experiments is a recursive bisection approach combined a multilevel partitioner hMetis [14]. It is a *global placement* since it stops at at a certain $m \times n$ level. Then a bounding box global router is used to route the global placement output. The peak routing demand is the maximum number of crossings over all vertical and horizontal tile boundaries.

ckt	*#cells*	*#nets*	r	$D_p^{'}$	D_p	*time*
ibm01	12,036	13,056	0.48	**30.3**	31	116
ibm02	19,062	19,291	0.51	**62.7**	67	208
ibm03	21,924	26,104	0.65	47.8	62	195
ibm04	26,346	31,328	0.62	**52.1**	52	218
ibm05	28,146	29,647	0.69	**89.1**	90	289
ibm06	32,185	34,935	0.57	82.3	60	305
ibm07	45,135	46,885	0.55	**86.8**	90	422
ibm08	50,977	49,228	0.55	111.9	100	511
ibm09	57,746	59,454	0.49	93.0	75	426
ibm10	67,692	72,760	0.47	135.8	112	669
ibm11	68,119	78,843	0.51	**53.9**	50	601
ibm12	69,026	75,157	0.52	**76.1**	76	771
ibm13	81,018	97,574	0.39	85.5	108	823
ibm14	147,088	147,605	0.46	**117.6**	111	1580

Table 1: Rent exponent r, predicted peak routing demand $D_p^{'}$ and real number D_p from global routing output. Runtime is in CPU seconds. Good estimates are listed in boldface.

Table 1 shows the comparison between the estimated peak routing demand $D_p^{'}$ and the real peak routing demand D_p. The test circuits are large industry designs selected from the IBM-PLACE benchmark suits [3], which are derived from ISPD98 benchmark [15]. Runtime is measured in seconds on a Sun workstation with a 400MHz CPU. It should be noted that the estimated peak demand is scaled by a factor $\sqrt{N_c/(mn)}$. This is because there are $N_c/(mn)$ cells in every tile, not one cell per tile as in the estimation model.

From Table 1 one can see that 8 out of 14 estimates are very close to real numbers. However, the estimation is not accurate for some circuits. There are a number of reasons for bad estimates. First, the uniform distribution model assumes that the probability of a cut net connecting to a cell is uniform. In the placement, however, the connected cells tend to be placed closer, causing the estimated value smaller than the real value. Second, we estimate the routing demand by summing up the number of nets crossing boundary b at each level. Each number is indeed a upper bound for all boundaries. The summation tends to be larger than the real value. Other reasons include the variation of cut ratio α, the chip area aspect ratio (which is not 1), . . . , etc.

[3] http://www.ece.nwu.edu/nucad/ibm-place.html

4. REGIONAL CONGESTION

In Section 3 we have discussed the peak congestion estimation problem. Another estimation requirement appears at early placement stages. In this section, we propose a routing demand estimation approach in the context of top-down placement.

For a given region r in a globally routed design, we use $D(r)$ to present the routing demand, which is the summation of the number of net crossings over all the tile edges within this region. In the top-down placement, estimating the routing demands for all the regions will give us a congestion map, which is valuable for early design evaluation.

Given a placement region, the nets which cause the edge crossings can be classified into two types: the *internal nets* which connect cells within this region and the *external nets* which span toward other regions or cross this region while connecting no cells. We use $ID(r)$ to denote the internal routing demand, which is the summation of the number of crossings caused by internal nets over all tile edges. Similarly, the external routing demand $ED(r)$ is the summation of the number of crossings caused by external nets for all tile edges. The total routing demand $D(r)$ for any region r is the sum of its internal routing demand $ID(r)$ and external routing demand $ED(r)$.

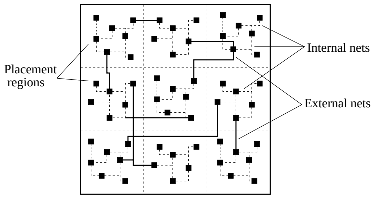

Figure 2: Internal and external routing demand

Figure 2 shows the concepts of internal routing demand and external routing demand at a top-down placement stage. The original circuit is divided into subcircuits and each subcircuit is assigned to a region. The dashed lines are internal nets. The thicker, solid lines represent external nets. The routing demand in a region consists two parts: net crossings caused by the internal nets and those by the external nets.

At the very coarse placement stage, the subcircuits are loosely coupled, i.e. the number of nets between partitioned subcircuits is much small than the number of internal nets. The routing demand of a region is primarily determined by the interconnect complexity of the subcircuit which belongs to the region. As the top-down placement flow goes into deeper levels, the routing demand of a placement region is determined by not only the internal complexity of the subcircuit in this region, but also the geometrical locations of other subcircuits and the interconnects between them.

4.1 Internal Routing Demand

In a typical top-down placement scheme, e.g., min-cut placement, the cells of a partitioned subcircuit will eventually be placed within the area that is assigned for this subcircuit. Therefore, estimating the internal routing demand becomes feasible. For a certain region in a top-down placement, the internal routing demand is proportional to the total routed wirelength after global routing [4] [4].

The wirelength estimation problem has been studied for many years. There are several successful estimation techniques based on Rent's rule: Donath's classical method [10], its extension [11] and more recent model [12]. In these methods the wirelength distribution of the entire design is predicted. However, in our work, we estimate wirelength for each subregion and take into account the locality of the Rent's rule. Since different subcircuits have different complexities and Rent parameters, wirelength estimation for subcircuits models the internal routing demand.

Both Donath's model and Davis's model are adopted in our routing demand estimation. Related results are evaluated in Section 4.4. When estimating total wirelength we are more concerned with the relative value (i.e., the comparable numbers for different regions) rather than the real wirelength results.

4.2 Rent Exponent Extraction

In order to estimate total wirelength of a region or the routing uemand, the Rent exponent needs to be extracted. A traditional way is using partitioning to get numbers of block size and external pins. Then a linear regression is performed when enough numbers are gathered. To make the Rent exponent extraction effective, a minimum number of partitioning is expected.

The part of Rent's rule curve in Region I implies true Rent exponent. Thus locating the data points becomes a key factor in this method. The following $DREE$ algorithm gradually increases the number of partitions when it partitions each subcircuit, and then performs linear regression on the latest N ($N = 4$ in our approach) data points. For each linear regression, we compute χ^2-probability Q which indicates the goodness-of-fit:

$$Q = \Gamma_q(\frac{N-2}{2}, \frac{\chi^2}{2}) \qquad (2)$$

where N is the number of fitting data points and $\Gamma_q(a, x)$ is the incomplete gamma function.

Once the quality Q of linear regression is greater than a threshold value (0.9 in our experiments) for each subcircuit, we claim these regression points are in Rent's rule region I. The algorithm terminates and outputs Rent exponent for each subcircuit.

The total running time cost of $DREE$ algorithm is dominated by running time of recursive bi-partitioning on the circuit, which is very fast due to the recent advances of multi-level partitioning techniques. The method of extracting Rent exponent is similar to the classical approach (e.g. [13]). The algorithm is *dynamic* in the sense that unnecessary partitionings are not performed once a good linear regression is obtained.

4.3 External Routing Demand

Internal routing demand can be estimated using Rent parameters, while estimating external routing demand requires the knowledge of interconnection between regions. Due to the locality of Rent's rule in large design, we can not assume a uniform number of interconnects between regions. However, the interconnect distribution of the current placement is known. For each region, we compute the external routing demand based on the interconnects which connect or pass through this region. The following figure shows a simple example of how to estimate the external routing demand

[4] We assume that the global tile is square.

Algorithm 1 DREE(Dynamic Rent Exponent Extraction)

```
Input: $n$ subcircuits $G_k$=($V_k$, $E_k$) $k = 1, ..., n$
Output: Rent exponent $r_k$ of each subcircuit $G_k$, $k = 1, ..., n$.
$k \leftarrow 0$, $m \leftarrow$ number of data points for line fitting
repeat
  $k \leftarrow k + 1$
  for each subcircuit $G_i$ ($i = 1, ..., n$) do
    Do $2^k$ way partitioning on $G_i$
    Compute $B_{i,k} = |V(G_i)|/2^k$ (average number of modules per partition)
    $P_{i,k}$= average number of external nets per partition
    Record point $(x_{i,k}, y_{i,k}) = (log B_{i,k}, log P_{i,k})$
  end for
  if $k \geq m$ then
    for each subcircuit $G_i$, perform a linear regression on m-points data set: $(x_{i,k}, y_{i,k})$, $(x_{i,k-1}, y_{i,k-1})$, ... , $(x_{i,k-m+1}, y_{i,k-m+1})$
    get line fitting result equation $f_i(x) = a_i x + b_i$ and quality of fitting $Q_i$
  end if
until $Q_i \geq 0.9$ for $i = 1, \ldots, n$
$r_i = a_i, \quad i = 1, \ldots, n$
```

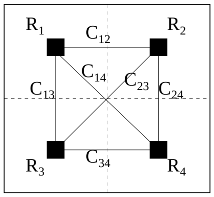

Figure 3: External routing demand analysis

In Figure 3 a design area is divided into four regular rectangle regions: R_1, R_2, R_3, R_4. Let the number of interconnects between R_i and R_j be C_{ij}. Assume that the distance between the center of adjacent regions is 1 unit. For region R_1, the wire which connects R_1 and other regions contributes one unit to its routing demand. A wire which may pass through R_1 (e.g. C_{23}) statistically contributes a half unit to R_1's routing demand because this wire has a 50% chance to be routed in this region. Then the estimate of external routing demand (ED) of region R_1 is:

$$ED_1 = C_{12} + C_{13} + C_{14} + \frac{1}{2}C_{23}$$

Similarly we can get external routing demand for R_2, R_3 and R_4.

In general, we assume that there are n regular rectangle regions. To estimate the external routing demand for a region, the interconnects between every pair of regions will be evaluated. Among them the ones which connect or pass the evaluated region are counted. Therefore the external routing demand estimate for region k is:

$$ED_k = \sum_{1 \leq i,j \leq n, i \neq j} C_{ij} \rho_{ij}(k)$$

where C_{ij} is the number of interconnects between region i and region j. $\rho_{ij}(k)$ is the *probability density function* which indicates the likelihood a wire from region i to region j passes a given region k. It can be calculated by equally assigning probability for interconnects passing from one region to its neighborhood regions.

4.4 Estimation Results

We conduct experiments to estimate the routing demand for every region in the coarse placement. By summing up the wirelength estimate and inter-region wire estimate, we obtain the routing demand of each region. This estimate will be compared with real routing demand map.

To get the real routing demand distribution, we continue the placement process from the point where we make estimation. The placement algorithm is a typical recursive bisection flow using a state-of-the-art multilevel partitioner hMetis [14]. After placement, a high-quality global router based on maze routing and rip-up and re-route is employed. Then we extract the routing demand, which is the number of net crossings on the edges of the global routing tiles within every region. Such a result is the real routing demand distribution which reflects the wire requirement on a design.

For better comparison, we scale both the estimated routing demand map and the real routing demand map. The scaling process is simply dividing every routing demand value associated with a region by the average routing demand value of the whole chip area. After scaling, for each region, we have a scaled estimated routing demand D_e and a scaled real routing demand D_r. The estimation error for region r is defined by $Error(r) = |D_e - D_r|/D_r \times 100\%$. The overall estimation error for the design is the average value of the estimation errors for all regions.

All the experiments have been done on a Sun Ultra10 workstation with a 400MHz CPU. Table 2 shows the overall estimation error for three different approaches. The first approach uses Donath's wirelength estimation model and routing estimation method proposed in Section 4.3. The second approach is similar to the first one. The only difference is that it uses Davis's wirelength distribution model. The third approach estimates regional congestion based on Davis's wirelength model only, without routing estimation. The three approaches use the same DREE algorithm to extract Rent exponent. Since the running time for external routing demand estimation can be ignored comparing with the Rent exponent extraction process, we only report the runtime for DREE algorithm.

The results in Table 2 show that the proposed approach is an effective way to estimate congestion. By combining either Donath's or Davis's wirelength model with the routing estimation method, we can predict relative congestion with small errors. The comparison between approaches with or without routing estimation shows that: in general, wirelength only can not estimate congestion well. There are some different cases for which wirelength itself produces good estimates. These cases are mostly in 2 × 2 placement level where the routing estimation is not as important as later levels.

It should be noted that for a given circuit, the estimation becomes harder as the number of regions increases. Benchmarks ibm03, ibm04 and ibm14 show the trend. A global router uses detours to avoid routing in a congested area. When regions are large, the detours for congested spots in a given region are still counted as the routing demand for this region. However, if regions are small, the detours contribute routing demand to neighboring regions, which makes estimation difficult. In general, estimates of large regions are more accurate than those of small regions. This suggests that designers use actual global routing to get congestion information at later top-down placement stages.

ckt	*level*	Donath's	Davis's	Davis's	*time*
		+ *RE*	+ *RE*	only	(sec_s)
ibm01	2×2	9.0%	10.5%	12.8%	100
	4×4	8.2%	11.8%	6.2%	155
	8×8	10.3%	11.2%	26.5%	266
ibm02	2×2	8.2%	11.8%	6.2%	170
	4×4	8.7%	9.8%	11.2%	267
	8×8	10.3%	11.6%	18.5%	463
ibm03	2×2	8.2%	7.0%	10.0%	153
	4×4	11.6%	14.2%	18.5%	256
	8×8	13.2%	13.6%	25.3%	491
ibm04	2×2	8.2%	6.5%	17.2%	184
	4×4	11.7%	12.0%	15.7%	297
	8×8	13.4%	14.0%	20.2%	592
ibm11	2×2	15.2%	9.0%	2.8%	527
	4×4	7.6%	7.9%	17.2%	745
	8×8	8.5%	9.0%	25.0%	1682
ibm12	2×2	11.5%	8.2%	9.5%	663
	4×4	5.8%	4.3%	8.4%	824
	8×8	8.2%	6.7%	11.2%	1845
ibm13	2×2	13.2%	7.2%	5.2%	689
	4×4	10.9%	11.1%	16.0%	845
	8×8	9.0%	9.0%	30.1%	1974
ibm14	2×2	8.5%	6.5%	6.8%	1713
	4×4	9.4%	7.1%	13.6%	1463
	8×8	10.9%	11.1%	19.5%	3362
average		10.0%	9.6%	14.7%	

Table 2: Estimation error and runtime comparison for three approaches: Donath's wirelength model combining routing estimation (RE), Davis's wirelength model combining routing estimation and Davis's wirelength estimation only. Runtime is in CPU seconds. Circuits statistics are the same as Table 1.

5. CONCLUSION

In this paper, we analyze peak congestion of a design based on Rent's rule. We also propose a regional congestion estimation method that takes both internal and external routing demand into account. Experiments on large industry benchmarks show that the peak congestion for L-shape routing model can be estimated. For the regional congestion, the estimation error is on the average less than 10% compared with the real congestion.

Our future research will focus on faster estimation techniques by including Rent exponent prediction methods. A study on circuit Rent exponents [16] is a step toward this objective. Furthermore, estimation models for various placement schemes (e.g. quadratic placement) will be studied.

6. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments on the manuscript.

7. REFERENCES

[1] C. Sechen and A. Sangiovanni-Vincentelli. "TimberWolf3.2: A New Standard Cell Placement and Global Routing Package". In *Design Automation Conference*, pages 432–439. IEEE/ACM, 1986.

[2] G. Sigl, K. Doll, and F. M. Johannes. "Analytical Placement: A Linear or a Quadratic Objective Function". In *Design Automation Conference*, pages 427–432. IEEE/ACM, 1991.

[3] A. E. Caldwell, A. B. Kahng, and I. L. Markov. "Can Recursive Bisection Alone Produce Routable Placements?". In *Design Automation Conference*, pages 153–158. IEEE/ACM, June 2000.

[4] M. Wang and M. Sarrafzadeh. "Behavior of Congestion Minimization During Placement". In *International Symposium on Physical Design*, pages 145–150. ACM, April 1999.

[5] C. E. Cheng. "RISA: Accurate and Efficient Placement Routability Modeling". In *International Conference on Computer-Aided Design*, pages 690–695, 1994.

[6] P. N. Parakh, R. B. Brown, and K. A. Sakallah. "Congestion Driven Quadratic Placement". In *Design Automation Conference*, pages 275–278. IEEE/ACM, June 1998.

[7] M. Wang, X. Yang, K. Eguro, and M. Sarrafzadeh. "Multi-Center Congestion Estimation and Minimization During Placement". In *International Symposium on Physical Design*, pages 147–152, 2000.

[8] L. Scheffer and E. Nequist. "Why Interconnect Prediction Doesn't Work". In *International Workshop on System-Level Interconnect Prediction*, pages 139–144. ACM, April 2000.

[9] B. Landman and R. Russo. " On a Pin Versus Block Relationship for Partitions of Logic Graphs ". *IEEE Transactions on Computers*, c-20:1469–1479, 1971.

[10] W. E. Donath. "Placement and Average Inter-connection Lengths of Computer Logic". *IEEE Transactions on Circuits and Systems*, 26(4):272–277, April 1979.

[11] D. Stroobandt and J. V. Campenhout. "Accurate Interconnection Length Estimations for Predictions Early in the Design Cycle". *VLSI Design, Special Issue on Physical Design in Deep Submicron*, 10(1):1–20, 1999.

[12] J. A. Davis, V. K. De, and J. Meindl. "A Stochastic Wire-Length Distribution for Gigascale Integration(GSI) - Part I: Derivation and Validation". *IEEE Transactions on Electron Devices*, 45(3):580–589, Mar 1998.

[13] L. Hagen, A. B. Kahng, F. J. Kurdahi, and C. Ramachandran. "On the Intrinsic Rent Parameter and Spectra-Based Partitioning Methodologies". *IEEE Transactions on Computer Aided Design*, 13(no.1):27–37, Jan 1994.

[14] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. "Multilevel Hypergraph Partitioning: Application in VLSI Domain". In *Design Automation Conference*, pages 526–529. IEEE/ACM, 1997.

[15] C. J. Alpert. "The ISPD98 Circuit Benchmark Suite". In *International Symposium on Physical Design*, pages 18–25. ACM, April 1998.

[16] X. Yang, E. Bozorgzadeh, and M. Sarrafzadeh. "Wirelength Estimation based on Rent Exponents of Partitioning and Placement". to appear in *Proc. International Workshop on System-Level Interconnect Prediction*, April 2001.