On Routing Demand and Congestion Estimation for FPGAs

Shankar Balachandran Parivallal Kannan Dinesh Bhatia
Center for Integrated Circuits and Systems
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
PO Box 830688, Richardson, TX 75083, USA
Email:{shankars,parik,dinesh}@utdallas.edu

Abstract

Interconnection planning is becoming an important design issue for ASICs and large FPGAs. As the technology shrinks and the design density increases, proper planning of routing resources becomes all the more important to ensure rapid and feasible design convergence. One of the most important issues for planning interconnection is the ability to predict the routability of a given placed design. This paper provides insight into the workings of recently proposed method by Lou et. al. [8] and compares it with our proposed methodology, fGREP [6]. We have implemented the two methods for a generic FPGA architecture and compare the performance, accuracy and usability of their estimates. We use the well known FPGA physical design suite VPR [2], as a common comparison tool. Our experiments show that fGREP produces far better routing estimates but at larger execution times than Lou's method. Insight into what makes the methods work and where they falter are also discussed in detail.

1. Introduction

Interconnect estimation is gaining more and more attention in the modern CAD flows. As designs become larger, interconnections become more complex to manage. Optimizing interconnections for wirelength solves just part of the problem. In order to achieve design closure in reasonable time, good congestion estimation techniques are required. These techniques need to identify potential "trouble spots" and provide the designers with possible alternatives. Accurate and fast congestion estimations can guide routing, which is traditionally a hard and very time consuming task.

FPGA routing is a particularly daunting task. The routing structures are rigid and hence a number of iterations are required for convergence. This is becoming more critical as commercial FPGAs grow in size and density. Commercial CAD tools spend majority of the time in routing. Estimates made in the earlier design stages can be very useful for making routing simpler. In this paper we provide insight into the workings of recently proposed estimation method by Lou et. al. [8] and compare it with our proposed methodology, fGREP [6]. We have implemented the two methods for a generic FPGA architecture and compare the performance, accuracy and usability of these estimates. We use the well known FPGA physical design suite VPR [2], as a common comparison tool. Our experiments show that fGREP is accurate and shows an average of 7% deviation from the actual detailed routes. Lou's method is away from the detailed results by an average of 76%. The later approach is faster than fGREP in computing the congestion and routing demands.

The paper is organized as follows. Previous works on routing estimation and motivation for our work are provided in Section 2. Section 3 deals with some preliminaries that are common to both the estimation methods. We describe both the estimation methods, complete with formulations, illustrations and assumptions in Sections 4 and 5. We then compare these methods based on actual detailed routes produced by a well known detailed router, VPR [2]. The experimental framework and the results from these comparisons are presented in Sections 6 and 7 respectively. More discussion on the inner workings and the accuracies of estimation techniques are presented along with time-quality tradeoffs in Section 8.

2. Related Work

There are many interconnect estimation techniques that are available which produce estimates ranging from congestion maps to channel by channel estimates. fGREP [6] is a new method that produces routing estimates on every channel in the FPGA. Lou's method [8], another new method, produces routing estimates on a region by region basis for ASICs. These two methods show great promise because they predict track usages on very fine level routing regions. Both these methods can be used to identify potential "hot spots" of routing congestion. These methods have signifi-

cant advantages over other methods in that they are :

Simple - The estimation methods used are very simple. Lou's method uses exact formulae and fGREP uses Breadth First Search(BFS) mechanism. Earlier methods due to El Gamal [3] and Brown et. al. [14] are stochastic based and involve complex calculations of conditional probabilities.

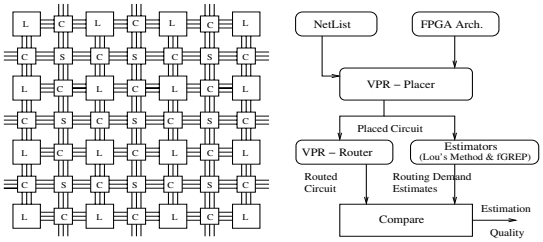
Fast - Both the methods are extremely fast. In contrast, methods like BDD based estimation [16] are very slow and sometimes exceeds incremental routing in runtimes.

Usable - These methods estimate actual track usages and thus can used to get both global and local routing usages. Target FPGA devices can be chosen based on the track usages estimated. This is unlike Rent's Rule [1] based estimation methods, which produce estimates on coarser routing regions.

Accurate - Both the methods conform to routing models used by actual routers. These methods assume that routers will try to use shortest possible paths most of the times. Many estimation methods do not have such routing information built into them.

There are variants of popular methods that are proposed recently. Van Marck et. al.'s [9] work on predicting local variations in interconnections based on Rent's rule is used by Sadowska et. al. [12] in their placement method. Wei [7] used Rent's rule and arrived on a statistical model for hierarchical FPGA routing prediction. Some other works that indirectly address the routing resource prediction or evaluation include the congestion minimization techniques due to Wang and Sarrafzadeh[15], simultaneous place and route by Nag and Rutenbar[11], and wireability analysis for gate arrays by Sastry and Parker[13].

3. Preliminaries



(a) Generic Island–Style FPGA Architecture (b) Design Flow

Figure 1. Island Style FPGA Architecture and Flow of the Experimentation

Our design flow is illustrated in Figure 1(b). We use VPR[2] for our experimental testbed. VPR's placer produces a placement solution for the given netlist and the FPGA architecture shown in Figure 1(a). The placement solution forms an input to estimation methods and VPR's router. VPR's router produces a detailed route for the most tight value of track width W_v, which is found by the binary search method[2]. The output of estimation methods is a routing demand value D_i for each channel v_i in the routing graph. The maximum value of D_i is the estimated peak routing demand or the track width. This is compared with W_v. The individual D_i values are compared with the channel occupancy values of VPR's detailed route, to bring out the local estimation accuracy of estimation methods. Since channels are the basic routing elements in both methods, we use the terms interchangeably.

The estimation methods discussed in this paper share a few concepts. First of all, both the methods are post-placement estimation methods. They take placed circuits and estimate routing demands on every region in the layout. Both these methods are based on the fact that every net exacts routing demands on all routing elements. These demands are then summed up to get final demand values. Both the methods assign zero demand to elements outside the bounding box of the net. The following two sections briefly give the salient features of both the methods. More details can be found in [6] and [8].

4. fGREP Routing Estimation Method

fGREP is an estimation method in which the estimation model is based on the concept of routing flexibility over the routing elements. Flexibility is defined as the number of alternatives available for a terminal to be reached from outside. This reachability is based on the number of elements available at the same distance from the terminal. The alternatives in reaching a terminal are restricted to be a subgraph of the routing graph(G) that contains only those elements that are in the bounding box of the net.

4.1. Routing Model

fGREP models the routing fabric as a graph $G(V, E)$, where V represents the channels in the FPGA and E represents the switchboxes in the FPGA. There exists an edge (v_i, v_j) only if there is a switchbox connecting the two corresponding channels in the FPGA. Every element in V is assigned a demand value due to every single net. These individual demands are then summed to get final demand. The routing model used in the experimentation assumes connection box flexibility $F_C = 1$ and switch box flexibility $F_S = 1$. However, the method by itself is not restrictive and can take in any routing structure.

4.2. Routing Demand - Demand due to a Terminal

fGREP takes a non-traditional approach in defining demands. A multi-terminal net n_k consists of a number of terminals and each terminal corresponds to some vertex v_i in the routing graph. Assume that a terminal v_i is in an infinite plane and different paths of infinite lengths are extended. The routing elements in these paths are assigned a demand

that is equal to the ratio of the number of paths the elements are in, to the total number of paths possible. These paths are also restricted to be non-returning. We define l_{ij} of the vertex v_j appearing in some such path to be the level of v_j in the breadth first search tree with v_i as the root. A collection of such vertices which are in the same level k from vertex v_i belong to the set L_{ik}. Formally, $L_{ik} = \{v_j \in V | l_{ij} = k\}$.

It is proven in [6] that in the infinite plane case, the ratio of number of paths that use a vertex v_j at level k is at least $\frac{1}{|L_{ik}|}$. Thus the demand due to v_i on v_j is $\frac{1}{|L_{ik}|}$. Furthermore, fGREP restricts the routing elements to those only inside the bounding box. The level sets are identified using breadth first search over the bounding box and the elements are assigned demands equal to $1/L_{ik}$.

4.3. Multiple Terminals of a Net and Multiple Nets

For a multi-terminal net, terminals that are closer to a routing element is expected to exact more demand than the terminals that are far away. Hence regions of influence are created around each terminal, and the demands assigned to all the elements in a region are due to the terminal which influences it. These regions of influence are created implicitly during the breadth first search, and demands are reassigned when closer terminals are detected. The result of this assignment is that every terminal v_j in the bounding box of net n_i gets a demand value D_j^i. All the other elements outside the bounding box get zero demand. The final demands assigned is the sum of demands due to every net and is given by $D_j = \sum_{i=1}^{\#nets} D_j^i$.

4.4. Illustration

We explain the concept of demand and the various terms involved using illustrations below. Fig 2(a) shows a hypothetical routing graph, on which v_i is the current root vertex. The vertices at the same level from v_i are all shown to be connected by dotted lines. Figures 2(b,c) show the terminals and the bounding box of the two terminal net, on the FPGA layout. The routing demands on the channels due to each of the terminals is also shown. Demands of all the channels with the same level k gets the same demand value $1/|L_k|$. Fig 2 shows the interaction of the terminals of a two terminal net. The demand entries in regular typeface are those due to terminal t_1 and those in boldface are due to terminal t_2. The entries with circles on them are equidistant from both the terminals, and are on the border of regions of influence of both the terminals. The maximum of the demands due to two terminals is assigned for them, which in this case happens to have the same value of $1/9$.

5. Lou's Routing Estimation Method

Lou et. al. [8] proposed a novel method for routing estimation for placed ASIC circuits. Their method makes use of two-terminal nets, hence every net is decomposed into two-terminal segments. Minimum Spanning Tree(MST) and Steiner Minimal Tree(SMT) are noted in [8] as two methods that can be potentially used. In this section, we describe the routing model that Lou's method is based on, how demands are calculated on that model, and how an island-style FPGA can be modeled in terms of Lou's model.

5.1. Routing Model

First of all, the layout is divided into equal rectangular grids. The model then assumes the following :
1. All nets are considered to be two terminal nets.
2. The nets are all routed with shortest length.
3. The nets are allowed to bend only once inside a grid cell.
4. The bends are assumed to be on the center of the grid cells.
5. The basic model is valid for nets occupying at least 2x2 grids. The degenerate cases are considered separately.
6. The pins are assumed to occupy left bottom and right top corners.

Assumption 1 forces every net to be decomposed using MST or SMT . Now for every two-terminal pair the demands are computed based on the theory detailed in subsection 5.2. The basic idea behind the method is that different shortest length paths bend in different grid locations. The number of paths that use a particular grid element is thus dependent on the location of the element. This results in every element having a different probabilistic usage than the others. For all elements in the grid, the paths arrive either from the left or bottom, and leave either via the right or top. The probabilistic usages are based on the number of paths that arrive at a particular grid cell (either from left or bottom) and the number of paths that leave the cell (either through right or top). Since bends are allowed, there are four possible routing structures in a grid cell. These are illustrated in Figure 3 in the grid element located at (i,j).

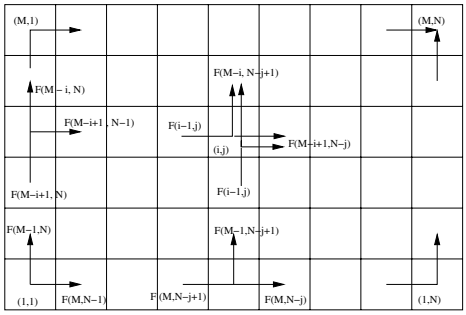
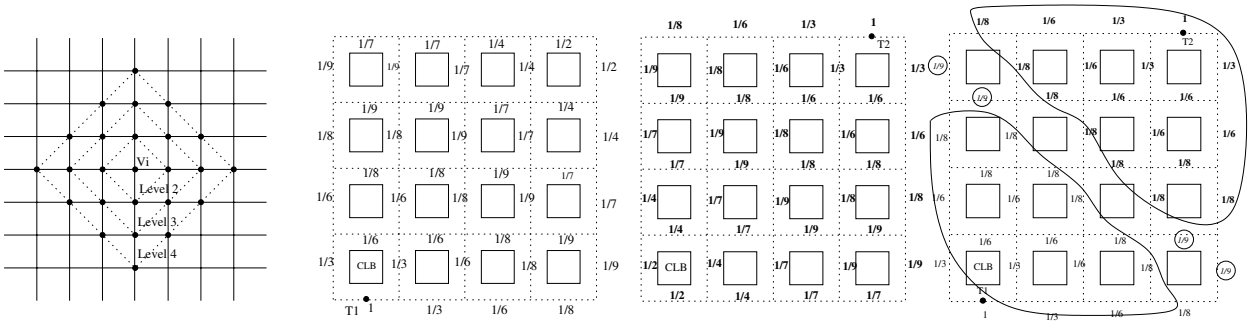


Figure 3. Enumeration of Paths at Different Grid Locations in Lou's Model

5.2. Problem Formulation

Assume that the pins are located in the left bottom and right top of a $M \times N$ mesh. The total number of ways to optimally route a two terminal pair is $F(M, N)$ and is



a) Levels In a Routing Graph b) Demands due to terminal T1 c) Demands due to terminal T2 d) Interaction of Multiple Terminals

Figure 2. fGREP - Illustration of Level and Demands

calculated as

$$F(M,N) = \begin{cases} F(M-1,N)+F(M,N-1) & M,N \geq 2 \\ 1 & M,N = 1 \end{cases}$$

This formula cannot be used directly to calculate the track usage for all the elements in the grid. This is because the horizontal and vertical demands of any grid point need not be the same. Refer to Figure 3. The grid element in the right and bottom has just one path going through the element, and that path will demand a half-horizontal track and a half-vertical track. The elements in the first column has $F(M-i+1, N-1)$ paths leaving horizontally and $F(M-i, N)$ paths vertically. The paths that enter at the bottom and leave at the top need one full vertical track and no horizontal track. On the other hand, paths entering from the bottom and leaving through the right, need half-horizontal track and half-vertical track. To account for such changes in demand based on how the paths turn, a probability matrix that captures usage is defined. It is computed as

$$P(M,N) = \begin{bmatrix} (P_x(M,1), P_y(M,1)) & \cdots & (P_x(M,N), P_y(M,N)) \\ \vdots & \cdots & \vdots \\ (P_x(1,1), P_y(1,1)) & \cdots & (P_x(1,N), P_y(1,N)) \end{bmatrix}$$

The matrix has the property that $P_{x|y}(i,j) = P_{x|y}(M-i+1, N-j+1)$. Because of this symmetric nature, the values can be computed for lower triangular matrix and then copied to the upper triangular matrix. The entries in the lower triangle are computed as follows:

$$P_x(i,j) = \frac{1}{F(M,N)} \times \begin{cases} F(M,N-1) & \text{case a: i=1,j=1} \\ 1 & \text{case b: i=1,j=N} \\ F(M-i+1,N-1) & \text{case c: 1<i<M,j=1} \\ \frac{F(M,N-j+1)+F(M,N-j)}{2} & \text{case d: i=1,1<j<N} \\ \frac{F(i,j)F(M-i+1,N-j)+F(i,j-1)F(M-i+1,N-j+1)}{2} & \end{cases}$$

$$P_y(i,j) = \frac{1}{F(M,N)} \times \begin{cases} F(M-1,N) & \text{case a: i=1,j=1} \\ 1 & \text{case b: i=1,j=N} \\ \frac{F(M-i+1,N)+F(M-i,N)}{2} & \text{case c: 1<i<M,j=1} \\ F(M+1,N-j+1) & \text{case d: i=1,1<j<N} \\ \frac{F(i,j)F(M-i,N-j+1)+F(i-1,j)F(M-i+1,N-j+1)}{2} & \end{cases}$$

These probabilities are calculated based on the position of the grid element (i,j). Case(a) is for grid position $(1,1)$, case(b) for $(1,N)$, case(c) for rest of the elements in 1^{st} row, case(d) for rest of the elements in 1^{st} column, and case(e) for rest of the elements in the lower triangular matrix. P_x values stand for probabilities of horizontal usage, and P_y for vertical. Thus a value of $(0.5, 0.8)$ means that a single horizontal track in the region will be used 1 in 2 times and a single vertical track will be used 4 in 5 times if the two-terminal pair is routed optimally in all possible ways.

5.3. FPGA Model

Lou's model captures the routing model for ASIC circuits directly because routing is done on free areas in

ASICs. Since FPGA's routing structure is not free area based, we built a model for FPGAs which essentially captures the rectangular grid structure for routing. Refer to Figure 4 in which a rectangular grid is overlaid on top of an island-style FPGA. A single grid element in our model corresponds to a switchbox and the four half channels abutting the switchbox. In [8], the authors have calculated the number of paths that use individual right, left, top and bottom segments in every grid element, but combined the results to give a horizontal-vertical pair. In our FPGA model, we use the individual values separately, and assign them to half channels. Thus every channel on the FPGA will be on two grid elements exactly. For a vertical channel, the grid on top will supply the half channel demand for the top half, and the grid in the bottom will supply the half channel demand for the bottom half. Similar case applies to horizontal channels.

We use the relative orientations of the two terminals to dictate the grid elements that are included for demand calculation. We have chosen the grids such that, the grid away from the direction of routing is excluded. The demands for the overlaid grid are now calculated using Lou's formulation and these demands are assigned to the four half-channels. This procedure is repeated for every two terminal pair, and for all the nets. The end result is the demand on all the routing channels.

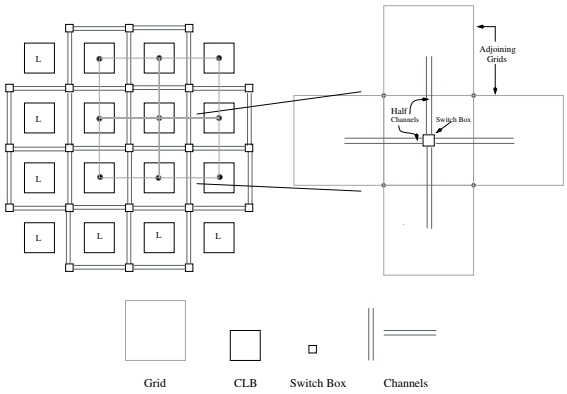


Figure 4. FPGA Model used in Lou's Estimation Method

6. Experimentation

To evaluate the estimation qualities of Lou's method and fGREP, we compare the results with actual detailed routes produced by VPR [2]. The FPGA architecture used is a generic island-style FPGA as defined in VPR, but with one I/O pad per row/column. The architecture has 4-input LUTs, with switchbox flexibility $F_s = 3$ and connection-box flexibility $F_c = 1$. Both the systems were implemented in C and executed on a Pentium 800 MHz machine running Linux. The flow of experimentation is described in Section 3. Circuits from ISCAS-89 benchmarks are used in this experimentation. First VPR is used to place the circuits. The placed results are fed to both the estimation methods. The estimation results produced are on a channel by channel basis. The estimates are individually compared with VPR's detailed router results on every channel and the mean and standard deviation of the differences are calculated to get a global picture. The peak channel widths predicted are compared against the lowest channel width that VPR used to route the circuit successfully.

Lou's method uses the FPGA model detailed in Section 5.3. The nets are decomposed into two terminal nets using Minimum Spanning Tree Method and the formulation is used on every pair. Special cases like terminals on a single row/column or in the same grid are handled as detailed in [8]. fGREP also takes the same routing architecture and placed circuits. The benchmarks that we used are the ones that are used in [6]. The benchmarks range in size from 1263 to 8384 CLBs and 1072 to 8444 nets.

7. Results

The estimation methods are compared for global and regional accuracies which are the important parameters for any estimation method. Execution times must also be considered for a more complete evaluation, since accurate estimation is not useful if it is not fast. These evaluation parameters are tabulated in Table 1. Summary of results are presented in Table 2.

Table 2. Comparison of Lou's Method and fGREP with VPR - Summary of Results

	Lou	fGREP	VPR_G	VPR_D
Total #Tracks	381.9	201.87	147	217
%Diff Tracks	76.00%	6.97%	32.26%	0
Runtime	9.32s	478s	-	2852s

The comparisons are done on a channel by channel basis with the detailed router results from VPR. Global accuracy is evaluated by comparing the peak routing demands estimated by each of these methods with those from VPR (W_v). W_{lou} and W_{fgr} are the peak demand values estimated by Lou's method and by fGREP respectively. When compared against W_v, we see that estimates produced by fGREP are closer to W_v in all the cases. The average difference is about 7%. W_{lou} exceeds W_v by 1.4 to 2.6 times, with an average inaccuracy of 76%. Thus Lou's method does not capture peak routing demand accurately. In fact, in all the cases Lou's method overestimated the peak routing demands. This factor is crucial because target FPGA devices are usually chosen based on both device size and routing availabilities. Lou's method will force designers to choose bigger FPGAs.

Local accuracies are brought out by comparing the mean and standard deviation of the differences between the estimation methods and VPR. These are tabulated as M and σ

Table 1. Comparative Results for the 20 Biggest ISCAS-89 Circuits

Ckt	*#Cells*	*#Nets*	W_{lou}	W_{fgr}	W_g	W_v	T_{lou}	T_{fgr}	T_v	M_{lou}	σ_{lou}	M_{fgr}	σ_{fgr}
alu4	1523	1536	17.724	9.974	7	11	0.203	4.956	38	2.666	1.976	1.447	1.086
apex2	1879	1916	21.136	10.539	8	12	0.208	4.751	59	2.813	2.111	1.652	1.276
apex4	1263	1271	21.162	11.956	9	13	0.143	2.560	43	2.708	2.154	1.879	1.396
bigkey	1708	1935	15.471	7.466	6	9	0.329	46.748	101	0.861	1.424	0.587	0.746
clma	8384	8444	25.760	11.059	8	13	1.411	118.280	549	2.810	2.131	1.697	1.247
des	1592	1847	16.075	8.072	6	8	0.439	26.840	76	0.760	1.151	0.593	0.652
diffeq	1498	1560	15.975	8.087	6	8	0.154	2.352	23	2.582	1.860	1.114	0.813
dsip	1371	1598	18.260	7.558	6	7	0.372	55.887	181	0.680	1.459	0.429	0.688
elliptic	3605	3734	21.632	10.461	7	11	0.602	19.430	237	2.475	1.897	1.381	1.041
ex1010	4599	4608	18.926	10.728	8	12	0.592	32.923	131	2.783	2.029	1.609	1.196
ex5p	1065	1072	19.628	12.974	10	14	0.162	1.855	35	2.667	2.061	1.853	1.461
frisc	3557	3575	20.902	11.762	9	14	0.493	15.307	150	2.723	1.996	1.662	1.269
misex3	1398	1411	18.626	10.018	6	11	0.160	3.097	45	2.719	1.987	1.508	1.143
pdc	4576	4591	24.594	16.162	11	16	0.718	35.321	460	3.365	2.464	2.132	1.523
s298	1932	1934	19.638	7.590	6	8	0.284	9.317	60	2.887	2.119	1.143	0.814
s38417	6407	6434	14.733	8.899	6	8	0.772	28.148	203	2.386	1.704	1.104	0.823
s38584.1	6448	6484	15.248	8.942	6	9	1.424	44.173	248	2.275	1.699	1.136	0.851
seq	1751	1791	19.728	10.384	8	12	0.203	4.073	53	2.766	2.032	1.680	1.249
spla	3691	3706	23.063	11.865	9	15	0.474	19.888	132	2.952	2.236	1.973	1.438
tseng	1048	1098	13.619	7.375	5	6	0.118	2.280	28	1.797	1.639	0.948	0.73

respectively. Lesser M values imply that the estimates are closer in different regions in the FPGA. σ values closer to 0 in conjunction with low mean values provides more confidence in the estimation methods. From Table 1 we see that fGREP has lesser mean values than Lou's method in all the cases. Moreover, the means are closer to 1 for majority of the cases for fGREP, which means fGREP estimates channel demands within a difference of 1 track most of the times. Lou's method estimates are within a track difference only in 3 of the 20 cases. Even in these cases the peak routing demands are off by 71-160%. Also, the deviation values represented by columns marked σ show that most of the deviation values are more than 1 in both the cases, but fGREP's deviations are all less than 1.5. fGREP has both of deviations and mean values less than Lou's method. This gives us more confidence in fGREP.

Another interesting aspect we studied in the experiment is to find out if global routers can be used as estimators. We used VPR's global router and noted the peak demands reported by it as W_g. VPR's global router uses PathFinder [10] algorithm, which is also used in the detailed router. Global router works on the channel connectivity graph, and does not take flexibilities into account. By inspecting the column headed W_g, we see that the global router consistently underestimates the peak routing demand. On an average, the estimates are away from the detailed routing results by 33%. The global router estimates are worse than fGREP's in 95% of the cases, but is better than Lou's estimates in all the cases. More discussion on the nature and accuracies of the estimates are in Section 8.

Now, a discussion on execution times is warranted. Execution times are in seconds and are marked by columns headed with T. The column headed by T_v is the execution time of VPR(detailed) when it is initiated with the maximum channel demand, W_v. W_v is actually found by a previous run which finds the lowest possible channel width using binary search. The execution times marked are thus for routing with a pre-specified required channel width. From the table we see that Lou's method is 20 to 150 times faster than fGREP, and is 55 times faster on an average. All the execution times of both the estimation methods are lesser than the runtimes of VPR. This is required of any estimation method.

In Lou et. al. [8], the authors mention that better accuracies will be obtained if the terminals are decomposed based on Steiner Minimal Trees. We do not have comparisons based on SMT based decomposition. We believe that the estimates will be closer than those made by MST based decomposition, but at the expense of large execution times. Comparisons of algorithmic complexities involving MST and SMT based calculations against those of fGREP are discussed in Section 8.

8. Discussion

The differences in accuracies of estimates and the execution times motivated us to investigate the actual reasons behind such behavior. In this section we discuss the salient features of both the methods that could explain the accuracies and the execution times.

8.1. Estimation Methodology

The concept of demand is a striking difference between the two methods. In Lou's method the demands are based on the number of alternatives available in a finite plane, namely the bounding box. In fGREP the demands are assigned as though the terminal is in an infinite plane. This lets fGREP to calculate the demands without considering the other terminals in the net. In Lou's method the demands assigned must always be in conjunction with other terminals in the net, because the formulation is based on optimal paths between a pair of terminals.

Demands placed by Lou's method and fGREP over the rectangular grid has significant differences. Figures 5(a,b) show the distribution of demands made on a grid with two terminals, one in the left bottom and the other in the right top of the grid. Darker regions have higher demands than the lighter ones. It is clear from the figure that Lou's method places high demands along the main diagonal (in the bottom left to top right direction) connecting the two terminals, and the demands are lesser as we move to the other corners. This is due to the fact that lesser number of optimal paths use the corner and edge elements, and more paths concentrate on the center. But, in fGREP, the demands are higher for the grid elements that are closer to the terminals, and lesser for those that are away from the terminals. This is clear from the main diagonal in the top left to bottom right direction having lesser demands.

The way the demands are composed together for different terminals in a net is another big difference. Lou's method decomposes terminals on a net, and terminal pairs are considered in succession. Figure 5(c) shows how the demands are composed by Lou's method for a MST based decomposition. Regions bound by every pair of two terminals are marked in grey and they get some demand values. The elements in white do not receive any demands. This is based on the assumption that routers will try to route the net closer to the MST, SMT etc. However, this does not capture the possibility that the router may not be able to route close to the tree formed. For instance, if the detailed router encounters more congestion around the MST, the "escape routes" will be through the other elements in the bounding box. These elements are shown in white in Figure 5(c). These receive zero demand, and thus the estimates will be away from detailed routes whenever escape routes are taken. fGREP, on the other hand, assigns demands to all the elements in the bounding box. It does not make any implicit assumption about the router. Since demand is based on number of ways a terminal can be reached from outside, this encompasses all possible paths that a net may be routed, irrespective of the way in which routing is done.

Another significant deviation is terminal decomposition. Decomposition of a n-terminal net results in $n-1$ regions. Each of these regions share edges with other regions. This

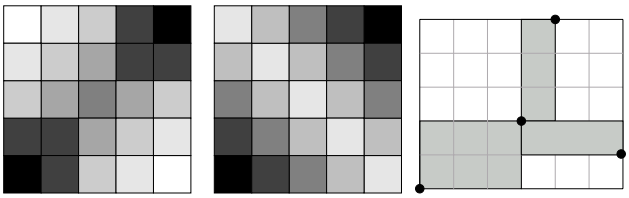


Figure 5. Illustration of demands for a two-terminal net by (a)Lou and (b)fGREP. Demand assignment for multi-terminal nets by Lou's method is shown in (c)

results in demands adding up on the edges where regions meet. No clever way of splitting terminals can circumvent this problem. This explains to some extent the inaccuracy of Lou's method. But, fGREP would create regions of influence around terminals and will not decompose nets. If there are elements that are in more than one region of influence, maximum of the demands due to the different regions will be assigned, and not the sum of the demands. Thus common elements do not get higher demands than is necessary. Again, no assumption is made on the router's inner working, namely whether it decomposes the nets or not.

fGREP has its own limitations too. It does not accurately model demands placed by terminals on the peripheries. Since the formulation is based on demands on an infinite plane, it suits terminals in the center of the bounding box well. However, terminals towards the periphery is more sensitive to the boundaries created by the bounding box. This effect is most acute for two-terminal nets. In two-terminal nets, both the terminals are on the periphery. The effect of such relative locations is that the number of alternatives available in a particular level in the infinite plane model is cut down. The demands placed on the available alternatives are thus higher than is necessary. However, Lou's model captures this correctly. Demand maps in Figures 5(a,b) make this clear. Therefore, on a design with full of two-terminal nets we expect Lou's method to give more accurate results.

8.2. Modeling Complexity

Lou's method is rigid, in the sense that a rectangular grid must be overlaid on the layout to use the formulae. It should be noted that the Lou's model was originally proposed for ASIC circuits, where rectangular grids are imposed on free areas. Applying this to any routing model is not possible. For an architecture that cannot be modeled as grids, the possible paths and ratio of paths have all to be reformulated and used. In our case, since the underlying architecture was island-style based, coming up with an accurate Lou's model for it was simple and direct.

8.3. Algorithmic Complexities and Execution Times

Lou's method involves MST calculation for all nets and calculating demands for all the two-terminal pairs. The algorithmic complexities of these methods are $O(N \cdot t_{max}^2)$ and $O(N \cdot W \cdot H)$ respectively. N is the number of nets, t_{max} is the maximum number of terminals in any net, and W and H are the dimensions of the FPGA. The overall complexity is $O(N(t_{max}^2 + W \cdot H))$. fGREP does BFS search from every terminal and thus the overall complexity is $O(t_{avg} \cdot N \cdot W \cdot H)$. For our circuits, $t_{max}^2 \approx W \cdot H$. This results in Lou's method being faster than fGREP by a factor of t_{avg}, which can be expected from the formulations. For SMT based demand calculations for Lou's methods, such speedups cannot be expected. Computing SMT of multi-terminal nets is NP-complete [4]. A well known approximation method called the Batched 1-Steiner method(B1S) is due to Griffith et. al [5] and is established to work well in practice. The time complexity of Lou's method using this SMT decomposition is $O(N(t_{max}^3 \cdot k + W \cdot H))$, where k is the number of runs of B1S method. In such a case fGREP will be faster than Lou's method by a factor of $\frac{k \cdot t_{max}}{t_{avg}}$. We do not have results using SMT based decomposition. Lou et. al. claim that such an estimation method will give better results. But as discussed earlier in this section, even SMT based decomposition will suffer from the boundary effects just like the MST based method. It remains to be seen how accurate the estimations are with such a method.

9. Conclusion

In this paper we have compared two new post-placement routing demand estimation methods, our method fGREP [6] and Lou's method [8]. Our experiments conclude that fGREP's estimation mechanism is more accurate than Lou's method, both on global and channel-by-channel basis. We find that Minimum Spanning Tree based decomposition of nets results in inaccurate estimates in Lou's method, and is 76% away from actual occupancies on an average. On the contrary, fGREP is just 7% away from the actual demands. We also observe that Lou's method is faster by more than a factor of 55 on an average. Our complexity analysis of Lou's method indicate that if Steiner Minimal Tree based decomposition is done, the execution times of Lou's estimation will be slightly slower than fGREP.

Future work may include adopting the fGREP model to ASICs. Congestion estimation for ASICs must deal with different pin locations in a grid, finding grid dimensions that will cut down execution times without appreciable loss of accuracy etc. Also, ASICs may have macro-cells and blackbox IP cores which block routing. In [8], Lou. et. al have provided different ideas to deal with these. fGREP can also handle these cases directly. It will be interesting to see how fGREP's estimation accuracies are affected for free area models. Also, since the estimation methods are very fast, they can be used inside placement for congestion minimization.

References

[1] H. B. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI*. Addison Wesley, Reading, MA, 1990.

[2] V. Betz and J. Rose. VPR: A New Packing, Placement and Routing Tool for FPGA research. In *Field-Programmable Logic and Applications*, pages 213–222. Springer-Verlag, Berlin, Sep 1997.

[3] A. A. E. Gamal. Two-Dimensional Stochastic Model for Interconnections in Master Slice Integrated Circuits. *IEEE Trans. CAS.*, Feb 1981.

[4] M. Garey and D. Johnson. The Rectilinear Steiner Problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32:826–834, 1977.

[5] J. Griffith, G. Robins, J. Salowe, and T. Zhang. Closing the Gap: Near Optimal Steiner Trees in Polynomial Time. *IEEE Transactions on Computer-Aided Design*, 13(11):1351–1365, November 1994.

[6] P. Kannan, S. Balachandran, and D. Bhatia. fGREP - Fast Generic Routing Demand Estimation for Placed FPGA Circuits. In *Lecture Notes in Computer Science*, Belfast, Northern Ireland, August 2001. 11th International Workshop on Field-Programmable Logic and Applications, FPL, Springer.

[7] W. Li. Routability Prediction for Hierarchical FPGAs. In *Proc. Great Lakes Symposium on VLSI*, 1999.

[8] J. Lou, S. Krishnamoorthy, and H. Sheng. Estimating Routing Congestion using Probabilistic Analysis. In *Proceedings of the 1999 International Symposium on Physical Design (ISPD)*, 2001.

[9] H. V. Marck, D. Stroobandt, and J. V. Campenhout. Toward an Extension of Rent's Rule for Describing Local Variations in Interconnection Complexity. In *Proceedings of the 4th International Conference for Young Computer Scientists*, pages 136–141, 1995.

[10] L. McMurchie and C. Ebeling. PathFinder: A Negotiation-Based Performance-Driven Router for FPGAs. In *ACM Symp. on FPGAs, FPGA95*, pages 111–117. ACM, 1995.

[11] S. K. Nag and R. Rutenbar. Performance-driven Simultaneous Placement and Routing for FPGAs. *IEEE Transactions on CAD*, June 1998.

[12] G. Parthasarathy, M. Marek-Sadaowska, and A. Mukherjee. Interconnect Complexity-aware FPGA Placement Using Rent's Rule. In *To appear in, Proc. Intl. Workshop on System Level Interconnect Prediction (SLIP)*, April 2001.

[13] S. Sastry and A. Parker. Stochastic Models for Wireability Analysis of Gate Arrays. *IEEE Trans. on CAD*, Jan 1986.

[14] S.Brown, J.Rose, and Z.G.Vranesic. A Stochastic Model to Predict the Routability of Field Programmable Gate Arrays. *IEEE Transactions on CAD*, pages 1827–1838, Dec 1993.

[15] M. Wang and M. Sarrafzadeh. Congestion Minimization During Placement. In *Proceedings of the 1999 International Symposium on Physical Design (ISPD)*, 1999.

[16] R. Wood and R. Rutenbar. FPGA Routing and Routability Estimation via Boolean Satisfiability. In *ACM International Symposium on FPGAs FPGA98*. ACM, June 1998.