On Metrics for Comparing Routability Estimation Methods for FPGAs

Parivallal Kannan, Shankar Balachandran, 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
{parik, shankars, dinesh}@utdallas.edu

ABSTRACT

Interconnect management is a critical design issue for large FPGA based designs. One of the most important issues for planning interconnection is the ability to accurately and efficiently predict the routability of a given design on a given FPGA architecture. The recently proposed routability estimation procedure, fGREP [6], produced estimates within 3 to 4% of an actual detailed router. Other known routability estimation methods include RISA [5], Lou's [7] method and Rent's rule based methods [1] [11] [9]. Comparing these methods has been difficult because of the different reporting methods used by the authors. We propose a uniform reporting metric based on comparing the estimates produced with the results of an actual detailed router on both local and global levels. We compare all the above methods using our reporting metric on a large number of benchmark circuits and show that the enhanced fGREP method produces tight estimates that outperform most other techniques.

1. INTRODUCTION

Interconnect prediction is the process of estimating the routing resource requirement and/or utilization, before actually performing the routing process. For the FPGA design flows, the term routability estimation is more appropriate, as the routing resources are fixed for a given device. Given a particular FPGA device and a design to be mapped onto the device, routability estimation is the process of identifying the number of routing elements needed to perform a complete routing. If the device has enough routing elements to satisfy the requirement, then the design is routable. The estimation process attempts to produce routing demand values on every programmable routing element on the device. Useful parameters derived from the estimation process are the *peak routing demand* and the *routing demand distribution*. For a successful routing, the device must satisfy both the peak routing demand and the routing demand distribution requirements. The problem of interconnect prediction has been studied extensively for the ASIC design flows. Most of the recent work in interconnect prediction and routability estimation is based on the Rent's rule [1]. Recently, some efforts have been made to address the problem for FPGAs [9].

DAC 2002, June 10-14, 2002, New Orleans, Louisiana, USA.

2. ESTIMATION METHODS

In general, a good routability estimation method should be :
Fast - Routability estimation is typically used inside other physical design tools and hence speed is critical
Generic - The method should be independent of the FPGA device architecture
Usable - The estimation should produce usable results for a wide variety of applications. Typically we are interested in peak routing demand, channel utilization and routing demand distribution
Accurate - Undoubtedly, the estimation should be reflective of actual numbers obtained from a detailed router. Either the numbers should directly correlate with the detailed routing results or at least the ratios should match over a large set of benchmarks.

Some of the routability estimation methods that satisfy these requirements and currently under wide usage are fGREP [6], RISA [5], Lou's method [7] and Rent's rule derivatives [11] [9]. In this paper, we analyze and compare all the above methods. fGREP is a theoretical method that uses the concept of routing flexibility to model routability. RISA [5] is an empirical method based on the wiring distribution map, derived from a large number of randomly generated optimal Steiner trees. Lou's method [7] is based on the ratio of the number of paths that use a specific routing region to the total number of paths possible. Yang et. al. [11] proposed the use of Rent's rule for congestion estimation.

All the methods except fGREP were originally proposed for ASIC design flows. However, they are generic enough in their formulation that a direct translation to an FPGA design flow is trivially possible. We have implemented all these methods for a generic island style FPGA architecture as described in [3]. To compare the quality of the estimates, we use the VPR [3] FPGA physical design suite's detailed router. In the following sections, we explain these methods in detail.

3. FGREP

fGREP is a theoretical routability estimation method in which the estimation model is based on the concept of routing flexibility over the routing elements. fGREP models the routing fabric of the FPGA as a graph $G(V, E)$, where V represents the channels in the FPGA and E represents the switchboxes. For each $v_i, v_j \in V$, there exists an edge $< v_i, v_j > \in E$ *iff* channels v_i and v_j share a switch box between them. Since it operates on the routing graph

and makes no assumptions of the routing architecture, fGREP can be applied to any FPGA architecture without any modifications.

A net $n_k \in N$, where N is the netlist, is made up of a set of terminals $T_k \in V$. Every terminal $t_k^i \in T_k$, exacts a certain routing demand called the *terminal-demand*, on all the routing elements inside the net bounding box. According to fGREP, this terminal-demand on a routing element at a distance of $l = q$ from the terminal is proportional to the total number of elements at the same distance l from the terminal. The distance is measured on a breadth-first search tree on the routing graph, with the terminal as the root. The set of equidistant ($l = q$) routing elements from a terminal t_k^i, is defined as the level set,

$$LS_k^{iq} = \{v_j \in V | l_{ij} = q\} \quad (1)$$

The terminal-demand on the routing element v_j is then,

$$TD_k^{ij} = \frac{1}{|LS_k^{iq}|}; where, l_{ij} = q \quad (2)$$

fGREP then derives a quantity called the *net-demand* for all routing elements inside the net bounding box. The net-demand of v_j is defined as the terminal-demand due to the terminal with the lowest value for the distance metric l_{ij}.

$$ND_k^j = TD_k^{ij} | l_{ij} = min(\forall_i l_{ij}) \quad (3)$$

The final *routing-demand* on the routing element v_j due to all the nets in the netlist is then,

$$D^j = \sum_{k=1}^{\#nets} ND_k^j \quad (4)$$

The terminal-demands due to all the terminals of all the nets are calculated by performing a breadth first traversal of the routing graph with the terminal v_i as the root vertex. At each step of the traversal, the level set is enumerated and the demands are assigned. The net-demand is then calculated from the terminal demands as per Equation 3. The total routing element demand due to all the nets is then calculated as per Equation 4.

4. ENHANCEMENTS TO FGREP

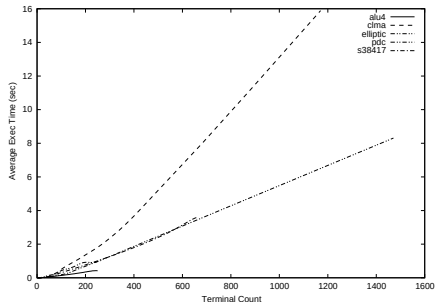


Figure 1: Terminal Count vs Avg. Execution Time

The runtimes of fGREP are high for large circuits with many high-fanout nets. If E_k is the set of routing elements in the bounding box of a net n_k and T_k the set of terminals, then the runtime is proportional to $|E_k| \times |T_k|$. Typically high-fanout nets span the entire device and hence add a severe penalty to the fGREP runtimes. This effect is clearly illustrated in Figure 1, which plots the average fGREP execution time per net against the number of terminals in the net, for a few standard benchmark circuits.

4.1 Zone Limited Search

Equation 2 produces the routing demands due to one net on all the routing elements inside the net bounding box. It can be observed that the operation produces *zones of influence* around each terminal. All the routing elements inside a zone have their net-demands produced by a single common terminal. Conversely, for all the routing elements outside a terminal's zone of influence, the net-demands are produced by other terminals. Hence, theoretically, the terminal-demands due to a terminal v_i need be assigned for only those elements in its zone of influence. The runtime complexity of the operation in Equation 3, is then only of the order of $|E_k|$, instead of $|E_k| \times |T_k|$. We propose to use this technique of limiting the terminal-demand assignments to the zone elements only to reduce the runtimes of fGREP.

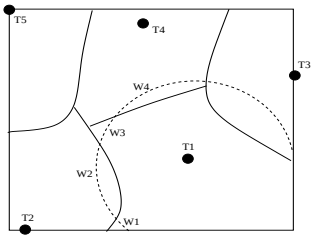


Figure 2: Zones, Wavefront Clipping and Fragmentation

The zone limited search technique cannot be directly applied to the fGREP method. fGREP relies on the enumeration of the level set as per Equation 1. The level sets are identified by starting a breadth-first search from the terminal. At each step of the search, the current elements form a wavefront, and expands outwards. By limiting the search to within the terminal's zone alone, fragmenting and clipping of the wavefront can occur. This has the effect of reducing the level set size and produce artificially high terminal-demand values. Fig 2 illustrates the clipping and fragmenting of the search wavefront. The wavefront for the terminal T_1 is shown by the dotted line. The zones are marked by solid lines. The wavefront W of T_1 is divided into 4 arcs, two of which (W_1 and W_3) are inside T_1's zone while the other two (W_2 and W_4) are outside the zone. According to fGREP, the cardinality of the level set at this distance will be $|W_1| + |W_3|$, which is far less than that of the complete wavefront. This will produce very high terminal-demands for the routing elements on this wavefront, which is clearly erroneous.

4.2 Zone limited Parallel Search

To overcome the wavefront clipping and fragmenting effects described above, the complete wavefront has to be maintained at all stages. But to gain speedup, the search has to be limited to the terminal's zone of influence alone. A simple yet effective solution to the problem is to maintain the complete wavefront for the terminal as long as at least one routing element on the wavefront is still contained in its zone. We propagate breadth first traversals from all terminals of a net simultaneously, every time maintaining the complete wavefront. If at any point, all the elements on the wavefront of a terminal are outside its zone (that is no element on the wavefront got its terminal-demand from this terminal) then we have completely discovered its zone and stop the search for that terminal alone. This process terminates when all the terminals' zones are discovered.

We observed that this enhancement to fGREP is very effective and resulted in up to 30X speedup in execution times while producing the same estimates. The estimation results and the run-times of fGREPcan be obtained from [6].In further discussion we will re-

fer to the enhanced fGREP method only, which we shall refer to as fGREP2.

5. RISA

RISA [5] is a very fast empirical estimation technique. It was originally proposed for ASIC design flows, but can be readily adapted to FPGA design flows. The method involves multiplying the net bounding box of a net by a pin-count dependent net-weight q, to produce the routing demand due to the net. The net-weights for various pin-counts are generated from a one time operation involving the production of a *wiring distribution map, (WDM)*. A WDM for a given pin-count M is produced by adding up and normalizing demands from optimal steiner trees for K sets of M random points (K is very large, $\approx$ 10000). The mean value of the WDM thus obtained is called the net-weight for a pin-count of M. Chang [5] provides experimental values for the net-weights for $1 \leq M \leq 50$ and net-weights for higher values of M, can be obtained by a linear regression process.

Having obtained the net-weights, the original expressions for estimating the routing demand on a routing region R_i due to a net n_k is,

$$D_k^{i,horizontal} = q \times \frac{w \times l}{Y \times L}; D_k^{i,vertical} = q \times \frac{w \times l}{X \times W} \quad (5)$$

where, (W, L) is the dimension of the routing region and (X, Y) is the dimension of the net bounding box of the net n_k and (w, l) is the overlap between the routing region and the net bounding box. For FPGAs, the routing region is a channel and hence $W = L = w = l = 1$. So, Equation 5 becomes,

$$D_k^{i,horizontal} = q \times \frac{1}{Y}; D_k^{i,vertical} = q \times \frac{1}{X} \quad (6)$$

The total routing demand on a routing element on the FPGA, as per RISA is then the sum of the demands due to all the nets. It can be observed that the routing demand due to a net is the same over all the routing elements in its net bounding box and is calculated once for every net.

6. LOU'S METHOD

Lou et. al.'s estimation method [7] is a recent routing estimation method developed for placed circuits in ASIC design flow. The chip area is divided into rectangular grids called *regions*. This method produces routing demands on different regions of the chip. Any given net in a circuit poses routing demands only on the regions that are in the bounding box of the net. For a two-terminal net, the demand assigned to a routing region is the number of tracks that the net needs in that particular region. This is dependent on the ratio of the number of shortest paths that use that particular region to the total number of shortest paths possible in routing the net.

Assume a net n_k with two terminals, one placed in the lower left corner and the other in the top right corner of a $M \times N$ grid in the chip. Let the total number of shortest paths possible to route this net be $F(M, N)$. This value can be calculated as

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

The demand on a particular region (i, j) is clearly dependent on the number of paths $F(i, j)$ that pass through the region.

In addition to the number of paths possible through a region, the usage is also dependent on whether the path bends in a particular location or not. If a particular path does not bend in (i, j), it places a demand of one full track in the direction of path and zero demand in the other direction for that location. However, if a path bends in (i, j), that path will place a demand of only half track in each of the directions. Considering both the number of paths through a particular region and the nature of the bends that these paths make, the number of tracks needed by the net n_k in region (i, j) is calculated as $P_x^k(i, j)$ and $P_y^k(i, j)$, the horizontal and vertical usages for the net n_k respectively. The usage matrix for the bounding box gives the *probability usage matrix* $P^k(M, N)$ where every matrix entry is a pair $(P_x^k(i, j)), (P_y^k(i, j))$. The regions outside the bounding box are assigned zero demands. For the formulae that are used to calculate the exact usages, refer to [7] .

In [7], Lou. et. al. suggest that the multi-terminal nets be decomposed as a Minimum Spanning Tree (MST) or a Rectilinear Steiner Tree(RST) and then use the two-terminal model for each such pair. The usage matrices of all these pairs are calculated and summed up to get demand due to the net. The final demand due to all the nets on a region is the sum of the individual demands due to each net similar to Equation 4.

In [2], we adapted the Lou's model to FPGA design flows. A rectangular grid was overlaid on an island-style FPGA, where a single grid element corresponds to a switchbox and its four incident *half-channels*. Every channel in the FPGA will be on two grid elements in this overlaid grid. Lou's method is used on this grid for estimation. The horizontal and vertical usages estimated for a region are then assigned to the half-channels. Thus demands are obtained on a channel by channel basis. Experiments on 20 large FPGA benchmarks are reported in [2].

7. ENHANCEMENTS TO LOU'S MODEL

Lou et. al. [7] note that MST decomposition may result in inaccuracies because of over-counting on the overlapping net segments. They do not provide any solution to alleviate the problem. In [2], we established that the problem with segment overlaps applies to any two-terminal decomposition, including the RST based decomposition. Further, we claimed that even if a clever net decomposition may remove large net overlaps, the bounding boxes of the pairs may still overlap thus resulting in inaccuracies. Here we propose a simple yet effective enhancement to the Lou's model that addresses the overlapping-segment problem in this section.

We decompose the two-terminal nets using MST. For all the two-terminal pairs of a single net, we calculate the probability usages for the two-terminal bounding region. For the regions that are on overlapping bounding boxes, we assign the *maximum* of the demands due to different pairs, similar to fGREP's scheme in Equation 3. We call this scheme Lou(Max). This is in contrast to [7], where the authors add up the demands on the overlapping segments. The enhancement is based on the observation that the regions in the overlapping segments actually belong to the bounding box of one single net and adding up the demands will just result in over-estimation on those regions. By finding the maximum of the demands, we do not over-estimate on these segments. Further, we also ensure that the demand assigned is at least as much as the demand posed by any of the two-terminal pairs. This simple enhancement improves the quality of the estimation by as much as 103% compared to the FPGA model proposed in [2]. This is clearly illustrated in the Table 1, where the peak demands estimated by both the methods are compared against the actual peak demands obtained by performing detailed routing using VPR.

8. RENT'S RULE BASED METHOD

Rent's rule is an empirical observation that states the relationship between the number of blocks B in a circuit and the number of

Table 1: Comparison of Lou's Method with Lou(Max)

Circuit	W_{lou}	$W_{lou(max)}$	W_{vpr}	$\%better$
alu4	17.724	13.866	11	35.07
clma	25.760	18.849	13	53.16
dsip	18.260	11.032	7	103.26
s298	19.638	12.423	8	90.18
spla	23.063	19.197	15	25.77

external connections P of the circuit. Specifically,

$$P = T_b B^r \tag{7}$$

where T_b is the average number of interconnections per block and r is the Rent exponent. Yang et. al. [11] list reports of Rent's rule being successfully used to estimate wirelength, and state that since wirelength is a measure of routing demand, Rent's rule can be used for estimating congestion. They also propose methods to estimate the peak routing demand and regional routing demand using Rent's rule. The methods are proposed for ASIC design flows and only the peak routing demand formulation can be directly ported to FPGAs. The regional routing demand method uses expressions for total interconnect lengths that may not be valid for FPGAs. There is no work that uses Rent's rule to estimate routing demand on local regions for FPGAs. For our comparison purpose, we use the peak routing demand estimation method alone from [11]. Yang et. al. [11] perform a rough min-cut based placement to estimate the Rent's parameter. We perform the same operations on a placed circuit, but do area partitioning instead.

The basic idea behind the method is to recursively partition a self-similar circuit and place it hierarchically on a layout divided into equal rectangular tiles. The number of edges crossing these tile boundaries become the routing demand of that region. The cut-set sizes $C_{i,j}$ that result from such partitioning at every level i follow the relation,

$$C_{i,1} = C_{i,2} = \cdots = C_i,\ i = 1, 2, \cdots, 2H \tag{8}$$

where H is the number of hierarchical levels. Also, the ratio between the cut sizes of two successive levels i and $i+1$ is given by $\alpha = \frac{C_{i+1}}{C_i} = 2^{-r}$, where r is the Rent's exponent. Using this expression the upper bound of the maximum routing demand of a tile boundary, C_{max}, is derived. By extending this method for a uniform distribution of cut nets over the partitioning area, the authors have arrived at a modified upper bound of the number of crossings to be

$$C_{max} < \frac{C_1}{\sqrt{N_c}}(\frac{1}{2} + 2\alpha)\frac{\sqrt{N_c}\,\alpha^{2H} - 1}{2\alpha^2 - 1} \tag{9}$$

where N_c is the total number of gates in the circuit. For any design, r and C_1 can be found by recursive bipartition. The authors also note C_1 corresponds to the Region II of Rent's curve where there is usually a small dip in the number of wires that come out of blocks. A correction is applied to Eqn. 9 where C_h/α^{h-1} is used instead of C_1. In our experiments we have used $h = H/2$ which corresponds to the middle hierarchies. In our experimentation we take a placed FPGA circuit and identify the cuts across different partitioning levels by dividing the areas recursively. Rent's exponent is calculated as the slope of the log-log plot of the number of cells and the number of nets plotted over different partitioning levels. We use the FPGA dimension instead of $\sqrt{N_c}$ in Equation 9. We consider the resulting C_{max} for the circuit as the peak routing demand estimated by this method.

9. OTHER METHODS

Brown et. al. [10] developed a stochastic model for routability based on the probabilities of making individual C_T two-point connections. The probability that a connection C_i can be routed, $P(R_{C_i})$, is modeled as a sequence of conditional events on the switches in the path. Routability of the circuit is then given as Routability $= \frac{1}{C_T}\sum_{i=1}^{C_T} P(R_{C_i})$. Routability is the percentage of nets that can be successfully routed in the FPGA's routing architecture. Predictions are done for island-style FPGAs by varying the switch box and connection-box flexibilities (F_S and F_C respectively). For every circuit a single routability number is produced. The results are validated with the number of unrouted nets produced by an actual router. The method does not produce global or local routing demands.

In [9], Parthasarathy et. al. perform placement by modifying the cost function in VPR to account for interconnection. Rent's exponent for the routing architecture(p_a) and the design(p_d) is calculated and the minimum device size is scaled by a factor of $C = N_{gates}^{\frac{p_d}{p_a}-1}$ to accommodate the design's complexity. For every change in the placement, the changes in local Rent exponent of the design p_d^l is calculated. The modified cost function is $\sum_{n=1}^{N_{nets}}(1 + |p_d^n - p_a|)(wire\_length_n)$. The authors compare the track usage of VPR's placement and the modified placement. However, the scaling of device(C) that was performed is not mentioned. VPR is known to produce very tight placements and hence the device scaling factor is of critical importance.

Chan et. al. [4] predict the routability of unplaced circuits using Rent's exponent based wirelength calculations. They derive expressions for the routing requirements inside and outside all routing regions and then assign confidence of routability on a scale of 3 - Unroutable, Easily routable and Marginal.

10. ESTIMATION QUALITY METRIC

In this section we propose a uniform reporting metric for routability estimation methods. The driving need for such a metric is the requirement to be able to quantitatively compare the various estimation methods for specific applications. Most reports on estimation methods do not try to compare the estimation quality with known results, which can be independently asserted. fGREP is a notable exception, in that it compares its results with a well known detailed router. RISA incorporates its estimation technique inside a placement loop and compares the placement quality with that of the solution produced by the same placement loop without routability estimation. The two placement solutions are routed with a global router and two indices *overCon* and *dsCost* are used for comparison. The first index represents the number of global grids with demand higher than supply and the latter represents the linear summation of the demand overshoot. RISA does not report which global router was used. In [7], there is no quantitative comparison with detailed or global router results. Only two congestion maps, one estimated and another produced by a commercial router are reported. As in RISA, the details about the router are not reported. In [11], the peak routing demand estimates are compared with a *L-Shaped* global router. The global router approximates all routes as *L* shapes along the net perimeter, and is a crude approximation to regular global and detailed routers. From the above discussion, it is evident that the estimation community needs a common metric by which existing and new estimation methods can be compared and independently verified.

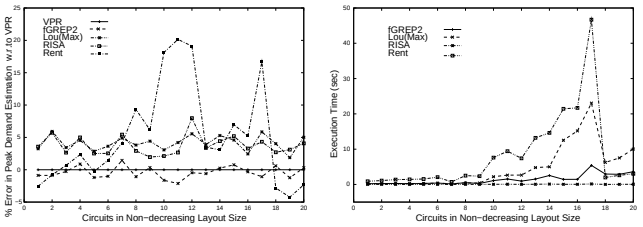
The metric should be based on well known, real world routers and should capture both regional and global estimation quality. For FPGA research, a very commonly used place and route tool suite

is the VPR [3]. VPR is available as a free download in source form and can be freely modified for research purposes. VPR's detailed router is based on the PathFinder [8] negotiated congestion algorithm and is generally considered to produce very tight results among the FPGA research community. VPR and other PathFinder based routers derived from it currently hold the FPGA place and route challenge [3]. So, using the VPR's detailed router as the common denominator for comparison among various estimation methods is a valid choice.

We propose four parameters as adequate and required quality metrics. They are the *Peak Demand error*(W), *Mean of regional errors*(μ), the *standard deviation of the errors*(σ) and the *runtimes*(t). Global estimation quality can be stated by comparing the peak routing demands with the maximum channel width needed by VPR to route a placed design. For regional estimation quality, a more detailed metric has to be designed. We are interested in the correlation between the estimates produced on local routing regions and the actual channel utilization reported by VPR. A good measure of the correlation which is very easy to calculate is the mean of the errors between the estimated and routed channel utilization. The standard deviation of the errors adequately represents the distribution of the errors.

11. EXPERIMENTATION AND RESULTS

We have implemented fGREP2, enhanced Lou's method, Rent's rule based estimation method as per [11] and RISA for a generic island style FPGA architecture. We compare all these methods using the four parameters that we derived in the previous section. We used the largest benchmarks available from the ISCAS-89 benchmark set. All experiments were conducted on a standard 800MHz Pentium PC running Linux.



(a) Comparison of Errors in Estimating Peak Demands (b) Comaprison of Execution Times

Figure 3: Comparison of Peak Routing Demands W and Execution Times

The results are tabulated in Tables 2 and 3. The Table 2 tabulates the peak routing demands and execution times for VPR and all the estimation methods. The first 5 columns in the Table 2 show the peak routing demands as per VPR, fGREP2, RISA, Lou's method and the Rent's rule based method respectively. The column labelled T_{VPR} is the execution time for the VPR's detailed router run with a fixed channel width of $W = W_{VPR}$. The last four columns in the Table 2 show the execution times for the estimation methods. All times are shown in seconds. The row labelled *Totals* in the Table 2 gives the sum of the tracks and runtimes over all the benchmark circuits. The row labelled $\frac{W_{error}}{W_{vpr}}$ gives the percentage error of the peak routing demand estimates, with respect to VPR. The row labelled $\frac{T_{vpr}}{T_{est}}$ gives the ratio of the execution times of the estimation methods with respect to VPR's detailed router. Table 3 tabulates the mean and standard deviation of the errors between the estimation methods and VPR's detailed router.

Fig 3(a) plots the peak routing demands as estimated by all the methods with respect to VPR. It can be seen that fGREP2 is the closest to VPR. The Rent's rule based method is very much away from VPR. The estimates are also randomly distributed and do not show any distinct trend. Similar results have been reported by the authors themselves in [11]. RISA and Lou's method are better but they too lack distinct trends and have random swings of about 3 to 5 tracks. To gain sufficient confidence levels, an estimation method should either closely match real world results or at least show comparable estimates for different nets. Random swings indicate that there are a number of circuits for which the estimation method is completely wrong, while there are other circuits for which the estimates are better.

Fig 3(b) plots the execution times for all the estimation methods. It can be seen that RISA has the lowest execution times, closely followed by fGREP2.

The accuracy of the regional estimation is best captured by the mean and standard deviation as explained before. To better illustrate the effect, we plot the row and column averages of the local estimation errors. Fig 4 shows the row and column averages of the estimation errors for the benchmark circuit *diffeq* and *s38417*. In these plots, the smaller the numbers, the better is the estimate. It can be seen that fGREP2 produces the best regional estimates followed by RISA and Lou's method.

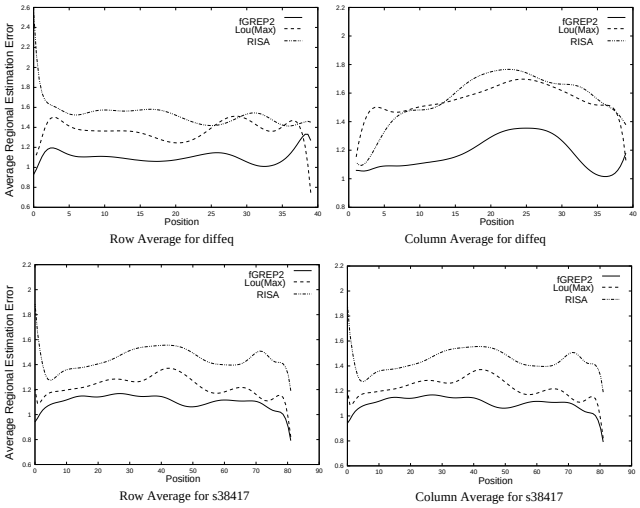


Figure 4: Average of the regional estimation errors

12. CONCLUSION

We analyzed all the routability estimation methods available till date and proposed enhancements to two of the methods fGREP and Lou's method. We discussed the need for consistent reporting of routability estimation results and proposed a minimum set of comparison metrics. We implemented four routability estimation methods for FPGAs and compared them using our new metric. We conclude that the method fGREP2 is the most accurate while RISA is the fastest.

13. REFERENCES

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

[2] S. Balachandran, P. Kannan, and D. Bhatia. On Routing Demand and Congestion Estimation for FPGAs. In *Proc. 15th Intl. Conf. on VLSI Design and 7th ASP-DAC*, January 2002.

Table 2: *P*eak Routing Demands and Execution Times (seconds)

Circuit	VPR	fGREP2	RISA	Lou(Max)	Rent	T_{VPR}	T_{fGREP2}	T_{RISA}	$T_{Lou(Max)}$	T_{Rent}
alu4	11	9.802	13.506	13.866	10.717	38	0.2654	0.0125	0.2927	1.541
apex2	12	10.92	14.911	15.815	21.322	59	0.4884	0.0225	0.4148	2.494
apex4	13	12.12	18.716	18.931	12.195	43	0.2533	0.0132	0.2230	1.091
bigkey	9	7.760	12.105	10.866	4.7610	101	3.0146	0.0269	7.5189	2.602
clma	13	11.92	17.327	18.848	29.713	549	4.8111	0.1359	23.047	46.68
des	8	8.308	12.075	13.021	5.6691	76	3.6362	0.0631	10.030	3.015
diffeq	8	8.881	12.995	12.535	10.324	23	0.2035	0.0078	0.2679	1.433
dsip	7	7.606	9.699	11.032	4.1760	181	2.8295	0.0196	6.2362	1.977
elliptic	11	10.53	18.979	16.557	30.111	237	0.9493	0.0317	2.6474	7.408
ex1010	12	11.39	15.376	15.894	15.499	131	1.4727	0.0479	4.7849	13.20
ex5p	14	13.12	17.571	17.231	11.461	35	0.2031	0.0115	0.1862	0.904
frisc	14	12.37	16.117	17.063	32.125	150	1.0845	0.0463	2.1932	7.600
misex3	11	10.72	13.649	14.430	11.682	45	0.2824	0.0145	0.2600	1.377
pdc	16	16.24	20.418	21.287	19.067	460	2.4975	0.1031	4.9885	14.61
s298	8	8.340	9.963	12.423	14.150	60	0.2792	0.0104	0.4135	2.332
s38417	8	8.764	13.192	12.597	14.963	203	1.3543	0.0352	12.530	21.43
s38584.1	9	8.676	12.283	11.437	14.238	248	1.6993	0.0346	15.158	21.69
seq	12	10.99	14.530	15.629	13.468	53	0.3905	0.0194	0.3782	2.072
spla	15	12.87	17.663	19.197	35.149	132	1.4808	0.0597	2.6312	9.422
tseng	6	7.430	11.431	10.890	10.118	28	0.1486	0.0058	0.2133	0.717
Totals	217	208.785	292.446	299.549	320.908	2852	27.344	0.7216	94.416	163.595
$\frac{W_{error}}{W_{vpr}}\%$	-	3.79	34	38	47.9	-	-	-	-	-
$\frac{T_{vpr}}{T_{est}}$	-	-	-	-	-	-	104X	3952X	30X	17.5X

[3] 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.

[4] P. K. Chan, M. D. Schlag, and J. Y. Zien. On Routability Prediction for Field Programmable Gate Arrays. In *Proc. 30th DAC*, 1993June 1993.

[5] C.-L. E. Chang. RISA: Accurate and efficient Placement Routability Modeling. In *Proc. of ICCAD*, 1994.

[6] P. Kannan, S. Balachandran, and D. Bhatia. fGREP - Fast Generic Routing Demand Estimation for Placed FPGA Circuits. In *11th International Workshop on Field-Programmable Logic and Applications, FPL*. Springer-Verlag, Berlin, August 2001.

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

[8] 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.

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

[10] 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.

[11] X. Yang, R. Kastner, and M. Sarrafzadeh. Congestion Estimation During Top-down Placement. *IEEE Transactions on CAD*, 21, Jan 2002.

Table 3: μ and σ w.r.to VPR's Detailed Router

Circuit	fGREP2		RISA		Lou(Max)	
	μ	σ	μ	σ	μ	σ
alu4	1.471	1.107	1.726	1.307	1.601	1.257
apex2	1.690	1.301	1.960	1.584	1.773	1.416
apex4	1.927	1.425	2.246	1.776	1.773	1.454
bigkey	0.577	0.733	0.881	0.794	0.623	0.879
clma	1.735	1.266	1.932	1.495	1.791	1.405
des	0.589	0.645	0.809	0.777	0.629	0.811
diffeq	1.125	0.821	1.527	1.174	1.439	1.116
dsip	0.425	0.681	0.901	0.670	0.459	0.838
elliptic	1.415	1.076	1.621	1.285	1.482	1.186
ex1010	1.654	1.223	1.908	1.436	1.763	1.346
ex5p	1.868	1.470	2.272	1.832	1.799	1.424
frisc	1.720	1.313	1.784	1.387	1.782	1.358
misex3	1.555	1.180	1.849	1.462	1.712	1.299
pdc	2.163	1.553	2.287	1.767	2.251	1.752
s298	1.180	0.850	1.317	0.992	1.593	1.234
s38417	1.104	0.823	1.628	1.246	1.402	1.095
s38584.1	1.120	0.854	1.544	1.172	1.316	1.026
seq	1.720	1.286	1.970	1.547	1.790	1.414
spla	2.021	1.476	2.126	1.667	1.897	1.488
tseng	0.937	0.731	1.173	0.938	1.139	0.931