ABSTRACT
Congestion is a fundamental problem in VLSI design flows. Typically, it is handled by feeding back density information to the placers and routers. Fast and accurate congestion estimation is key in order to obtain a design flow with less iterations and higher predictability.

Fast congestion prediction is based on an accurate approximation of the actual routing engine. In this paper we show experimentally that the number of two-pin nets with more than two bends in the actual router is negligible. It is also established that the ratio between the number of L-shapes and Z-shapes is more or less a constant.

A fast and accurate algorithm for congestion prediction is developed. The above observations are translated into probabilities, that are used to “smear” out a net over its possible realizations. Extensive experimental evidence is provided using industrial designs.

Categories and Subject Descriptors
B.7.2 [Hardware, Integrated Circuits, Design Aids]: Layout, Placement and Routing

General Terms
Algorithms, Design, Experimentation

Keywords
Congestion, Congestion Prediction, Routing

1. INTRODUCTION AND MOTIVATION
Wire congestion can be regarded as an over-using of the finite routing resources on the chip. In an automated IC implementation flow, congestion problems are typically dealt with by feeding back wire density information into the routers and placers. This flow is not fully automated, unfortunately. When automatic detouring fails, the designer must adjust his/her floorplan. The flow problem is that congestion information only becomes available after routing, which is one of the last and slowest design steps. As a result, when design completion seems near, due to congestion a large part of the (physical) design trajectory may have to be re-performed.

If quick and accurate congestion prediction is available early in the design cycle, it can be used “under the hood” in routers and placers. In this way congestion problems can be dealt with before they actually occur. It also enables floorplan designers to evaluate the alternatives with congestion in mind. A flow with less iterations, but higher predictability, and higher designer productivity is possible.

In this paper, we present a probabilistic method to estimate congestion, based on placement information, before global routing. In our approach, a number of likely paths is considered. Probabilistic usages are assigned to the regions that these paths go through. Conceptually, the net is “spread” over its possible realizations. The idea of using such a statistical approach is not new. Our contribution is based on the practical observation that after routing, the bulk of the nets will be implemented according to a limited set of patterns (called L-shaped or Z-shaped).

In previous works, congestion prediction was based on Rent’s rule [9], wiring distribution models [8], or probabilistic models [4, 1, 5]. Congestion may also be calculated during global routing, and used for ripup-and-reroute schemes [2]. Among the works that use congestion information to improve placement or routing are [7, 8], and [3, 2], respectively. In [1, 5], floorplanning with congestion minimization is performed.

Our approach extends that of [4]. In that paper, the analysis is based on two-pin nets (multi-pin nets are broken up). All detour-free paths between two pins are considered, and given an equal probability of being selected by the global router. Each net contributes the probability of occupying a track in the regions it may be routed through to the probabilistic usage of these regions. This approach yields a high probabilistic usage at the center of the box spanned by a net (see Fig. 1-a). We will provide experimental proof that in practice, most nets are routed with one or two bends. This suggest a high probabilistic usage at the borders of the bounding box of the net. In our approach, we take this into account (Fig. 1-b).

We will start the remainder of this paper with some definitions (Sec. 2), followed by the models we use for congestion (Sec. 3). In Sec. 4 we will present our experimental results, and in Sec. 5 we will draw conclusions.
This may be a fraction, when for instance a usage of ordinates (horizontal and vertical) is considered. The horizontal (vertical) usage of a bucket is the number of occupied horizontal (vertical) tracks. This may be a fraction, when for instance a connection ends at a pin in the bucket. \( H_{\text{bucket}} \) and \( W_{\text{bucket}} \) are the height and width of a bucket. The distance between the left, right, bottom and top border of a bucket and a pin \( p \) in that bucket is denoted by \( l_p, r_p, b_p, \) and \( t_p \) respectively. The pin’s coordinates are referred to as \( x_p \) and \( y_p \). A net \( n \) may span a number of buckets. The width and height of this netbox are \( w_n \) and \( h_n \), and are expressed in number of buckets.

Our approach is based on the analysis of two-pin nets. Since we try to mimic the behavior of the global router, we treat multi-pin nets the same: they are broken up into two-pin nets using a Minimum Spanning Tree algorithm. Our experimental results confirm that this simplification does not significantly impair accuracy.

In Fig. 2, different kinds of nets are shown. As in [4], flat nets and short nets are special cases: they are supposed to be routed without bends. They reside fully in a single row/column or in a single bucket, respectively.

Figure 2: The different kinds of nets.

\( U_{\text{hor}}(i,j) \) and \( U_{\text{ver}}(i,j) \) are the horizontal and vertical probabilistic usages due to a net \( n \) in the bucket with coordinates \((i, j)\), with \( t \) the coordinate in vertical direction. We use the term congestion or usage to denote the maximum of horizontal and vertical congestion or usage. The horizontal and vertical capacities of a bucket are defined as \( C_{\text{hor}}(i,j) = H_{\text{bucket}} \sum l_i / p_{\text{hor}} \) and \( C_{\text{ver}}(i,j) = W_{\text{bucket}} \sum l_i / p_{\text{ver}} \), where \( p_i \) denotes the minimum pitch at layer \( i \).

2. DEFINITIONS

The following definitions are illustrated by Figs. 3-5. Our placement area is divided in buckets\(^3\), the same way the global router does. The size of a bucket is typically 10 × 10 wire tracks. This bucket size is the result of the trade-off between runtime for global routing (bigger buckets is faster), and accuracy of congestion analysis (smaller is better). The horizontal (vertical) usage of the bucket it is in is the horizontal (vertical) distance between its pins and \( |x_a - x_b| \) and \( |y_a - y_b| \).

3. CONGESTION MODEL

The base algorithms that are used in global routing minimize wire length. Experimental evidence shows that they are effective in this (on average only 1.3% of the two-pin nets in our test suite are detoured), and we may thus assume detour-free routing. Global routing algorithms are also tuned towards reducing the number of vias. In our test suite, only 1.2% of the two-pin nets had more than two bends. Therefore, in our model we consider detour-free routing with at most two bends (Z-shape). In [1, 5], similar models are used.

The pin positions and the possible shapes for the net implementation will affect the overall the routing resources. We will first discuss the usages of short nets, flat nets, and \( L \) and \( Z \) shapes, and then how they are combined, based on probabilities that are extracted from real-life designs.

In the analysis of two-pin nets, we will assume that the two pins are located in the lower-left corner, and upper-right corner of their net box. Results for other configurations can be obtained similarly.

3.1 Usage of short nets

Under the assumption of detour-free routing, the length of a two-pin short net \( n \) (Fig. 3) is equal to the Manhattan distance between its pins \( a \) and \( b \). The horizontal (vertical) usage of the bucket it is in is the horizontal (vertical) distance normalized for bucket-size:

\[
U_{\text{hor}}(i,j) = \frac{|x_a - x_b|}{W_{\text{bucket}}} = \frac{H_{\text{hor}}(i,j)}{205} \quad \text{and} \quad U_{\text{ver}}(i,j) = \frac{|y_a - y_b|}{H_{\text{bucket}}} = \frac{H_{\text{ver}}(i,j)}{205}.
\]

\( l_i \) and \( p_{\text{hor}} \) and \( p_{\text{ver}} \) are the sets of layers with preferred horizontal and vertical directions, respectively. Blockages are taken into account by subtracting the number of blocked tracks from the capacity of a bucket. The (probabilistic) horizontal and vertical congestion of a bucket is defined to be the ratio between (probabilistic) usage and capacity.

\( l_i \) and \( p_{\text{hor}} \) and \( p_{\text{ver}} \) are the sets of layers with preferred horizontal and vertical directions, respectively. Blockages are taken into account by subtracting the number of blocked tracks from the capacity of a bucket. The (probabilistic) horizontal and vertical congestion of a bucket is defined to be the ratio between (probabilistic) usage and capacity.

\( l_i \) and \( p_{\text{hor}} \) and \( p_{\text{ver}} \) are the sets of layers with preferred horizontal and vertical directions, respectively. Blockages are taken into account by subtracting the number of blocked tracks from the capacity of a bucket. The (probabilistic) horizontal and vertical congestion of a bucket is defined to be the ratio between (probabilistic) usage and capacity.
3.2 Usage of flat nets

For a vertical flat net (see Fig. 3), we find for the vertical usage of the bottom bucket: \( U_{ver} = \frac{t_n}{h_{bucket}} \), which accounts for the distance to the border of the bucket. Similarly, for the top bucket, we find \( U_{ver} = \frac{t_n}{h_{bucket}} \). For the buckets in between, the vertical usage is 1, since the wire traveling through the bucket will occupy a full vertical track. The total horizontal distance that has to be traveled is distributed equally among the involved buckets: each gets a horizontal usage of \( U_{hor} = \frac{\left| x_n - x_b \right|}{h_n w_{bucket}} \).

3.3 L-shaped nets

Nets with \( w_n > 1 \) and \( h_n > 1 \) need to have at least one bend. An L-shaped route (see Fig. 4) can be thought of as a horizontal flat net plus a vertical flat net. The usage due to an L-shape is the sum of the usages of the two virtual flat nets. Two L-shapes are possible. Therefore, the usage due to each L-shape is multiplied by a factor 1/2 to find the total L-usage. L-shapes with \( w_n = h_n = 2 \) are referred to as small L-shapes. They differ from other L-shapes in that no Z-shape is possible.

\[
U_{hor} = \frac{r_n}{W_{bucket}} \quad U_{ver} = \frac{l_n}{H_{bucket}} \quad U_{ver} = \frac{1}{2} \frac{h_n}{H_{bucket}} \quad U_{hor} = \frac{1}{2} \frac{w_n}{W_{bucket}}
\]

Figure 4: Usages due to an L-shape.

3.4 Z-shaped nets

Z-shapes are the most complicated case. If both \( w_n \) and \( h_n \) are larger than 2, we can have two orientations: horizontal and vertical, named after the orientation of the center piece of the Z-shape.

Now let’s consider the usage of vertical Z-shapes (Fig. 5). For the leftmost bucket in the bottom row, the horizontal usage is \( \frac{r_n}{W_{bucket}} \). For the other buckets in the bottom row, the horizontal usage consists of two terms: the first for the case the vertical segment will start in the bucket. The chance of this happening is \( \frac{1}{w_n - 2} \), since there are \( w_n - 2 \) candidate buckets for the bend. The horizontal usage in that case would be \( \frac{1}{2} \) since the bend would on average be in the middle of the bucket. The other term is for the case the bend occurs to the right of the bucket. The probability for this is \( \frac{w_n - x - 2}{w_n - 2} \), where \( x \) is the horizontal position of the bucket, relative to the leftmost buckets, and the usage would be 1. Only if the bend occurs in a bucket, it gets vertical usage. The total vertical usage of \( \frac{t_n}{H_{tile}} \) is therefore evenly spread over the candidate buckets.

For the top row, the reasoning is similar. Horizontal bucket usage consists of two terms. One for the case that the bend occurs in that bucket, and one for the case the bend occurs to its left. Vertical usage is evenly spread over the buckets. Buckets in the center of the netbox have a chance of \( \frac{1}{w_n - 2} \) that a vertical track is occupied by a vertical Z-shape, resulting in a vertical usage of \( \frac{1}{w_n - 2} \).

The number of horizontally and vertically oriented Z-shapes is \( h_n - 2 \) and \( w_n - 2 \). In order to find the total Z-usage, we therefore scale the horizontal and vertical Z-usages with \( \frac{h_n - 2}{h_n} \) and \( \frac{w_n - 2}{w_n} \), respectively, as we did with a factor \( \frac{1}{2} \) for the L-shapes, and sum them all up.

\[
U_{hor} = \frac{x_n - 1}{w_n - 2} + \frac{1}{w_n - 2} \cdot \frac{1}{2} U_{ver} = \frac{1}{w_n - 2} \quad U_{hor} = \frac{1}{2} \frac{h_n}{H_{tile}} \quad U_{ver} = \frac{1}{2} \frac{w_n}{W_{tile}}
\]

Figure 5: Usages due to vertically oriented Z-shapes.

3.5 Combination of usages

Of all nets that could become an L- or Z-shape, a large fraction will become an L-shape, a smaller fraction will become a Z-shape, and a small fraction will be detoured, or have more than two bends. We define

\[
\alpha = \frac{\#nets_L}{\#nets_L + \#nets_Z}.
\]

Over a number of designs, \( \alpha \) has an average value which we will use to combine the probabilistic usages:

\[
U_{LZ} = \alpha_{avg} U_L + (1 - \alpha_{avg}) U_Z.
\]

The basic procedure is shown in Fig. 6.

Essentially, \( \alpha \) is the part of the two-pin nets that is optimally routed: with only one bend. If this is not possible, the router will use two bends, and so on. A better router will be able to route more nets optimally, resulting in a larger value of \( \alpha \). If a design is more congested, the router will have a harder job, resulting in a lower \( \alpha \). This observation is supported by our experimental evidence. Nonetheless, the results are such that an average can be used.

3.6 Properties of usages

The idea behind probabilistic congestion prediction is to “spread” a net over its possible realizations. Each horizontal track is spread over a column, and a vertical track is

\[
\frac{\#nets_L - 2}{\#nets_L + \#nets_Z}.
\]
spread over a row. With $xl_n$ and $yb_n$ the left and bottom bucket coordinate of the netbox of a net $n$, the following must therefore be true.

$$\sum_{i=yb_n}^{yb_n+h} U_{k, n}^i \left( i, j \right) = 1 \quad \forall \quad xl_n + 1 \leq j \leq xl_n + w - 2 \quad (3)$$

$$\sum_{j=xl_n}^{xl_n+w_n} U_{v, n}^j \left( i, j \right) = 1 \quad \forall \quad yb_n + 1 \leq i \leq yb_n + h - 2 \quad (4)$$

It can be shown that our manipulations such as e.g. the combination of L-usage and Z-usage preserve this property. This is the case, because the method is essentially based on probabilities, that always sum up to 1.

### 3.7 Blockages

The parts of a layer that may not be used for routing and placement, are called blockages. They are subtracted from the capacity of a given bucket, and affect the congestion thus. **Full blockages** block all layers in a (set of) bucket(s). They are dealt with similarly to simple blockages in [4]: each bucket with a distance to the blockage lower than $D$ gets a weight associated with it: $w = 2^{-d} \cdot n$, where $d$ is the Manhattan distance to the blockage, and $n$ the number of unblocked neighboring buckets. After running the probabilistic usage estimation, the total usage of the blockage is divided over the neighboring buckets, proportional to their weights. The difference with [4] is that we can also handle blockages larger than a single bucket. **Line blockages** block an entire row or column of the bounding box of a net. It is solved by introducing two virtual pins at the periphery of the blockage (Fig. 7). Note that in the presence of too many or too complicated blockages, a probabilistic method is not the way to go.

### 4. EXPERIMENTAL RESULTS

In our experiments, we used the Blast Chip 4.0 physical synthesis software by Magma Design Automation[6]. Programming was performed primarily in Magma-TCL. We maintained the usage per bucket. On a net-by-net basis, probabilistic usage is added to the buckets in the pattern. This yields a complexity of $O(\#\text{nets} \cdot \#\text{buckets})$. Because most nets are short in practice, the actual run time will be closer to $O(\#\text{nets})$.

![Figure 6: Basic prediction algorithm.](image)

Table 1 shows the main statistical data of the industrial designs we used for our experiments. These were blocks or entire chips with various applications. The numbers correspond to the state of the design after global routing. For the two-pin nets, $\alpha$ (the ratio between the number of L-shapes and number of L- plus Z-shapes), the percentage of the nets that has a detour, and the percentage of the non-detoured nets that have more than two bends was extracted. These numbers are shown in Table 2. This table clearly shows that detoured nets and nets with more than 2 bends can be ignored. It is also evident that the harder a design is (higher utilization), the more nets are detoured. As a result, slightly more nets in congested designs have more than two bends.

![Figure 7: To avoid a line blockage, two virtual pins and thereby three new 2-pin nets and netboxes are introduced.](image)

Table 2: Results for the benchmarks

<table>
<thead>
<tr>
<th>chip</th>
<th>#cells</th>
<th>#nets</th>
<th>#buckets</th>
<th>est. wire-len. [\text{m}]</th>
<th>utiliz. 2pin</th>
<th>#bends&gt;2 [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>191427</td>
<td>242251</td>
<td>62556</td>
<td>23.156</td>
<td>57.4</td>
<td>69</td>
</tr>
<tr>
<td>b</td>
<td>10180</td>
<td>10862</td>
<td>22650</td>
<td>7.072</td>
<td>71.3</td>
<td>55</td>
</tr>
<tr>
<td>c</td>
<td>49709</td>
<td>45315</td>
<td>327750</td>
<td>2.394</td>
<td>99.1</td>
<td>64</td>
</tr>
<tr>
<td>d</td>
<td>7737</td>
<td>7830</td>
<td>408321</td>
<td>1.140</td>
<td>85.7</td>
<td>80</td>
</tr>
<tr>
<td>e</td>
<td>17102</td>
<td>20028</td>
<td>33920</td>
<td>0.69</td>
<td>84.8</td>
<td>75</td>
</tr>
<tr>
<td>f</td>
<td>46871</td>
<td>40801</td>
<td>55328</td>
<td>1.40</td>
<td>1.2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chip</th>
<th>$\alpha$</th>
<th>detoured [%]</th>
<th>#bends&gt;2 [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.60</td>
<td>0.60</td>
<td>0.55</td>
</tr>
<tr>
<td>b</td>
<td>0.67</td>
<td>0.15</td>
<td>0.042</td>
</tr>
<tr>
<td>c</td>
<td>0.56</td>
<td>3.73</td>
<td>2.0</td>
</tr>
<tr>
<td>d</td>
<td>0.50</td>
<td>2.68</td>
<td>3.7</td>
</tr>
<tr>
<td>e</td>
<td>0.61</td>
<td>0.83</td>
<td>0.19</td>
</tr>
<tr>
<td>f</td>
<td>0.65</td>
<td>0.39</td>
<td>0.69</td>
</tr>
<tr>
<td>avg</td>
<td>0.60</td>
<td>1.40</td>
<td>1.2</td>
</tr>
</tbody>
</table>
Figure 8: Segment distributions for six industrial designs ((a)-(f)), and the average (g).

Figure 9: Segment distribution for a 5x5 grid, according to Lou.

Fig. 10 shows how the difference between predicted and actual usages is distributed over the map. In Fig. 11, we see the distribution of the errors. The error is expressed as a percentage of the highest usage in the design. The errors are randomly distributed over the map, and more than 65% of the buckets is within 5% error. Only a few buckets (<1%) have more than 20% error.

5. CONCLUSION

In this paper, we presented a fast technique to predict congestion, suitable for incorporation in routers or placement engines. The technique is based on experimental evidence that shows that only a negligible number of nets will have detours, and that the number of nets with many bends can also be ignored. All possible realizations of a net under these assumptions are considered, and the net is effectively “smeared out” over these realizations. The way this “smearing” occurs is based again on experimental evidence. It is shown that the resulting congestion prediction matches actual congestion well.

6. REFERENCES

Figure 12: Results for the first chip: predicted usage (a), usage according to global routing (b), and final usage (c).

Figure 13: Results for the second chip: predicted usage (a), usage according to global routing (b), and final usage (c).