A Predictive Distributed Congestion Metric and its Application to Technology Mapping*

Rupesh S. Shelar
Department of ECE
University of Minnesota
Minneapolis, MN.
rupesh@ece.umn.edu

Sachin S. Sapatnekar
Department of ECE
University of Minnesota
Minneapolis, MN.
sachin@ece.umn.edu

Prashant Saxena
Intel Labs. (CAD Research)
Intel Corporation
Hillsboro, OR.
prashant.saxena@intel.com

Xinning Wang
Intel Labs. (CAD Research)
Intel Corporation
Hillsboro, OR.
xinning.wang@intel.com

ABSTRACT

Due to increasing design complexity, routing congestion has become a critical problem in VLSI designs. This paper introduces a distributed metric to predict routing congestion for a premapped netlist and applies it to technology mapping that targets area optimization. Our technology mapping algorithm is guided by a probabilistic congestion map for the subject graph to identify the congested regions. Experimental results on the benchmark circuits in a 90nm technology show that congestion-aware mapping results in a reduction of 37%, on an average, in track overflows as compared to conventional technology mapping.

Categories and Subject Descriptors

B.6.3 [**Design Aids**]: Automatic syntheis; Optimization

General Terms

Algorithms, Design

Keywords

Congestion prediction, technology mapping

1. INTRODUCTION

With increasing design complexity, designs are increasingly becoming wire-limited [1], thus aggravating the problem of routing congestion. Although exact routing congestion information is known only after global routing, failure to address congestion prior to this point implies that the designer is left with few degrees of freedom. Moving one step back, to placement, provides greater flexibilities, but is still not enough and it is known that this can still lead to significant design iterations.

It is imperative to address congestion issues early in the design process to allow for more freedom to reduce congestion, and our work addresses this issue during the synthesis step. Technology mapping, which nowadays is interleaved with physical design for better delay estimation, provides powerful capabilities for absorbing long interconnect wires into internal connections within complex gates, or for splitting complex gates into simpler gates, thus helping to alter the overall distribution of wires in the layout. During this step, the routing congestion problem may be attacked with relatively more freedom (albeit relatively less information) than during placement and routing.

While congestion is an important consideration for technology mapping, the overriding objectives continue to be metrics such as area or delay or power. Therefore, it is more appropriate to use congestion as a constraint rather than as an objective. While optimizing for area and delays, it is desirable to ensure that the final netlist does not have congested spots, so that long detours are avoided and the netlist remains routable. Typically, very few places in the circuit (ideally, zero) should have congestion values that are greater than some threshold, and the final netlist should be well optimized from the area and/or delay perspectives. Some related work in the past is summarized as follows. Stok *et al.* proposed a clustering of closely placed cells during technology mapping so that the matching choices covering distantly placed cells in the subject graph are ruled out [2]. This approach may result in long wires in the final netlist, and more importantly, may be so limiting as to leave a significant portion of the design space unexplored. Pandini *et al.* proposed wirelength as a metric to be minimized during technology mapping in order to reduce the congestion [3]. Although large wirelength may be correlated with high congestion, the correlation is rather poor, and therefore, this may not result in an effective optimization. This observation has been borne out by recent work by the same authors [4], who state that such a metric, when considered during technology mapping employing a traditional cost function ($K_1 \times$Area$+K_2 \times$Delay$+K_3 \times$Wirelength, where K_1, K_2, and K_3 are constants), may not result in decreased congestion. As pointed out by them, congestion is a local property

*This work was supported in part by SRC under contract 2002-TJ-1092 and those under award NSF CCR-0098117.

ISPD'04, April 18–21, 2004, Phoenix, Arizona, USA.

that varies from bin to bin, and it is difficult to capture its effects using a global metric like wirelength. This observation led them to the conclusion that congestion can only be targeted using iterative placement and technology mapping. However, such a conclusion is valid only when the congestion optimization is performed using an indirect global metric in a traditional fashion. Instead of trying to absorb the congestion information into a single metric, we work with information about the distribution of congestion over the entire layout. The contributions of our work can be summarized as follows.

- Using empirical data obtained from several benchmarks, using different scripts, placement algorithms, and libraries, we show the fidelity between the congestion maps for the subject graph and the mapped netlists, and then exploit this fidelity during technology mapping.

- Instead of using an indirect metric such as wirelength, we use probabilistic congestion estimates to guide our technology mapping; these were shown in [5] to have good fidelity with post-routing congestion.

- The congestion cost function is defined such that the mapper chooses area-optimal matches when the corresponding wires are likely to pass through a sparsely congested region, while congestion-optimal matches are chosen when the corresponding wires are likely to pass through a densely congested region. Thus, different optimization modes are applied at different places in the circuit depending on the context. To the best of our knowledge, such selective optimization has not been applied during technology mapping in the published literature.

2. PROBLEM DEFINITION

Routing congestion depends on the following factors: the connectivity of the network, its placement, and the routing solution. Since there is relatively less freedom to attack routing congestion during the placement and routing stages, we concentrate on the first factor in this paper. The technology mapping step makes crucial decisions regarding the connectivity of the network, since the mapping of primitive gates to the library cells determines the set of wires that will be present in the circuit netlist. Traditionally, this has been carried out without any placement information. Although this has changed in recent physical synthesis vendor offerings, most approaches focus on the prediction of wirelength based on bounding box estimates that ignore congestion. The estimation of routing congestion without a placement for a network is, if not impossible, liable to be highly inaccurate, and one may have to rely on high level metrics such as adhesion [6]. However, this is a very new metric and several open questions about it remain unanswered: for example, whether it can be measured in a computationally efficient manner, and whether its fidelity is valid for mapped netlists. On the other hand, probabilistic congestion estimation [5] used after the placement of a mapped network has been demonstrated to correlate well with the congestion map generated after the routing, on both academic and industrial benchmark circuits. The estimation method divides the layout into bins and computes the congestion for a given bin under all possible routes for a given net. The congestion is defined as follows.

DEFINITION 2.1. *The congestion for a given bin is the ratio of number of tracks required to route the nets through the bin to the number of tracks available.*

We use the probabilistic method of [5] to guide our technology mapping algorithm. However, even such a method is difficult to adapt, since only the premapped netlist is available prior to technology mapping, and the level of correlation between the probabilistic congestion maps of the premapped netlist and the mapped netlist has not been studied in the past. One contribution of this work is to perform such a study. From empirical evidence obtained using different logic synthesis scripts and placement algorithms on a variety of benchmarks, we show a good congestion correlation between premapped and mapped netlists. Once we establish the congestion correlation between the premapped and mapped netlist, the problem of congestion-aware technology mapping can be defined as follows.

PROBLEM DEFINITION 1. *Given a subject graph of a network and a library of gates, synthesize an area-optimized network such that the maximum (horizontal/vertical) congestion over all of the bins is less than the given threshold.*

3. CONGESTION FIDELITY

This section explores the level of fidelity between the congestion estimates before and after technology mapping for any given circuit. For a given circuit, a netlist before technology mapping contains primitive gates such as 2-input NANDs, and technology mapping creates a netlist consisting of a set of gates from the given library. We refer to a netlist before technology mapping as a *subject graph* or a *premapped netlist* and that after technology mapping as a *mapped netlist*. Intuitively, the premapped and mapped netlist for a given circuit share the same global connectivity since the mapper absorbs some wires of the subject graph into the internal nodes of library cells, leaving other wires untouched. This points towards the possibility of good fidelity between congestion maps for premapped and mapped netlists. However, congestion also depends on the placement of elements (primitive gates or gates in the library) in the netlist. Placement algorithms used by commercial tools and in academia are typically based either on recursive multi-level bisectioning or force-directed quadratic programming. It would be useful to understand, even empirically, whether these placement algorithms react to the same global connectivity and block area constraints in a similar way. If so, there may be a good congestion correlation between premapped and mapped netlist. We explore this issue by performing a set of experiments using different placers, different logic synthesis scripts, different libraries, and different benchmarks.

3.1 Experimental setup

To verify the fidelity between congestion estimates before and after technology mapping, we placed several premapped netlists, and the corresponding mapped netlists using the same block area and the same placement of input/output terminals. Two different placement algorithms were used – a recursive bisectioning based algorithm in a publicly available tool, Capo [7], and a force-directed quadratic algorithm, Kraftwerk [8], implemented in a proprietary industrial placer. Different scripts, such as *rugged, boolean, algebraic, espresso,* and *speedup* in SIS [9] were used for preprocessing the netlists before technology mapping employing different libraries in SIS as well as an industrial library used for high performance microprocessor designs. Mapping was performed in SIS using the *map -s -n 0 -AFG -p* command that performs area and fanout optimization. No layout information was used to guide this technology mapping. Placement using Capo [7] was performed with default options to minimize the total wirelength based on half perimeter bounding box estimates, while placement using Kraftwerk [8]

was performed to minimize total wirelength as well as congestion.

The premapped netlist is an abstract Boolean network. Since the number of nodes in this netlist is large, the area of primitive gates must be scaled by a certain factor to present the same white space constraints for the placement as the mapped netlist. This factor is computed *a priori* as a ratio of the targeted gate area to the area of premapped network. Note that this factor is readily available given the block area, the percentage area utilization, the premapped netlist, and the cell library, and does not require any testcase-specific tuning.

3.2 Experimental results

We show results for a few representative benchmarks: C432, C6288, C7552, and an industrial circuit containing an instruction decoder (IDC) in a high-performance microprocessor. Similar results are observed on other circuits, but are not shown here due to space limitations. Apart from the vastly different functionalities, the sizes of these benchmarks also vary from a few hundred cells to a few thousand cells. Figures 1 (a) and (b) show congestion maps for the benchmark IDC for the mapped and premapped netlists, respectively. The placement of both the networks is performed using Kraftwerk. In these plots, the XY plane shows the two dimensions of the layout area, while the Z-axis depicts the congestion. Visually, one can conclude that the distribution shown in Figure 1(b) is similar in nature to the congestion map shown in Figure 1(a).

Representative results for some ISCAS'85 benchmarks and the IDC circuit using different scripts, libraries, and placers are shown in Table 1. Columns 2, 3, and 4 show the scripts used, the number of cells in the mapped netlists, and placement tools used, respectively. Technology mapping in SIS [9] is performed using the area and fanout optimization option, employing the lib2.genlib library in SIS and an industrial library. It is worth noting that the mapped netlist is fanout-optimized, which possibly restructures the network after the mapping and may affect the global connectivity adversely. Columns 5 (6) and 7 (8) in the table show the average and maximum horizontal (vertical) congestion, respectively, while columns 9 and 10 show the statistical correlation between the congestion in premapped and mapped netlist. The correlation is defined as $\frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$, where $E[]$ is the expectation, μ is the mean, σ is the standard deviation; in our case, X and Y correspond to the congestion in the premapped and mapped netlists, respectively. A correlation value closer to 1 (-1) means that two random variables are strongly positively (negatively) correlated, while a value close to 0 means that variables are weakly correlated [10].

3.3 Justification based on experimental results

In spite of fanout optimization that may affect the global connectivity and hence congestion fidelity, the congestion correlation between subject graph and mapped netlist is always greater than 0.6, and is often quite close to 1, for all the netlists. One may deduce the following based on these experimental results.

- Across different libraries, scripts, benchmarks, fanout optimization, and placement algorithms, a good correlation exists between the congestion map for the subject graph and congestion map for a mapped netlist.

- The reasons for the congestion correlation are likely to be the similarities in the global connectivity in the subject graph and the mapped netlist, the similar block area and I/O terminal constraints, and the way any reasonable placement algorithms react to such resemblances in global connectivity and the block area constraints.

4. CONGESTION-AWARE MAPPING

For the purposes of congestion-aware technology mapping, the sparsely congested and densely congested regions must be identified. From the above experiments that demonstrate the congestion correlation between a subject graph and its mapped netlist, we can conclude that the former netlist is accurate enough for this purpose. The primary objective of our congestion-aware technology mapper is area minimization, and we employ a variation of a dynamic programming-based technology mapping algorithm [11]. The technology mapping procedure involves the matching and covering phases: the former comprises storing the set of optimal matches at each node, while the latter involves constructing the network by selecting from the matches stored during the matching.

4.1 Example

A pure area minimization objective during technology mapping can result in poor congestion, and Figure 2 illustrates a case where suboptimal area matches may reduce congestion. Assume that all of the bins, shown as dashed squares in the figure, are congested and a match for the AOI33 function is considered. The inputs to the match enter through top and bottom bins on the left, while the output leaves from the middle bin on the right. Figure 2(a) shows one possible match containing two three-input NANDs, a two-input NAND, and an inverter, while Figure 2(b) and Figure 2(c) show an alternative match, an AOI33, under two different placements. To simplify the computations, if we use the number of bin-boundary crossings as the congestion metric, instead of the probabilistic congestion metric, then the cost for the match in Figure 2(a) is 12, while that for the AOI33 matches in Figures 2(b) and (c) are 20 and 15, respectively. The latter also happens to be the minimum over all placements for the area-optimal AOI33 match. It is clear that the match in Figure 2(a) distributes the logic and therefore, creates lower congestion. This example also highlights limitations of the placement in alleviating congestion, when area-optimal matches are chosen.

The cost of wires depends on the context: wires are inexpensive in sparsely congested regions, but are expensive in densely congested regions due to possible detours and hampered routability. One way to reduce this cost in densely congested zones without penalizing the design excessively is to account for their congestion contributions only in those zones. Our congestion-aware mapping heuristic serves this purpose well: in densely congested spots, it considers probabilistic routes based on the center-of-gravity locations for all possible matches and chooses the match that minimizes the congestion, while in sparsely congested spots, it chooses area-optimal matches.

4.2 Congestion cost computation

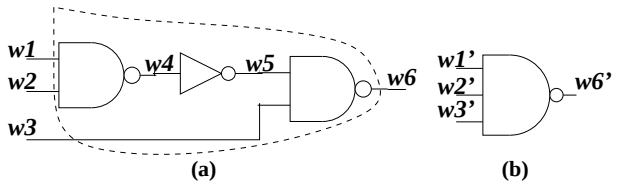

Figure 3: Computing the congestion cost of a match: (a) An example subject graph, (b) One possible match.

The congestion-aware mapping heuristic requires the assignment of a congestion cost, along with an area cost, to each match. The congestion cost depends on the total congestion caused due to the

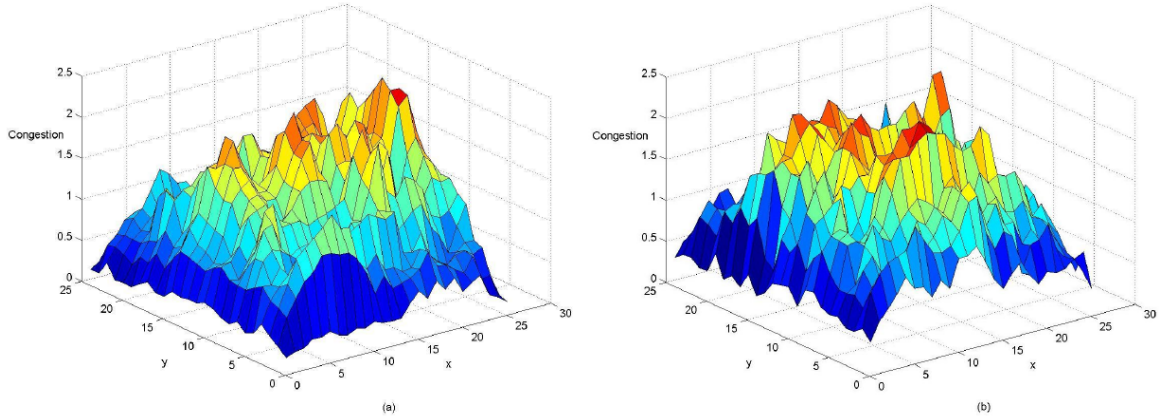

Figure 1: Vertical congestion for IDC for (a) the mapped netlist and (b) the premapped netlist. *script.boolean* is used for preprocessing the netlist and Kraftwerk [8] is employed for placement.

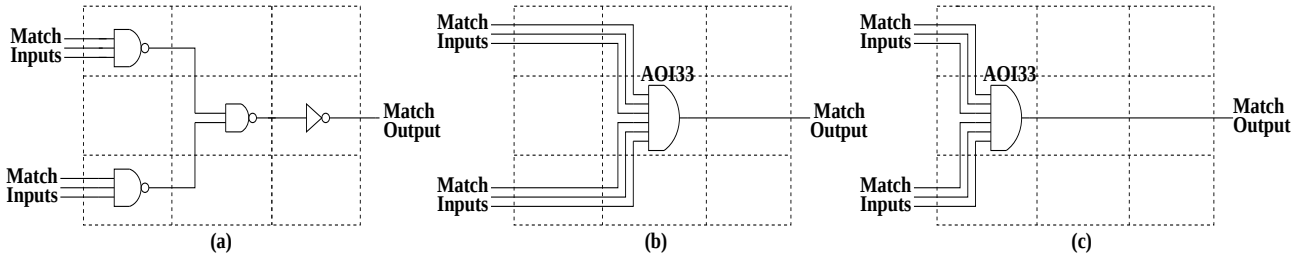

Figure 2: Mapping choices: (a) Sub-optimal area and required tracks = 12, (b) Area-optimal and tracks required = 20, (c) Area-optimal and tracks required = 15.

nets subsumed by a match, its fanin nets and its fanout nets. Specifically, it is given by,

$$cost^C_{Match} = \Sigma\, cost^C_{net\_created} - \Sigma\, cost^C_{net\_subsumed} \qquad (1)$$

where, $cost^C_{Match}$ is the congestion cost of the match, $cost^C_{net\_created}$ ($cost^C_{net\_subsumed}$) is the congestion cost of the nets created (subsumed) by the match. For example, for a 3-input NAND match shown in Figure 3(b) corresponding to the subject graph shown in Figure 3(a), the congestion cost is as follows:

$$cost^C_{Nand3} = C_{w1'} + C_{w2'} + C_{w3'} + C_{w6'} - (C_{w1} + C_{w2} + C_{w3} + C_{w4} + C_{w5} + C_{w6}) \qquad (2)$$

The nets $w1'$, $w2'$, $w3'$, and $w6'$ correspond to the new location of the match and the fanins and fanouts of the match; we compute the new location of a match as the center of gravity of the locations of its fanin and fanout gates. Multi-terminal nets are modeled using cliques for the congestion computation, and congestion contribution of each edge is scaled by a factor of $2/n$, where n is the number of edges.

The congestion cost of a wire depends on the route and the congestion in the bins that the route passes through. Probabilistically, all of the routes in the bounding box of the net are assumed to be equally possible[1] [5]. If a congestion (say 0.4) in a bin in the bounding box of the net is small as compared to the threshold congestion

[1] This assumption may not always be true. Typically, routers try to minimize vias and therefore, for two terminal nets only L and Z routes are considered. Such information can be taken into account while generating the congestion map.

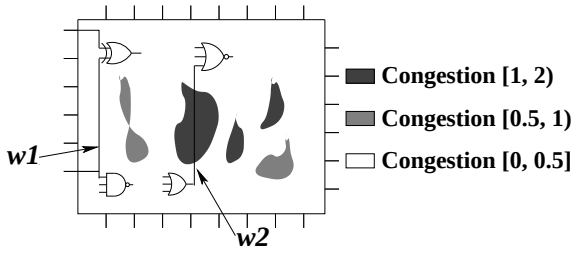

Figure 4: Context-dependent congestion cost for the wires.

(say 1.0, for instance), then the congestion contribution of that net for that bin is assumed to be 0. This is because a small value of the congestion metric corresponds to the availability of numerous tracks, and the routability of the net through the bin is unaffected. However, if the bin is congested, then the probabilistic congestion contribution of the net to that bin must be considered as its routability is hampered. In case of Figure 4, wires *w1* and *w2* will have different congestion costs even though the shortest routes in both the cases may have the same length; the congestion cost of *w1* will be zero and the congestion cost of *w2* will have a positive value as its bounding box contains congested bins. The following equation captures this causality relation between routability and congestion while computing the congestion cost of a net, $cost^C_{net}$,

$$cost^C_{net} = \Sigma_{\{Bin \in BoundingBox(net): C(Bin) > C_{max}\}} C^{Bin}_{net} \qquad (3)$$

where $C(Bin)$ is the congestion in a bin, C_{max} is the threshold

Example	script	# Cells	Placer	congestion after/before mapping				Correlation	
				Max. H	Max. V	Ave. H	Ave. V	H	V
C432	rugged	257	Capo	1.27/1.45	1.46/1.99	0.41/0.47	0.48/0.65	0.91	0.90
C432	algebraic	237	Capo	1.22/1.15	1.38/1.6	0.37/0.35	0.44/0.51	0.97	0.96
C432	boolean	375	Capo	1.04/1.55	1.42/1.68	0.43/0.45	0.51/0.67	0.95	0.94
C432	speedup	265	Capo	1.08/1.22	1.25/1.5	0.34/0.41	0.40/0.55	0.92	0.91
C6288	rugged	2311	Capo	1.73/1.34	1.88/2.00	0.69/0.57	0.81/0.82	0.85	0.86
C6288	algebraic	2275	Capo	1.37/1.79	1.55/2.20	0.50/0.73	0.60/0.98	0.76	0.78
C6288	boolean	2329	Capo	0.89/0.85	1.05/1.32	0.40/0.40	0.48/0.66	0.75	0.71
C6288	speedup	4182	Capo	1.11/1.10	1.34/1.39	0.41/0.48	0.51/0.66	0.78	0.81
C7552	algebraic	1521	Kraftwerk	2.60/2.70	2.70/2.40	0.61/0.71	0.66/0.71	0.81	0.76
C7552	rugged	2060	Kraftwerk	2.04/2.05	2.27/2.26	0.65/0.69	0.71/0.79	0.64	0.68
C7552	boolean	1582	Kraftwerk	2.23/2.50	2.50/2.00	0.61/0.74	0.66/0.71	0.82	0.83
C7552	espresso	1457	Kraftwerk	1.68/2.10	1.85/2.20	0.64/0.69	0.69/0.79	0.73	0.65
C6288	algebraic	2528	Kraftwerk	1.60/1.48	1.05/1.35	0.52/0.61	0.58/0.64	0.77	0.76
C6288	rugged	2391	Kraftwerk	1.50/2.00	2.00/2.00	0.53/0.62	0.58/0.63	0.63	0.62
C6288	boolean	2583	Kraftwerk	1.49/1.79	1.61/1.82	0.47/0.54	0.53/0.57	0.64	0.70
C6288	espresso	2549	Kraftwerk	1.76/1.79	2.06/2.09	0.52/0.62	0.59/0.66	0.61	0.64
IDC	rugged	972	Kraftwerk	1.25/1.30	1.13/1.47	0.65/0.60	0.60/0.65	0.67	0.68
IDC	algebraic	800	Kraftwerk	2.09/1.67	2.06/1.80	0.50/0.47	0.53/0.45	0.70	0.61
IDC	boolean	1622	Kraftwerk	1.75/1.78	1.52/1.23	0.57/0.59	0.64/0.65	0.67	0.66
IDC	espresso	2233	Kraftwerk	1.89/1.93	2.17/2.24	0.51/0.55	0.56/0.55	0.75	0.74

Table 1: Congestion comparison for the netlists before and after technology mapping. Max. (Ave.) corresponds to maximum (average), while H (V) corresponds to horizontal (vertical).

congestion, and C_{net}^{Bin} is the congestion due to the specific net within the bin. It is easily seen that this definition filters out the contributions of uncongested bins from the congestion cost.

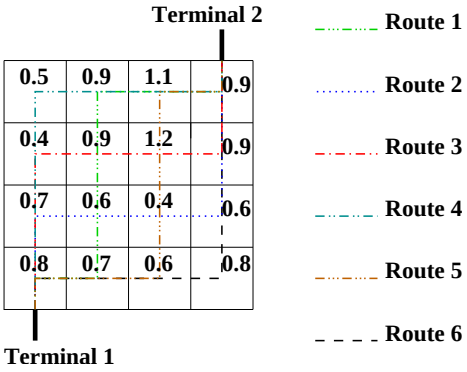

Figure 5: Computing the congestion cost of a wire probabilistically as in [5].

The bounding box for a two-terminal net is shown in Figure 5. It contains 16 bins, and the congestion value associated with each bin is shown in the figure. For the net connecting terminals 1 and 2, six possible L- and Z-shaped routes are shown for the purpose of illustration[2]. To compute the congestion cost, if the threshold value of congestion (C_{max}) is set to 1.0, then we consider only the congested bins for which congestion value is greater than 1.0, i.e., bins for which the congestion metric is 1.1 and 1.2. Three routes (route 1, 4, and 5) pass through the bin with congestion 1.1, while two routes (route 3 and 5) pass through the bin with congestion 1.2. Assuming all the routes to be equally possible, the demand (the ratio of number of paths passing through the bin to the total number of paths) for tracks in the latter bin is $\frac{2}{6}$. Similarly, the demand for tracks in the former bin is $\frac{3}{6}$. Using the definition 2.1, congestion contribution of the net for these bins can be computed by dividing the demands by the number of available tracks (N_{Tracks}). Employing Equation 3, the congestion cost of the net is given by

$$cost_{net}^{C} = \frac{1}{N_{Tracks}} \times (\frac{2}{6} + \frac{3}{6}) \quad (4)$$

The congestion cost for a match can be calculated from that of its incident nets. A positive cost implies that it may increase the congestion beyond the threshold value in some bins, while a negative cost implies that it may decrease the congestion in some of the bins where congestion exceeds the threshold value.

4.3 Description of the algorithm

Algorithm 1 CongestionMatch(Match1(C_1, A_1, D_1), Match2(C_2, A_2, D_2)).

```
if (C1 == C2) then
  if (A1 < A2)||((A1 == A2)&&(D1 < D2)) then
    return Match1;
  else
    return Match2;
  end if
end if
if (C1 < C2) then
  return Match1;
else
  return Match2;
end if
```

Algorithm 1 shows the pseudo-code for choosing the best match at a node during the matching phase of the mapping algorithm. The triplet (C_i, A_i, D_i) denotes the congestion cost, area cost, and delay cost associated with match i. The function CongestionMatch() is called for every match at a node during the matching phase to decide the best one to be stored at the node. The congestion cost is given priority over the area and delay only in congested regions, and area-optimal matches will be chosen for the nodes in the sparsely congested regions, as stated by the following proposition.

PROPOSITION 1. *If bins in bounding boxes of all of the nets,*

[2]In practice, we use probabilistic congestion estimates that consider river routes as well.

corresponding to all of the matches at a node, have congestion values that are smaller than the threshold congestion, then an area-optimal match will be stored as the best match at that node.

PROOF. This is a direct consequence from the fact that the congestion cost for all nets corresponding to all of the matches for such a case is zero from Equation (3), and the pseudocode shows that under this scenario, the area-optimal match is always chosen. □

REMARK 1. The above result is important for congestion-aware mapping, since previous work in [4] has shown that the traditional way of considering the cost, ($K_1 \times$ Area $+ K_2 \times$ Wirelength) during technology mapping requires different values of K_2 in the different regions in the circuit as a single value of K_2 fails to capture the importance of congestion in different regions. Choosing a single value of K_2 may correspond to the case in which entire circuit is uniformly congested with a single congestion value. In reality, the congestion in the circuit varies continuously from 0 to 1, or is even >1, while the routability changes in a discrete manner: in case of a bin with congestion value > 1, at least, some nets are detoured, or are unroutable, while routability of all the nets is unaffected when the congestion for the bin is <1. Assigning the congestion cost to the nets in the congested bins accounts for this discrete nature of routability and also allows the mapper to select area-optimal matches in the sparsely congested regions. Both of these purposes are critical and are served by our algorithm, while previous approaches [2, 3] have not addressed these.

The time complexity of our congestion-aware technology mapping is almost unchanged from that of a conventional technology mapping. The congestion cost computation of a match takes $O(|Nets_{Match}| \times N_{Bins})$, where $|Nets_{Match}|$ is the number of nets associated with a match and N_{Bins} is the number of bins over entire layout; $N_{Bins}$is a constant for a given layout. Therefore, congestion cost computation takes $O(|Nets_{Match}|)$ time, the same as that required for structural matching used in the SIS mapper [9].

Pre-routed blockages in the design can be incorporated into our congestion cost by reducing the appropriate number of tracks in the corresponding bins. Most placers are adequate at handling blockages. Therefore, subject graph nodes or mapped cells are not placed in blocked areas. While long wires may require repeaters that are not visible in the subject graph, observe that these buffers do not change the congestion cost.

4.4 Limitations of the algorithm

Since this technology mapping procedure is applied to tree structures after the initial subject graph generation and the decomposition of DAG's into trees, the algorithm does not have any control over high fanout nets, or over the fanout nets created due to matches at the roots[3] of the trees. The congestion due to these high fanout nets is controlled by the structure of initial network and fanout optimization. The effectiveness of the congestion-aware mapper proposed here is influenced by the scripts used for technology independent optimization, technology decomposition, and fanout optimization after technology mapping.

In our current implementation, we do not update the congestion map dynamically during technology mapping. However, this update can be easily carried out during the covering phase, thus allowing a more accurate selection of the best match stored at a node, provided multiple congestion-aware matches are stored in addition to an area-optimal one.

[3] All of the nodes in the tree have a fanout of 1 but for the root.

5. EXPERIMENTAL RESULTS

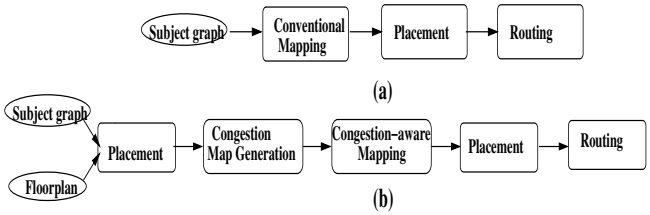

Figure 6: Conventional and congestion-aware mapping flows.

The probabilistic congestion estimation algorithm from [5] and the congestion-aware technology mapper were implemented in C/C++ and incorporated in SIS [9]. The subject graphs were created by running *script.rugged* followed by *tech_decomp -o 2* in SIS [9]. We present a set of experimental results obtained using a force-directed quadratic placer, Kraftwerk [8], and a proprietary industrial maze router. The experimental flow used in our experiments is as shown in Figure 6. For congestion-aware mapping, a subject graph was first created and placed using Kraftwerk. The congestion map for the subject graph was then generated and used in our congestion-aware mapper. After technology mapping, the circuits were placed using Kraftwerk followed by global routing using proprietary router. In all of our experiments, a bin-size of $4.8 \times 4.8 \mu\text{m}^2$ was used.

Table 2 shows the post-routing results obtained using our in-house Kraftwerk placer and maze router for conventionally mapped and congestion-aware netlists. Technology mapping is performed employing a proprietary cell library used in high-performance microprocessor designs. Our experiments use a 90nm technology and allow the router to use 4 metal layers[4]: metal 1 with no preferred direction, metals 2 and 4 for the horizontal direction, and metal 3 for the vertical direction. The entries of the form 'a / b' in the Columns 3 through 7 mean 'a' ('b') corresponds to conventionally (congestion-aware) mapped netlist. The block area shown in Column 2 is used for both of these netlists for the benchmarks shown in Column 1. Since the same block area is used for both the netlists, there is no area penalty. Columns 3, 4, and 5 show the average row utilization, the overflow after global routing, and the number of bins with congestion more than 1.0, respectively, while Columns 6 and 7 show the maximum and average congestion, respectively. For small benchmarks such as C1355, C432, and C880, a small number of bins are congested in the conventionally mapped netlists while none of the bins is congested in the congestion-aware mapped netlists. This shows that congestion problem for a few bins can be easily resolved by congestion-aware mapped netlist without any area penalty. C499 and C1908 show zero routing track overflows, while other small benchmarks have only a few congested bins, indicating that routing congestion is not a critical problem for designs up to a few hundred cells. As the design size grows beyond a thousand cells, routing congestion starts becoming a critical problem, as indicated by increased track overflows for benchmarks such as IDC, C6288, and C7552. In these cases, the congestion-aware mapped netlists have been able to reduce the track overflows by 87%, 43%, and 29% while the number of congested bins has decreased by 81%, 65%, and 25%, respectively. Based on the increase in average congestion for all of the benchmarks, accompanied by a

[4] While 90 nm and subsequent process generations have a large number of metal layers, the upper layers are usually reserved for global signal, global clock and power distributions, leaving block synthesis to operate in the lower layers [12].

Circuit	Area	Row utilization	Overflow	Congested bins	Congestion	
	μm^2	%		#	Maximum	Average
IDC	6919	63 / 70	83 / 10	32 / 6	1.3 / 1.2	0.53 / 0.60
C432	1728	66 / 69	1 / 0	1 / 0	1.1 / 0.9	0.35 / 0.37
C499	2618	64 / 73	0 / 0	0 / 0	0.9 / 1.0	0.34 / 0.40
C6288	16920	61 / 68	32 / 18	20 / 7	1.3 / 1.3	0.49 / 0.52
C7552	17633	61 / 67	655 / 461	258 / 193	1.3 / 1.3	0.65 / 0.65
C1355	2380	68 / 79	2 / 0	1 / 0	1.3 / 0.9	0.35 / 0.43
C1908	2457	68 / 78	0 / 0	0 / 0	0.8 / 0.9	0.34 / 0.40
C880	2534	71 / 82	4 / 1	2 / 1	1.3 / 1.2	0.42 / 0.48
Average		65 / 73	97 / 61	39 / 25	1.16 / 1.08	0.43 / 0.48

Table 2: Comparison of area-oriented mapping with congestion-aware mapping. Placement and routing was performed using an in-house force-directed placer and router for a 90nm technology.

reduction in the number of congested bins and the number of track overflows, we see that congestion-aware mapping tends to map the logic so as to distribute the congestion from densely congested regions to the sparsely congested regions.

For large benchmarks, the wiring distributions obtained after global routing showed significant improvements as a result of our congestion-aware technology mapping flow. The improvement in the wiring distribution is best exemplified by a reduction in the incidence of detours on the routes, where we define the detour of a route as the difference between its actual length and the total size of its minimum spanning tree (MST[5]).

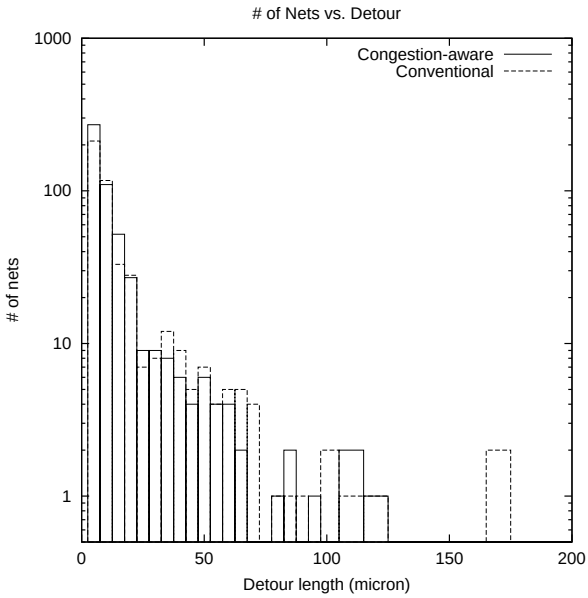

Figure 7: Number of nets vs. detour length (μm) for the IDC circuit.

Figure 7 shows plot of the number of nets vs. detour for the benchmark IDC. Similar wire distribution plots were obtained for other benchmarks. In the figure, the log-scale Y-axis shows the number of nets, while the X-axis shows the detour, in μm, for all the nets on a linear scale. The height of a dashed (solid) bar in the figure represents the number of nets in the conventional (congestion-aware) netlist for a given detour range. It can be observed that

[5]Because of the canonicity of MST's, MST estimates are used to compute the detours even though they tend to be overestimates as compared to minimum Steiner estimates.

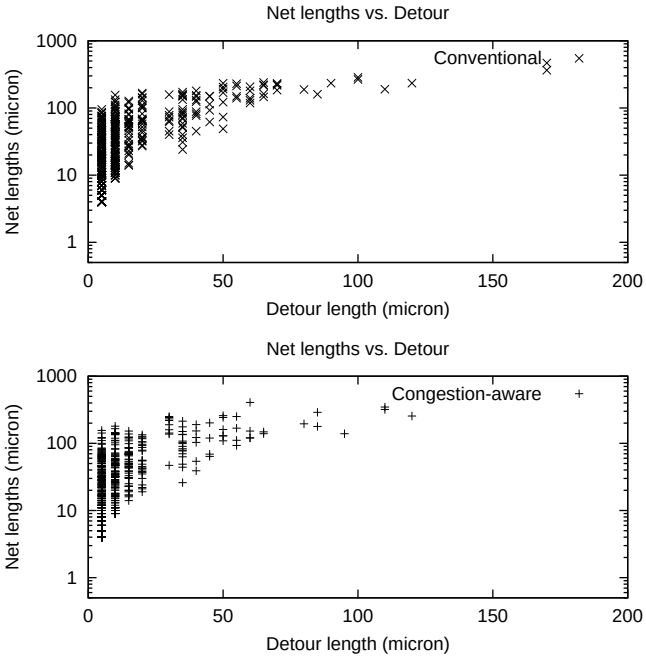

Figure 8: Scatter plots of net-lengths vs. detour length (μm) for the IDC circuit.

for shorter detour ranges number of nets in the congestion-aware netlist dominates their conventional counterpart, while as the detour length increases, the number of nets from the conventional netlist dominates that in congestion-aware netlist. Although the total number of wires increases in the congestion-aware case, most of this increase occurs at short wire lengths, as seen from the figure.

Figure 8 shows plot of net-length vs. detour length for all the nets in congestion-aware and conventionally mapped netlist for IDC. In the figure, the symbols '+' and '$\times$' indicate the actual length, in μm, of a net belonging to the corresponding detour range, in μm, specified on the X-axis, for the congestion-aware and conventionally mapped netlist, respectively. In the figure, a '$\times$' corresponding to 230 μm on the Y-axis and in the column for 70 μm on the X-axis implies that there is a net of length 230 μm whose detour length lies between 67.5 to 72.5 μm in the conventional netlist. It can be seen from the figure that the conventional netlist tends to have longer detours than the congestion-aware netlist, especially on its longer wires. The congestion-aware technology mapping not only tends to reduce the length of the long wires, but also tends to route them

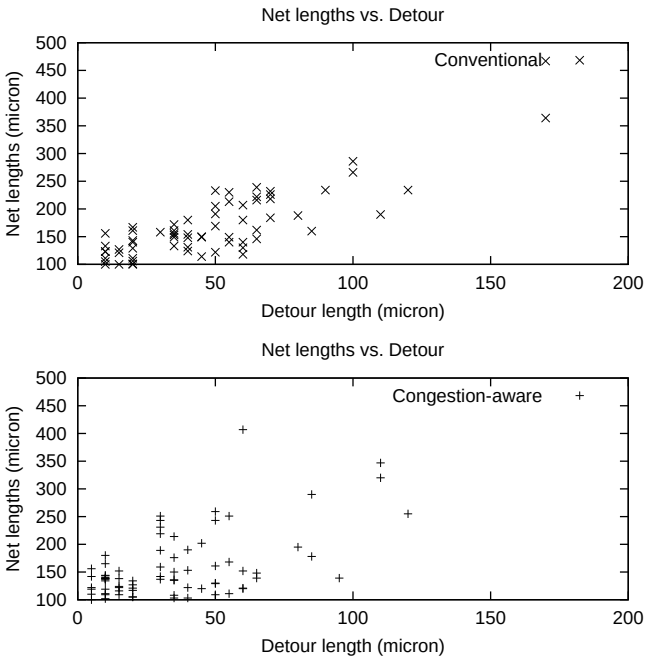

Figure 9: Scatter plots of net-length vs. detour length for long ($>$ 100 μm) nets in the IDC circuit.

with smaller detours (hence, making them more predictable prior to the routing).

Figure 9 shows the nets whose length is greater than 100 μm, since these are the nets that are usually responsible for the routing problems; '+' and '$\times$' have the same meaning as in Figure 8. Congestion-aware mapping tends to reduce the length of the longest wires, as is apparent from a larger population of '$\times$' as compared to '+' in the figure. This is achieved by allowing the shorter wires to have slightly longer detours as compared to conventional mapping. However, since the predictability of the short wires is usually not a problem, the increased detours of the short wires do not impact the design convergence adversely. Furthermore, the reduction in the detours of the wires under congestion-aware mapping also improves the predictability of their length, delay, load, and repeater requirements prior to routing.

6. CONCLUSION

We have introduced a distributed metric to predict routing congestion at the logic synthesis stage and demonstrated its fidelity to post-mapping routing congestion. Based on the congestion correlations between premapped and mapped netlists, we have performed congestion-aware technology mapping considering congestion information on the subject graph. Experimental results on a set of benchmarks show a consistent improvement in the congestion and better wiring distributions.

Acknowledgment

The first author would like to thank Cristinel Ababei from the University of Minnesota and the members of PlatoCBD™ and Quasar™ teams at Intel Corporation for their invaluable help on the design flow.

7. REFERENCES

[1] International technology roadmap for semiconductors, 2001 edition. http://public.itrs.net/Files/2001ITRS/Interconnect.pdf.

[2] L. Stok and T. Kutzschebauch. Congestion aware layout driven logic synthesis. In *Proc. ICCAD*, pages 216–223, 2001.

[3] D. Pandini, L. T. Pileggi, and A. J. Strojwas. Congestion aware logic synthesis. In *Proc. DATE*, pages 664–671, 2002.

[4] D. Pandini, L. T. Pileggi, and A. J. Strojwas. Global and local congestion optimization in technology mapping. *IEEE Trans. CAD*, 22(4):498–505, Apr. 2003.

[5] J. Lou, S. Thakur, S. Krishnamoorthy, and H. S. Sheng. Estimating routing congestion using probabilistic analysis. *IEEE Trans. CAD*, 21(1):32–41, Jan. 2002.

[6] P. Kudva, A. Sullivan, and W. Dougherty. Metrics for structural logic synthesis. In *Proc. ICCAD*, pages 551–556, 2002.

[7] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Can recursive bisection alone produce routable, placements? In *Proc. DAC*, pages 477–482, 2000.

[8] H. Eisenmann and F. M. Johannes. Generic global placement and floorplanning. In *Proc. DAC*, pages 269–274, 1998.

[9] A. Aziz, R. K. Brayton, and A. Sangiovanni-Vincentelli. Sequential synthesis using SIS. *IEEE Trans. CAD*, 19(10):1149–1162, Oct. 2000.

[10] M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Addison Wesley, Boston, MA, 3rd edition, 2001.

[11] K. Keutzer. DAGON: Technology Binding and Local Optimization by DAG Matching. In *Proc. DAC*, pages 341–347, 1987.

[12] P. Saxena, N. Menezes, P. Cocchini, and D. A. Kirkpatrick. The scaling challenge: Can correct-by-construction design help? In *Proc. ISPD*, pages 51–58, April 2003.