

Bounding the Efforts on Congestion Optimization for Physical Synthesis

Davide Pandini†, Lawrence T. Pileggi‡, and Andrzej J. Strojwas‡

†STMicroelectronics, Central R&D, 20041 Agrate Brianza, Italy

‡Department of Electrical and Computer Engineering

Carnegie Mellon University, Pittsburgh, PA 15213 U.S.A.

email: davide.pandini@st.com, pileggi@ece.cmu.edu, ajs@ece.cmu.edu

Abstract*

In this era of Deep Sub-Micron (DSM) technologies, interconnects are becoming increasingly important as their effects strongly impact the integrated circuit (IC) functionality and performance. Moreover, logic block size is no longer determined exclusively by total cell area, and is often limited by wiring resources, yet synthesis optimization objectives are focused on minimizing the number and size of library cells. Methodologies that incorporate congestion within the logic synthesis have been proposed in the past. However, in [15] and [16] it was demonstrated that predicting the true congestion prior to layout is not possible, since different layout regions can have very different routing demands, and the effectiveness of any congestion minimization approach can only be evaluated after routing is completed within the assigned die size. In these works, congestion minimization efforts at the synthesis level are controlled by means of a global weighting factor in the technology mapping cost function. Nevertheless, due to the lack of accurate congestion models, during logic synthesis it is not possible to estimate a priori which values of the congestion minimization factor will yield a congestion-free synthesized netlist. In this paper, we derive practical bounds, which limit the search space for an optimal congestion minimization factor that produces a routable netlist within fixed floorplan constraints. Although we believe that a top-down single-pass congestion-aware logic synthesis is not going to work in general, the bounds obtained in this work can be used in a practical and robust congestion minimization methodology, which can be implemented into any commercial design flow.

Categories and Subject Descriptors

B.6.3 [Logic Design]: Design Aids - Automatic synthesis, Optimization.

B.7.2 [Integrated Circuits]: Design Aids - Layout, Placement and routing.

General Terms

Algorithms, Performance, Design.

Keywords

Logic synthesis, Technology mapping, Physical design, Routing congestion, Optimization.

* This work was supported by the Central R&D of STMicroelectronics.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GLSVLSI'03, April 28-29, 2003, Washington, DC, USA.
Copyright 2003 ACM 1-58113-677-3/03/0004...\$5.00.

1 Introduction

In DSM technologies interconnects play a crucial role in the overall performance of VLSI systems [1] and their impact must be carefully considered in order to satisfy the design constraints during all phases of the top-down design flow [2][3][4], which broadly speaking consists of logic synthesis and physical design. While the timing closure problem has been extensively studied in academia [5][6][7], and several commercial flows have been proposed by the EDA industry, of equal importance is the impact of wiring on defining the block or chip size, in particular for logic synthesis, which traditionally has attempted to minimize cell area in order to minimize the size of a block or IC, without considering the interconnect area. However, since the DSM wirelength trends do not scale with feature sizes, cell area minimization can no longer guarantee overall block or die area minimization. As a result, although the total active area of the synthesized netlist can be made smaller than the block area, the design may not be routable within the assigned chip or block size. In physical design, the required routing resources are modeled by congestion, and placement and routing can sometimes fix, or avoid, potential congestion problems [8][9]. However, attempting to solve such problems at the physical design stage is too late, since the optimization potentials of layout tools are limited by the structure of the synthesized netlist. Ideally, area minimization which includes the impact of wiring via congestion models would occur during logic synthesis, where structural modifications to the netlist are still possible. The methodology presented in [15] and [16] addressed the problem of congestion minimization during the synthesis technology dependent step, i.e., technology mapping, where it is possible to exploit placement information to capture the connectivity of the technology independent representation of the circuit, and then explore the trade-offs between area (and/or delay) and congestion minimization when a fixed amount of routing resources are available. However, in [15] it was demonstrated that excessive focus on congestion minimization during technology mapping would also yield very sub-optimal solutions in terms of both cell area (and/or delay). *A priori* estimation of the efforts on congestion minimization, along with the other traditional synthesis optimization objectives such as area and/or delay, is a very difficult problem, since not only does congestion depend on the structure of the gate level netlist, but it also depends on the amount of routing resources available (i.e., the die area and the number of metal layers), and there is a lack of accurate congestion models at the synthesis level. In this work, we derive bounds for the congestion minimization efforts during technology mapping, which effectively restrict the search space for an optimal weighting factor that controls the impact of congestion minimization with respect to the other synthesis optimization objectives.

This paper is organized as follows. In Section 2 previous work in layout-driven synthesis is overviewed, and in Section 3 the congestion-aware technology mapping approach is described. In Section 4 the bounds that restrict the congestion minimization efforts during the synthesis phase of the design flow are obtained, while in Section 5, results showing the effectiveness of such

bounds in reducing the search space for routable synthesized netlists are presented. Finally, our conclusions are summarized in Section 6.

2 Previous Work

Significant progress, both in academia and industry, has been made to provide logic synthesis with physical information. In [10], Pedram and Bhat integrated technology mapping with a companion placement to include wiring contribution in area and delay minimization. Other works describe attempts at modeling routability during logic extraction in the synthesis technology independent step, either by using topological information such as the fanout range [11], or a companion placement of the logic network nodes [12]. More recently, two approaches which address the congestion problem during logic synthesis were presented by Kutzschebauch and Stok [13][14], where congestion was taken into account by means of the physical coordinates obtained from placing the technology independent netlist. However, these works do not consider that different layout regions can have very different routing demands, and more importantly, they do not consider the sub-optimality in terms of area (and/or delay) that results from introducing congestion into the synthesis optimization objectives. In contrast, in [15] it was demonstrated that the impact of congestion minimization with respect to the other synthesis optimization objectives must be carefully controlled, while in [16] the global and local nature of congestion was analyzed, and a methodology that removes local congested areas by means of incremental remapping was presented.

3 Congestion-Aware Technology Mapping

The negative effects on routability of excessive efforts on area minimization during logic synthesis have been discussed in [15], where the technology mapping algorithm proposed by Keutzer [17] and implemented in [18], was extended to include congestion among the optimization objectives. An additional term representing the interconnect cost of a match was included in the technology mapping objective function by means of the *congestion minimization factor* K (for simplicity we consider here only area minimization):

$$COST(m, v) = AREA(m, v) + K \cdot WIRE(m, v), \quad (1)$$

where $AREA(m, v)$ is the area cost of the match m at vertex v of the subject tree [17], and $WIRE(m, v)$ is the corresponding wire cost [15]. The critical issue in this approach is the impact of congestion minimization into (1), controlled by the factor K , which introduces a perturbation in the cost function, thus resulting in area (and/or delay) penalty. Uncontrolled congestion minimization may yield a very suboptimal solution in terms of cell area (and/or delay), and consequently an unroutable netlist within the assigned chip size. In contrast, if the impact of the interconnect cost in (1) is negligible, then the mapped netlist is structurally very similar to the minimum area solution ($K = 0.0$), which may be structurally unroutable, as it was shown in [15]. However, the approach proposed in [13] did not address this critical problem. Instead, the results reported in [15] support the claim that excessive (or negligible) weight on congestion minimization may yield unroutable netlists within the fixed die size.

4 Estimating the Bounds for the Congestion Minimization Factor

In [15] it was demonstrated that it is not possible to estimate *a priori* one global value for the congestion minimization factor K , which yields a routable synthesized netlist. However, it was also

demonstrated that there exists a range for the factor K , where by including the wire cost into the technology mapping cost function and controlling its impact with respect to the other synthesis optimization objectives, it is possible to synthesize structurally less congested netlists within fixed floorplan constraints. Consequently, for such range there exist the lower and upper bounds, i.e., K_{min} and K_{max} . When $K \leq K_{min}$, the structure of the mapped netlist is not significantly different from the minimum area netlist, since the wire cost has a negligible impact on the technology mapping cost function (1). On the other hand, when $K \geq K_{max}$, the wire contribution dominates the other synthesis optimization objectives, thus introducing a large cell area penalty.

Given an unbound netlist of base gates, all the nets belong to the set $N = \{n_1, \dots, n_N\}$. Since in our technology mapping approach the network DAG is partitioned into subject trees [17], every multifanout net is split into a number of two-point source-sink nets. The two-point net set $\tilde{N} = \{\tilde{n}_1, \dots, \tilde{n}_p\}$ has cardinality p bounded by the relation:

$$N \leq p = \sum_{i=1}^N |fanouts(n_i)| \leq N \cdot NetMaxFanouts, \quad (2)$$

where the function $fanouts(n_i)$ yields the fanouts for net n_i , and $NetMaxFanout = \max_{1 \leq i \leq N} (|fanouts(n_i)|)$. The nets in \tilde{N} are sorted by their non increasing length according to the source-sink Manhattan distance, using the geometrical coordinates obtained from the initial placement of the technology independent network. The operator $L1$ computes the Manhattan length for a two-point net, and the following relation holds:

$$L1(\tilde{n}_j) \geq L1(\tilde{n}_{j+1}) \quad 1 \leq j \leq p-1,$$

thus we can define $\Delta Net_{min} = L1(\tilde{n}_j) - L1(\tilde{n}_{j+1}) > 0$, such that:

$$L1(\tilde{n}_j) - L1(\tilde{n}_{j+1}) \leq L1(\tilde{n}_k) - L1(\tilde{n}_{k+1})$$

where $(1 \leq k \leq p-1) \wedge (j \neq k)$. The computational complexity of this step, given by the sorting algorithm, is $O(p \log p)$, where p is expressed by (2). A similar non increasing order sorting procedure, with respect to the cell area, is performed on the match set $M = \{m_1, \dots, m_M\}$, whose cardinality is the number of minimum area instances of the library cells. Hence, the following relation holds:

$$A(m_j) \geq A(m_{j+1}) \quad 1 \leq j \leq M-1,$$

where the operator $A(m_i)$ returns the area of the library cell corresponding to match m_i . The minimum area difference between two matches is defined as: $\Delta Match_{min} = A(m_j) - A(m_{j+1}) > 0$, such that:

$$A(m_j) - A(m_{j+1}) \leq A(m_k) - A(m_{k+1})$$

where $(1 \leq k \leq M-1) \wedge (j \neq k)$. Typically, the number of cells in a standard cell library is much less than the number of nets in modern VLSI circuits, thus the computational complexity of this step is dominated by the net sorting procedure. In order to estimate the lower bound K_{min} , we examine the following case: minimum area cost difference between two match candidates, and maximal wire-length difference between these two candidates. Consider the following expressions:

$$AREA(m_1, v) + K \cdot WIRE(m_1, v) \quad (3)$$

$$AREA(m_2, v) + K \cdot WIRE(m_2, v) \quad (4)$$

corresponding to different matches m_1 and m_2 at the same vertex v , which include both the cell area and the wire cost, and suppose the tree covering algorithm [15] must choose one of the two solu-

tions represented in (3) and (4). The congestion minimization factor K can be expressed by:

$$K = \frac{AREA(m_2, v) - AREA(m_1, v)}{WIRE(m_1, v) - WIRE(m_2, v)} \quad (5)$$

and in order to estimate K_{min} we focus on the smallest numerator and the largest denominator as possible. The wirelength contributions in (5) are obtained by the following expression:

$$WIRE(m, v) = \sum_{v_i \in fanins(m, v)} \left(L1(v, v_i) + \sum_{v_k \in fanins(match(v_i), v_i)} L1(v_i, v_k) \right)$$

where the operator $L1(v, u)$ computes the Manhattan distance between the geometrical coordinates of vertices v and u , the function $match(v)$ yields the match at vertex v , and the function $fanins(m, v)$ returns all the fanin vertices of match m at vertex v . After the net sorting step, the following relation holds: $L1(\tilde{n}_p) \leq L1(v, u) \leq L1(\tilde{n}_1)$, $\forall v, u$, and consequently also the following relations are satisfied:

$$WIRE(m_1, v) \leq \sum_{v_i} \left(L1(\tilde{n}_1) + \sum_{v_k} L(\tilde{n}_1) \right) \leq MFI \cdot (1 + MFI) \cdot L1(\tilde{n}_1) \quad (6)$$

$$WIRE(m_2, v) \geq \sum_{v_i} \left(L1(\tilde{n}_p) + \sum_{v_k} L(\tilde{n}_p) \right) \geq mfi \cdot (1 + mfi) \cdot L1(\tilde{n}_p) \quad (7)$$

where:

$$MFI = \max_{1 \leq i \leq M} (|fanins(m_i, v)|)$$

$$mfi = \min_{1 \leq i \leq M} (|fanins(m_i, v)|)$$

respectively. The minimum area difference between two matches is given by $\Delta Match_{min}$, and the following relation holds true:

$$0 < \Delta Match_{min} \leq AREA(m_2, v) - AREA(m_1, v). \quad (8)$$

Hence, the lower bound K_{min} can be obtained by using relations (6) (7), and (8):

$$K_{min} = \frac{\Delta Match_{min}}{MFI \cdot (1 + MFI) \cdot L1(\tilde{n}_1) - mfi \cdot (1 + mfi) \cdot L1(\tilde{n}_p)}. \quad (9)$$

In contrast, the estimation of the upper bound K_{max} is more complicated. The maximum area cost difference between two mappings is given by the difference between the area of the unbound netlist and the minimum area solution. The minimum wire cost difference between two different matching solutions is bounded by ΔNet_{min} , and K_{max} can be expressed as:

$$K_{max} = \frac{UnboundNetlistArea - MinimumAreaMapping}{\Delta Net_{min}}.$$

However, this upper bound is quite loose and not of practical interest, since from all our experimental results, a few percents of cell area penalty with respect to minimum area may yield an unroutable netlist within the assigned tight floorplan constraints. In order to obtain an upper bound of practical interest, we should estimate a K_{max} value, which effectively restricts the search space, and at the same time does not exclude potentially routable netlists. This is a difficult task, since at this stage we only have the initial placement

* Ideally, in order to restrict the search space for less congested netlists, the bound estimation should occur before technology mapping. The geometrical coordinates of the subject tree vertices correspond to the physical location of the technology independent netlist initial placement [15].

of the technology independent netlist, and have not performed any matching and covering as yet. The function $AREA(m, v)$ not only depends on the fanins of the match at vertex v , but it also depends on the logic depth of vertex v , i.e., its distance from the leaf vertices. In order to obtain a practical estimation of the upper bound we propose the following expression:

$$\tilde{K}_{max} = \frac{A(m_1) \cdot (1 + MFI) - A(m_M) \cdot (1 + mfi)}{\Delta Net_{min}}, \quad (10)$$

which in practice has performed well on all our experiments. It is worth pointing out that K_{min} and K_{max} have two different behaviors. When the minimum area solution yields an unroutable netlist within the fixed die size, by increasing K we include congestion among the synthesis optimization objectives. Therefore, small changes with respect to the structure of the minimum area netlist may yield a routable solution, while keeping the cell area penalty limited. However, since the cell area penalty impairs the effectiveness of this approach, the lower bound K_{min} must be as small as possible. Therefore, by estimating K_{min} with (9), we obtain a lower bound where for $K \leq K_{min}$ the netlist structure is not significantly different from the minimum area solution, and if the minimum area netlist is unroutable, then all the netlists obtained with $0 < K \leq K_{min}$ are also unroutable. On the other hand, as K increases, the structure of the mapped netlist changes more significantly, but the associated cell area penalty increment may soon yield unroutable netlists, such that when $K \geq \tilde{K}_{max}$, the netlist is unroutable within the fixed floorplan constraints. The estimation of both K_{min} and \tilde{K}_{max} can be performed efficiently, thus effectively restricting the search space for a routable mapped netlist using the cost function (1) within the range:

$$K_{min} < K < \tilde{K}_{max}. \quad (11)$$

Not every K within range (11) guarantees a routable solution, as it was shown in [15]. However, when the minimum area netlist is unroutable, the netlists obtained with $0 < K \leq K_{min}$ are unroutable as well, thus reducing the search space. Routable solutions may exist for $K \geq \tilde{K}_{max}$, but since a few cell area penalty percents may exceed the available routing resources under tight floorplan constraints, the upper bound expressed by (10) works well in practice.

5 Experimental Results

The bounds for the congestion minimization factor K expressed by (9) and (10) have been applied to the routability results obtained on the IWLS93 benchmarks SPLA and PDC, with the congestion-aware technology mapping algorithm presented in [15]. For completeness, the routability ranges are reported in Table 1. In

IWLS93 benchmark	K_{min}	K_{max}
SPLA	0.00025	0.005
PDC	0.0001	0.01

Table 1. Routability range obtained in [15]

our experiments we have considered all the single-output combinational library cells up to four inputs of a commercial cell library in 0.18 μ m technology, thus $mfi = 1$, and $MFI = 4$ respectively. The library cell sizes used for the bound estimation are reported in Table 2, and yield: $\Delta Match_{min} = A(ND2HS) - A(IVHS) = 4.096$.

Library Cell	IVHS	ND2HS	NR4HS
Area (μ m ²)	8.192	12.288	32.768

Table 2. Library cell sizes

The net length values* from the initial placement of the technology independent netlists are reported in Table 3, and the results for

IWLS93 benchmark	Net Maximum Length	Net Minimum Length	ΔNet_{min}
SPLA	92352	128	64
PDC	98048	128	66

Table 3. Net length for bound estimation

K_{min} and \tilde{K}_{max} obtained from (9) and (10) are shown in Table 4. These bounds effectively restrict the search space for a K value which yields a routable solution. Moreover, it is important to note that according to the routability results reported in [15], no routable solution exists outside the range expressed by relation (11).

IWLS93 benchmark	K_{min}	\tilde{K}_{max}
SPLA	0.00000222	2.304
PDC	0.00000209	2.234

Table 4. Bounds obtained from (9) and (10)

In order to further limit the search space for less congested netlists, we would ideally generate a tighter lower bound, i.e., increase the value of K_{min} , and a tighter upper bound, i.e., decrease the value of K_{max} . By examining the net length distributions of the technology independent representations of several circuits, we observed that most nets are short ones, and the few long nets are localized in the distribution tail. As a consequence, a tighter estimation of the congestion minimization factor K can be obtained in practice by neglecting both the shortest and the longest nets. Our heuristic does not consider all the net lengths with a number of instances less than 5% of the peak distribution, that corresponds to the net length with the maximum number of instances. We obtain a restricted net length range, as shown in Table 5, where most net

IWLS93 benchmark	$NetLen_{min}^R$	$NetLen_{max}^R$	Net instances within the restricted length range
SPLA	128	2624	81.1%
PDC	128	2688	79.3%

Table 5. Net length restricted range

instances are within this range. The net length values reported in Table 5 can be used to derive tighter bounds for the factor K , using the following equations:

$$K_{min}^R = \frac{\Delta Match_{min}}{MFI \cdot (1 + MFI) \cdot NetLen_{max}^R - mfi \cdot (1 + mfi) \cdot NetLen_{min}^R} \quad (12)$$

$$K_{max}^R = \frac{A(m_1) \cdot (1 + MFI) - A(m_M) \cdot (1 + mfi)}{NetLen_{max}^R - NetLen_{min}^R} \quad (13)$$

where with respect to (11) the following relation holds:

$$K_{min} < K_{min}^R < K_{max}^R < \tilde{K}_{max}$$

The restricted bounds of the congestion minimization factor K obtained with (12) and (13) using the values of Table 5, are reported in Table 6. Our heuristic has significantly reduced the search space for a less congested netlist, and at the same time, according to the results presented in [15] and summarized in Table 1, the routability range has been fully captured.

* The net length values reported in Table 3 and in all the other Tables in Section 5 are expressed in DBU, i.e., Database Units, and are consistent with all the experimental results reported in [15].

IWLS93 benchmark	K_{min}^R	K_{max}^R
SPLA	0.000078	0.059
PDC	0.000076	0.058

Table 6. Restricted K bounds

6 Conclusions

The wiring congestion is a very important design factor which directly impacts the time-to-market of complex System-on-Chip designs, and must be considered both globally in logic synthesis, and locally in physical design. However, traditional synthesis focused on area minimization may produce structurally unroutable circuits, and hence yield larger logic block areas due to wiring congestion. While global congestion can be effectively addressed during technology mapping, the suboptimality obtained by including congestion into the synthesis optimization objectives must be carefully evaluated. In this paper we have derived practical bounds for controlling the congestion minimization impact during logic synthesis, with respect to the other optimization objectives like area and/or delay. Our results demonstrate that such bounds effectively restrict the search space for synthesizing structurally more routable circuits within fixed floorplan constraints. In conclusion, our approach yields a practical methodology, which incorporates congestion directly within the synthesis multi-objective optimization.

References

- [1] H. B. Bakoglu, Circuits, Interconnections and Packaging. Reading, MA: Addison Wesley, 1990.
- [2] S. Hojat and P. Villarrubia, "An Integrated Placement and Synthesis Approach for Timing Closure of PowerPC Microprocessors," in *Proc. 1997 ICCD*, Oct. 1997, pp. 206-210.
- [3] L. T. Pileggi, "Achieving Timing Closure for Giga-Scale IC Designs," in *Proc. 1999 Intl. Symp. on Timing Issues*, Mar. 1999, pp. 25-28.
- [4] P. Gopalakrishnan, A. Odabasioglu, L. T. Pileggi, and S. Raje, "An Analysis of the Wire-Load Model Uncertainty Problem," *IEEE Trans. on Computer-Aided Design*, vol. 21, pp. 23-31, Jan. 2002.
- [5] L. N. Kannan, P. R. Suaris, and H. G. Fang, "A Methodology and Algorithms for Post-Placement Delay Optimization," in *Proc. 1994 Design Automation Conf.*, Jun. 1994, pp. 327-332.
- [6] A. H. Salek, J. Lou, and M. Pedram, "An Integrated Logical and Physical Design Flow for Deep Submicron Circuits," *IEEE Trans. on Computer-Aided Design*, vol. 18, pp. 1305-1315, Sept. 1999.
- [7] J. Cong, "Timing Closure Based on Physical Hierarchy," in *Proc. 2002 ISPD*, Apr. 2002, pp. 170-174.
- [8] G. Meixner and U. Lauther, "Congestion-Driven Placement Using a New Multi-Partitioning Heuristic," in *Proc. 1990 ICCAD*, Nov. 1990, pp. 332-335.
- [9] P. N. Parakh, R. B. Brown, and K. A. Sakallah, "Congestion Driven Quadratic Placement," in *Proc. 1998 Design Automation Conf.*, Jun. 1998, pp. 275-278.
- [10] M. Pedram and N. Bhat, "Layout Driven Technology Mapping," in *Proc. 1991 Design Automation Conf.*, Jun. 1991, pages 99-105.
- [11] H. Vaishnav and M. Pedram, "Minimizing the Routing Cost During Logic Extraction," in *Proc. 1995 Design Automation Conf.*, Jun. 1995, pp. 70-75.
- [12] M. Pedram and N. Bhat, "Layout Driven Logic Restructuring/Decomposition," in *Proc. 1991 ICCAD*, Nov. 1991, pp. 134-137.
- [13] T. Kutzschebauch and L. Stok, "Congestion Aware Layout Driven Logic Synthesis," in *Proc. 2001 ICCAD*, Nov. 2001, pp. 216-223.
- [14] T. Kutzschebauch and L. Stok, "Layout Driven Decomposition with Congestion Consideration," in *Proc. 2002 DATE*, Mar. 2002, pp. 672-676.
- [15] D. Pandini, L. T. Pileggi, and A. J. Strojwas, "Congestion-Aware Logic Synthesis," in *Proc. 2002 DATE*, Mar. 2002, pp. 664-671.
- [16] D. Pandini, L. T. Pileggi, and A. J. Strojwas, "Understanding and Addressing the Impact of Wiring Congestion During Technology Mapping," in *Proc. 2002 ISPD*, Apr. 2002, pp. 131-136.
- [17] K. Keutzer, "DAGON: Technology Binding and Local Optimization by DAG Matching," in *Proc. 1987 Design Automation Conf.*, Jun. 1987, pp. 341-347.
- [18] E. Detjens, G. Gannot, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang, "Technology Mapping in MIS," in *Proc. 1987 ICCAD*, Nov. 1987, pp. 116-119.