Interconnection Length Estimation for Optimized Standard Cell Layouts

Massoud Pedram
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720
Bryan Preas
Computer Science Laboratory
Xerox Palo Alto Research Center, Palo Alto, CA 94304

Abstract

In this paper, we present an accurate model for prediction of interconnection lengths for standard cell layouts. On the designs in our test suite the estimates are within 10% of the actual layouts. Our model abstracts the important features of placement, global routing and channel routing. The predicted results are obtained from analysis of the net list. No prior knowledge of the functionality of the design is used. Accurate prediction of interconnection length is useful for estimating the actual layout area, for evaluating the fit of a logic design to a fabrication technology, and for studying placement and routing algorithms.

1 Introduction

Interconnection analysis addresses two related problems: the wire (interconnection) length and distribution problem and the wiring area estimation problem. Many researchers have addressed the latter problem and good area estimation techniques are available. However, the wire length and distribution problem has not been solved satisfactorily. Interconnection length studies tend to be theoretical and hence not applicable to specific designs, empirical such as those relating Rent's rule parameters to the average wire length, or based on experience with previous design layouts. Early research into interconnection length estimation, although of theoretical interest, is too general to be useful for specific designs. Later work, which produces results that have the appropriate level of detail, requires knowledge of *Rent's exponent* or assumes particular wire length distributions. In practice assumptions about wire length distributions are either not verified or require fitting curves to the actual layout data.

A method which describes the wire length distribution as a function of the specific logic design before layout is needed [5]. Good physical design of large systems requires accurate area estimates of the individual modules for area planning, optimal placement, and routability predictions. This requirement encourages development of procedural models which include characteristics of the physical design processes, structural description of the logic design, and physical features of primitive cells. These models produce interconnection length estimates with high accuracy (without making arbitrary wire length distribution assumptions or fitting curves to the data).

Interconnection length models have many uses. They can evaluate the capability of a new fabrication technology. The models determine routability of the proposed logic design, subject to the constraints of the technology, and therefore, help the system designers trade off aspects of the design and the technology. Accurate interconnection length estimates are beneficial since they measure the quality of placement and global routing algorithms. Interconnection estimates are also useful during the technology mapping phase of the logic synthesis since they can predict the cost of various implementations.

The area required for interconnections within a circuit layout largely depends on the total length of wire that must be accommodated. Therefore, accurate estimation of total interconnection length is an essential part of any area estimation procedure. In fact, many area estimation techniques require the wire length for the logic design as an input parameter [4,7]. To obtain accurate area estimates, it is necessary to achieve high accuracy in estimating the wire length. This task is accomplished by the *procedural* model presented in this paper. Our model captures the characteristics of the physical design processes (placement, global routing and detailed routing) and the structural features of the logic design in order to accurately estimate interconnection length.

In this paper, we present an accurate interconnection length estimator for optimized standard cell layouts. Since interconnection length is a very strong function of a logic design, the first task is to extract relevant features which account for the wiring requirements of the logic design. We introduce a metric which captures the local influence of other nets over a net under consideration. We have concluded that this is a more pertinent and effective metric (as far as interconnection length estimation is concerned) compared to other metrics such as average pin per cell or average number of connected cells to any cell in the design.

The outline of the method is as follows. For each size of net (as measured by the number of pins), we compute the expected number of cells whose placement and wiring (to a first approximation) influence the optimal wiring of the net under consideration. By considering possible aspect ratios of the rectangle enclosing all such cells and considering various configurations of pins on the net and averaging over all such states, we compute the expected wire length and feedthrough count of the net. Summing over all nets, we obtain the total interconnection length and the total number of feedthroughs in the design.

Our wire length model relies on knowledge of the actual design processes (placement, global routing and detailed routing), and physical structures. The predicted results are obtained from analysis of the net list. No prior knowledge of the functionality of the design is used. The model considers multi-pin nets directly, and does not preprocess them into sets of 2 pin nets (as is often the case). Using these wire length estimates, the chip width and height can be computed by statistical area estimation [6,7,9] or by random offset track packing technique introduced in Section 4.

This paper is organized as follows. Section 2 gives an overview of the procedural approach to interconnection analysis, states our assumptions, and gives a high level description of our computational model. The details of our computational model of a optimized circuit layout process and the area estimation technique are presented in Sections 3 and 4. Results and complexity analysis are presented in Section 5, and conclusion is given in Section 6.

2 Model Overview

The inputs to the interconnection length prediction model are the logical design specification, the primitive cells invoked by the specification, and the fabrication technology. Following the standard cell topology double entry cells are placed in rows and interconnected in routing channels among the rows. The outputs of the model are the estimated total wire length, the wire length distribution, the estimated total number of feedthroughs, and the feedthrough distribution.

A standard cell layout is modeled as a regular, $w \times n$ array, where n is the number of rows and w is the average number of cells per row. Wires follow rectilinear paths, with horizontal segments on one layer (*metal1*) and vertical segments on another (*metal2*). The average cell width is computed from the cells actually used in the design. Our interconnection model assumes a placement optimization process, a minimal rectilinear Steiner tree global router and a left edge channel router. Our approach captures the important characteristics of these *algorithm classes* rather than those of specific algorithms. The features of the algorithm classes which are captured by our interconnection model are the following. The placement optimizer minimizes the sum over all the nets of the half perimeter length of the rectangle enclosing pins of each net. Pins inside the placement bounding box for the net are not optimized for that net.

The global router approximates a minimal rectilinear Steiner tree to connect pins on each net. We assume that wiring for a net

does not meander outside the bounding box defined by the pins on the net, that feedthroughs are placed at the intersections of cell rows and the edges in the Steiner tree connecting pins on the net, that no feedthrough is added to a row which contains a pin on the net, and that each net contributes at most one feedthrough to each cell row. This global routing paradigm tends to produce minimal *metal2* routes at the expense of more *metal1* routing. (See [8] for an opposing global routing paradigm that produces maximal *metal2* routing.) We assume a channel router which finds the shortest path inside the channel to connect pins on the net. The route does not meander outside the box enclosing these pins. Inside the channel each net is connected with trunks with no overlap along the length of the channel. Branches connect trunks to the pins. In addition, we assume that all branch layer conflicts can be resolved by adding horizontal jogs. Over-the-cell routing is not considered.

We estimate interconnection length and feedthrough count for each size of net and then sum the contribution over all the nets. *Metal1* wire length is expressed in units of average cell pitch and *metal2* wire length is expressed in units of average channel height. We compute the average interconnection lengths by spatially restricting the possible positions of the pins on the net to a bounding box within the $w \times n$ grid. Considering feasible aspect ratios for this bounding box and various pin configurations within the box and averaging over all such states, we compute the average interconnection lengths and feedthrough count for the net. By summing over all nets, the total interconnection length and the total number of feedthroughs is computed. There is no explicit dependence on a particular cell library or fabrication technology for estimation of wire length. However, such information is required when the total interconnection length is used to estimate the chip width and height. This work builds on the procedural model presented in [9] by including an optimized placement model, improved routing models and introducing a more accurate area estimation technique.

3 The Interconnection Length Model

The interconnection structure of a design is characterized by net *neighborhood populations* (NPs) which account for the local influence of other nets over a net in question. The NP for a net is the number of primary input/outputs (IOs) and cells which are at distance zero or one from the net. To compute the NP for a particular net, we find all the cells and IOs connected by this net (i.e. at distance 0 from the net). We then follow every other net which is connected to the cells until we visit the cells and primary IOs that are at distance one from the net. The NP for the net is the total number of *distinct* cells and IOs encountered in this manner. In the NP computation, we ignore nets which connect more than 25% of the cells in the design. (These are typically power and clock nets that go everywhere.) This procedure is repeated for all nets of given size resulting in the average neighborhood population for each size of net. At the end, $NP[d]$ contains the average neighborhood population for nets with d pins.

The NP for a net reflects the conflicting demands on a placement optimizer that is attempting to optimally place the cells directly connected to the net. To clarify this notion, assume that the placement optimizer is seeking a placement of d cells connected by exactly one net. The optimizer will cluster these cells in a bounding box of minimum half perimeter length. However, in reality, it is not possible to place cells connected to each net in such a minimum length bounding box due to competition from other nets. The placement and routing of the cells directly connected to a net of size d (to a first approximation) is influenced by a cell and IO population of size $NP[d]$.

We describe our abstraction of the placement optimizer. Consider a d-pin net with pins on a two-dimensional, $w \times n$ grid. The d pins on the net can be placed within an $x \times y$ bounding box where y ranges from k_1 to k_2 and $x = \lceil NP[d]/y \rceil$. k_1 is given by $\lfloor NP[d]/XSpan(d,n) \rfloor$ and k_2 is equal to the $\lceil YSpan(d,n) \rceil$ where $XSpan(d,n)$ and $YSpan(d,n)$ are the the expected cell span and the expected row span of the net if the net pins are randomly placed on the $w \times n$ grid. Due to the placement process which minimizes the half perimeter length of the rectangle enclosing all pins on the net, and due to conflicting demands of other nets, we consider the d pins to be uniformly distributed inside the $x \times y$ bounding box. Now,

$$W(x,y) = \frac{(w-x+1) \times (n-y+1)}{x \times avgCellWidth + (\gamma \times (y-1)+y) \times cellHeight}$$

where the numerator gives the count of all feasible subgrids of size $x \times y$ in a grid of size $w \times n$, and the denominator gives the half perimeter length of the $x \times y$ grid. γ is the ratio of the expected channel height to the cell height.

The average length of the net with d pins is given by

$$M1Length(d) = \frac{\sum_{y=k_1}^{k_2} W(x,y) \times M1LengthInBox(d,y)}{\sum_{y=k_1}^{k_2} W(x,y)}$$

where $M1LengthInBox(d,y)$ is the expected length of the net if it is restricted to $x \times y$ bounding box and is computed as follows

$$M1LengthInBox(d,y) = \sum_{i=1}^{Min(d,y)} (\frac{1}{y})^d \times \binom{y}{i} \times M1LengthContrib(i,d)$$

The first term gives the probability of placing d pins on some subset of y rows. The next term gives the number of ways we could select i rows from among y rows, and $M1LengthContrib(i,d)$ gives the contribution of a d-pin net occupying exactly i rows ($i \leq d$) to the $M1LengthInBox(d,y)$.

In order to compute $M1LengthContrib(i,d)$, we examine all different configurations (groupings) of d pins on i rows. We solve an integer equation whose solution is a list of *pin sets*. (See [9].) Each pin set represents a possible distribution of pins on rows. For example, if $i = 3$, $d = 6$, then solution to $Sets(i,d)$ is $((1,1,4),(1,2,3),(2,2,2))$. Now,

$$M1LengthContrib(i,d) = \sum_{sets} i! \times \frac{d!}{\prod_{k=1}^{d} rows[k]! \times \prod_{k=1}^{i} pins[k]!} \times ConfigLength(i,set)$$

where the first term gives the number of row permutations, and the second term gives the number of *distinguishable* pin-to-row assignments. Here, $rows[k]$ is the number of rows with k pins, and $pins[k]$ is the number of pins on the kth row.

Next, we compute the $ConfigLength(i,set)$ which captures some effects of a minimal rectilinear Steiner tree global router. It gives the expected length of the net when the net assumes the configuration described by a particular pin set. In addition, we must address the *channel sharing* problem, i.e., given a particular pin configuration what is the probability of these pins facing the same channel. This issue is important because pins on two adjacent rows can be connected within the shared channel. We compute $Sharing(y,i,l)$ which gives the probability of pins which are placed on i out of y rows sharing exactly l channels ($l \leq i/2$) by a recursive procedure. Then,

$$\begin{aligned} ConfigLength(i,set) = \sum_{r=0}^{i/2-1} ((\sum_{s=0}^{r} Sharing(y,i,s)) \times \\ (WireLength(pins[2r+1]) + WireLength(pins[2r+2])) + \\ (1 - \textstyle\sum_{s=0}^{r} Sharing(y,i,s)) \times \\ WireLength(pins[2r+1] + pins[2r+2])) + \\ (i/2-1) \times WireLength(2) + \\ (if\ IsOdd(i)\ then\ WireLength(pins[i]+1)\ else\ 0) \end{aligned}$$

$WireLength(m)$ gives the expected length of m-pin portion of the net when all m pins lie on one channel ($2 \leq m \leq 2x$) and is given in [9].

We compute sum of the lengths of *metal2* wires connecting all pins of the net as follows:

$$M2Length(d) = \frac{\sum_{y=k_1}^{k_2} W(x,y) \times M2LengthInBox(d,y)}{\sum_{y=k_1}^{k_2} W(x,y)}$$

$$M2LengthInBox(d,y) = \sum_{i=1}^{Min(d,y)} (\frac{1}{y})^d \times \binom{y}{i} \times B(i,d) \times (ChanSpan(i,y) + (d-i)/2)$$

$$B(i,d) = i^d - (\sum_{j=1}^{i-1} \binom{i}{j} \times B(j,d))$$

where $B(1,d) = 1$, and $B(i,d)$ gives the number of ways of placing d pins on exactly i rows. $ChanSpan(i,y)$ is the expected number of channels spanned by a net whose pins occupy i rows and is given in [9].

Next, we compute the expected number of feedthroughs added to all rows, $FTHeight(d)$, and to the central row, $CFeedthrus(d)$, by a net of size d as follows:

$$FTHeight(d) = \frac{\sum_{y=k_1}^{k_2} W(x,y) \times FTHeightInBox(d,y)}{\sum_{y=k_1}^{k_2} W(x,y)}$$

$$FTHeightInBox(d,y) = \sum_{i=1}^{Min(d,y)} (\frac{1}{y})^d \times \binom{y}{i} \times B(i,d) \times NumFeedthrus(i,y)$$

$$CFeedthrus(d) = \frac{\sum_{y=k_1}^{k_2} W(x,y) \times CFeedthrusInBox(d,y)}{\sum_{y=k_1}^{k_2} W(x,y)}$$

$$CFeedthrusInBox(d,y) = \frac{\sum_{i=Max(1,y-(n+1)/2)}^{Min(y,(n+1)/2)} PFTOnRow(i,y)}{(n-y+1) \times \sum_{i=1}^{y} PFTOnRow(i,y)} \times FTHeightInBox(d,y)$$

where $PFTOnRow(i,y)$ is the probability that a feedthrough will cross row i, and $NumFeedthrus(i,y)$ is the expected number of feedthroughs which are added to rows by a net whose pins occupy i rows. Expressions for both functions are given in [9].

The total *metal1* and *metal2* lengths required to connect all nets, the total number of feedthroughs crossing all rows, *totalFeedthrus*, and those crossing the central row, *cFeedthrus*, are obtained by summing the individual net contributions over all nets. For example,

$$totalM1Length = \sum_{nets} nets[d] \times M1Length(d)$$

where $nets[d]$ represents the number of nets with d pins.

4 The Area Estimation Technique

Given average wire length and wire length distribution, we could exploit known statistical area estimation techniques to estimate the total chip area and aspect ratio [6,7,9,11]. Instead, we introduce a new, more consistent technique based on random offset track packing to model the detailed routing process.

The *metal1* length for each net is divided equally into a number of segments as determined by the expected number of wire segments (trunks) for each size of net. The number and lengths of all segments for each size of net lying in each channel are given by

$$segmentsInChannel[d] = \lceil nets[d] \times Segments(d) \rceil / (n-1)$$
$$segmentLength[d] = \lceil M1Length[d] / Segments(d) \rceil$$

$$Segments(d) = \frac{\sum_{y=k_1}^{k_2} W(x,y) \times SegmentsInBox(d,y)}{\sum_{y=k_1}^{k_2} W(x,y)}$$

$$SegmentsInBox(d,y) = \sum_{i=1}^{Min(d,y)} (\frac{1}{y})^d \times \binom{y}{i} \times B(i,d) \times (\sum_{j=0}^{i/2} (i-j) \times Sharing(y,i,j))$$

Now, the *track packing* problem (in the absence of vertical and horizontal constraints) can be defined as follows: Given t segments which must be placed in tracks of equal length w and given that segment i requires l_i units of track length, the objective is to determine the minimum number of tracks needed to accommodate all segments. This is the well known bin packing problem and is NP-complete. There exist many heuristics which obtain packings that use a "small" fraction of tracks more than the optimal packing.

The simplification made by assuming that no pin constraints exist on the wire segments causes underestimation of the routing area. We remedy this by generating a uniformly distributed offset for each wire segment in the channel. We could build a horizontal constraint graph for these randomly positioned wire segments. The assignment of tracks to wire segments corresponds to the proper coloring of this constraint graph (which is by construction an interval graph). In the absence of vertical constraints, there exist simple optimal algorithms for coloring the interval graph. Our task, however, is much easier because we are only interested in computing the density of the channel which can be accomplished by a plane sweep algorithm. The total density of the standard cell layout (*totalDensity*) is the sum of channel densities over all channels. Ignoring vertical constraints in our area estimation model produces small errors because modern dogleg routers often route channels at density. This random offset track packing technique is a generalization of the statistical technique presented in [9].

Finally, chip width, chip height, and actual *metal1* and *metal2* lengths (in μmeters) are computed as follows:

$$avgCellPitch = avgCellWidth + cFeedthrus \times ftWidth/w$$
$$avgChanHeight = totalDensity \times trackSpacing/(n-1)$$
$$chipWidth = w \times avgCellPitch$$
$$chipHeight = n \times cellHeight + (n-1) \times avgChanHeight$$
$$actualM1Length = totalM1Length \times avgCellPitch$$
$$actualM2Length = totalFeedthrus \times cellHeight + totalM2Length \times avgChanHeight$$

5 Experimental Results

We implemented our interconnection model in the Cedar language running on Xerox Dorado workstations (2-MIPS machines) and incorporated the model into the DATools system developed at Xerox PARC [1]. Table 1 summarizes the examples used to test the predictions. The counter and the adder are circuits synthesized by the DATools system when no performance requirements are imposed. The adders are simple ripple-carry designs, the counters are carry-look-ahead designs. The *RSD* is part of a Reed-Solomon error correction circuit, and the *SnooperCtl* is part of a cashe controller. Primary1 and Primary2 are the benchmarks from the physical design workshop [10]. The placement is obtained by TimberWolfSC version 4.1 [12]. The global and detailed routers are discussed in [2,3].

Table 2 summarizes the experiments comparing our wire length, area and aspect ratio estimates with the actual results obtained after global and detailed routing of the circuits. Our estimates are all within 10% accuracy. We verified the detailed characteristics of the model by collecting data and generating statistics for the actual interconnection length and feedthrough count for each size of net and comparing it with our estimated values. Table 3 summarizes this comparison for Primary1. Detailed results for other examples and aspect ratios are comparable. Sources of error include average behavior modeling (rather than worst case behavior modeling) and incomplete characterization of the physical design processes.

We define $C(k)$ as follows:

$$C(k) = \sum_{i=1}^{k} ||Sets(i,k)||$$

Table 4 gives values of $C(k)$ for k ranging from 1 to 15. The complexity of our interconnection model is

$$O(n \times \sum_{k=2}^{d_{max}} C(k)).$$

The run time is relatively independent of the size of design but is strongly affected by the maximum size of net being considered (d_{max}). For this reason, we divide nets with more than 15 pins into cliques of smaller nets. This division introduces little error because, typically, there are few large nets. Each execution of the model requires 2-5 minutes for the example runs.

6 Conclusions

We have developed an interconnection model that predicts interconnection lengths and layout areas for standard cell layouts. These predictions are within 10% of the the actual lengths and areas over a wide range of layout aspect ratios and over a wide range of logic design characteristics. In order to achieve the required accuracy, we incorporated the important characteristics of the placement, global routing, and channel routing algorithms. This model is useful for floorplanning to generate shape constraint relations, for determining the fit of a logic design to a fabrication technology, and for evaluating placement algorithms.

References

[1] R. Barth, L. Monier, and B. Serlet, "Patchwork: layout from schematic notations," *Proc. 25th Design Automation Conf.*, pp.250-255, June 1988.

[2] J. Cong and B. Preas, "A new algorithm for standard cell global routing" in *IEEE Inl. Conf. on Computer-Aided Design*, pp. 176-180, November 1988.

[3] D. N. Deutsch, "A 'dogleg' channel router," in *Proc. 13th Design Automation Conf.*, pp. 425-433, 1976.

[4] A. El Gamal, "Two dimensional stochastic model for interconnections in master slice circuits," in *IEEE Trans. on Circuits and Systems*, Vol. CAS-28, no. 2, pp. 127-138, February 1981.

[5] D. Hanson, "Interconnection Analysis," in *Physical Design Automation of VLSI Circuits*, B. Preas and M. Lorenzetti editors, Benjamin/Cummings Publ., Menlo Park, CA, 1989, pp. 47.

[6] W. R. Heller, W. F. Mikhail, and W. E. Donath, "Prediction of wiring space requirements for LSI," in *Proc. 14th Design Automation Conf.*, pp. 32-42, June 1977.

[7] F. J. Kurdahi and A. C. Parker, "Techniques for area estimation of VLSI Layouts," in *IEEE Trans. on CAD*, Vol. 8, No. 1, Jan. 1989, pp. 81-92

[8] K. W. Lee and C. Sechen, "A new global router for row-based layout," in *IEEE Inl. Conf. on Computer-Aided Design*, pp. 180-183, November 1988.

[9] M. Pedram and B. Preas, "Accurate Prediction of Physical Design Characteristics of Random Logic," To be published in *Inl. Conf. on Computer Design*, October 1989.

[10] B. Preas, "Benchmarks for cell-based layout systems," *Proc. 24th Design Automation Conf.*, pp. 319-320, 1987.

[11] S. Sastry and A. C. Parker, "Stochastic models for wirability analysis of gate arrays," *IEEE Trans. on Computer-Aided-Design*, Vol. CAD-5, no. 1, pp. 52-65, January 1986.

[12] C. Sechen and K. W. Lee, "An improved simulated annealing algorithm for row-based placement," *Proc. IEEE Inl. Conf. on Computer-Aided Design*, pp. 478-481, November, 1987.

example	# cells	# IOs	# nets	# pins
16-b adder	144	52	177	546
SnooperCtl	95	30	114	331
RSD	210	89	211	670
64-b counter	478	130	585	1537
Primary1	750	73	903	2801
Primary2	2907	107	3029	8758

Table 1: A summary of the example circuits used for interconnection length / area estimator.

example	# rows	predicted				actual			
		m1Length	m2Length	area	aspect	m1Length	m2Length	area	aspect
SnooperCtl	5	24135	20101	8.063e5	1.11	22088	17509	7.601e5	1.04
16-b adder	8	23969	23632	8.142e5	2.662	22144	21533	7.618e5	2.50
RSD	6	59640	52288	1.696e6	1.08	62126	48339	1.610e6	0.98
64-b counter	12	226186	254601	5.837e6	1.526	238588	238243	5.361e6	1.543
Primary1	22	713923	545465	2.711e7	1.358	782690	491232	2.688e7	1.474
Primary2	34	3958229	3422324	1.085e8	1.360	4300110	3049935	1.131e8	1.476

Table 2: A comparison of estimates versus the actual results of total interconnection length (μ), area (μ^2) and aspect ratio.

# pins	estimated		actual	
	m1 length	ft count	m1 length	ft count
2	484	0.353	506	0.484
3	843	0.797	835	0.597
4	1073	1.026	1100	0.530
5	1417	1.114	1474	0.846
7	2028	1.106	2630	0.833
12	4280	0.852	5689	1.330

Table 3: Detailed comparison of *metal1* length and feedthrough count for various sizes of nets for Primary1 with 14 rows.

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C(k)	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176

Table 4: $C(k)$ values for k ranging from 1 to 15.