# Accurate Prediction of Physical Design Characteristics for Random Logic

Massoud Pedram
Department of Electrical Engineering and Computer Science
University of California, Berkeley, California 94720

Bryan Preas
Computer Science Laboratory
Xerox Palo Alto Research Center
Palo Alto, California 94304

**Abstract**

In this paper, we present an accurate model for prediction of physical design characteristics, such as interconnection lengths and layout areas, for standard cell layouts. This model produces accurate shape constraint functions (height versus width of the layout over a range of aspect ratios) by considering the logic design specification, the physical design process and the physical implementation technology. Random and optimized placements, global and detailed routing are each abstracted by procedural models that capture the important features of these processes. Equations that define the procedural model are presented. Predictions of layout characteristics that are within 10% of the actual layouts are achieved over a range of circuit functions and sizes. We have verified both the global characteristics (total interconnection length and layout area) and the detailed characteristics (wire length and feedthrough distributions) of the model. Accurate prediction of physical design characteristics are useful for floorplanning, for evaluating the fit of a logic design to a fabrication technology, and for studying placement algorithms.

## 1 Introduction

Interconnection analysis considers the logical design, physical design process and implementation technology to predict physical (layout) characteristics, such as pin utilization of packages, total interconnection length and layout area. Good physical design of large systems requires accurate estimates of the physical characteristics of the individual modules for area planning, optimal placement and routability predictions. Often the physical design process must be repeated if the estimates are not accurate. To achieve the required accuracy, proper abstractions of layout processes and physical structures are necessary. However, the appropriate level of detail is important because too much detail can result in performing the task rather than predicting the results.

Interconnection models have many uses. Good models can evaluate the capability of a new fabrication technology. Interconnection analysis can determine routability of the proposed logic design subject to the constraints of the technology, and therefore, helps the system designers trade off aspects of the design and the technology. It is also important to be able to predict whether each element of the design can be constructed in the given technology. Interconnection analysis lets the system designer predict the fit of the logical design to the implementation technology.

Chip floorplanning is a costly, time consuming task which may require several design iterations. In the process of floorplanning a large VLSI system, the floorplanning systems or the designers need to have accurate estimates of the total areas and aspect ratios of the individual modules. Having this information reduces the number of design iterations by excluding many of feasible but inferior solutions *a priori*. This improves quality of solutions. In addition to the floorplanning uses, having accurate estimates of the average interconnection length is beneficial since it is a measure of how good the placement and/or partitioning processes are.

Previous interconnection analysis efforts have failed to produce accurate (within 10%) estimates. Furthermore, in order to make the appropriate trade-offs during floorplanning, it is desirable to know the range of possible shapes (shape constraint functions) for the cells that compose a design. These issues are addressed by our procedural model which uses the structural features of the logic design and captures the characteristics of the physical design processes (placement, global and detailed routing).

In this paper, we present an accurate interconnection length and layout area predictor for standard cell layouts. For each size of net (as measured by the number of pins), we consider all possible distributions of pins on rows. By averaging over all such pin configurations, we compute interconnection lengths, feedthrough counts and channel spans of the nets. (See Figure 1.) Summing over all nets, we obtain the total interconnection length and the total number of feedthroughs crossing the rows. After computing the required number of wiring tracks and the number of feedthroughs crossing the longest row, we estimate the chip width and height. We have modeled the placement, global and detailed routing processes accurately enough to predict wire length and area within 10%.

The area predictor generates accurate shape constraint relations over a wide range of aspect ratios. We have verified our model by generating actual layouts using the TimberWolf placement, [SeSa86], and the global [CoPr88] and detailed router [Pre89] in the Xerox PARC DATools system [BaMS88]. However, the approach is more general and can incorporate other models of placement and routing.

The next section places our work in perspective through a survey of the previous research efforts in interconnection analysis. Section 3 gives an overview of the procedural approach to interconnection analysis, states

CH2794-6/89/0000/0100$01.00 © 1989 IEEE

our assumptions, and gives a high level description of our computational model. The details of our computational model for random placement is presented in Section 4. Optimized placement is briefly discussed in Section 5. A more complete description of the optimized placement model may be found in [PePr89]. Results are presented in Section 6 followed by conclusions in Section 7.

# 2 Prior Work

Interconnection analysis models are divided into three categories: empirical, theoretical, and procedural. Empirical studies produce expressions for physical characteristics by extracting information from actual designs and fitting curves to the data. Theoretical studies produce closed form expressions by making simplifying assumptions about the interconnection structure. Procedural models consider more detailed aspects of the actual design processes , physical structures and interconnection structure of the design to improve the accuracy of the predictions.[1]

## 2.1 Empirical Models

The initial work on the wiring requirements was performed by Rent in early 1960's. He derived *Rent's rule* which is a relationship between the IO count and the cell count of a design by fitting curves to the empirical data from various computer designs

$$ioCount = (averageCellSize) \times (cellCount)^r$$

where $r$ is *Rent's exponent.* Landman and Russo [LaRu71] studied the relation between cell versus IO counts and the Rent's exponent. They showed two different values of Rent's exponent must be used depending on the number of cells, that is, the circuits with larger cell counts and smaller package counts have smaller Rent's exponents. Donath [Don79] reported that values of Rent's exponent ranged as high as 0.75 for highly parallel designs and as low as 0.47 for highly serialized designs. Sastry and Parker [SaPa86] derived an interconnection length distribution that fitted actual designs.

These models require knowledge of empirical parameters (such as Rent's exponent) that are computed from actual design instances. An implicit assumption is that the design instances used in deriving the values of these parameters have the same interconnection structure and design characteristics as those of the design under consideration. This assumption limits the applicability of the empirical formulas.

## 2.2 Theoretical Models

Theoretical models produce closed form, mathematical descriptions of the physical characteristics from logic designs and physical implementation technologies. These models provide general trends but lack sufficient detail to represent individual designs accurately. They are useful when little is known about the actual design process. These models are divided into two categories: deterministic and stochastic.

Deterministic models rely on parameters extracted from actual design instances. The effects of the physical design processes are characterized by simple, measurable parameters. Donath [Don69] devised a plausible structure for a logic design which conforms to Rent's rule. He assumed a hierarchical structure where only a fraction of the pins inside a cell are connected to pins outside the cell (the "encoding" assumption). He showed that such a structure exhibits Rent's rule. He also demonstrated that a randomly constructed design does not conform to Rent's rule.

A major thrust in stochastic approaches models the interconnection characteristics of the design as a stationary process. The wiring requirements are computed by making assumptions about the probability distributions of wires. An early attempt to formalize the characteristics of computer logic designs was published by Donath [Don70]. He defined a top-down hierarchical design approach in which each step of the expansion of the hierarchy is modeled by the substitution of a pattern of interconnected cells for each block. These patterns are selected randomly from a fixed pattern library by a stochastic process. Based on this model, Donath established the relation between the cell-to-pin ratio and performance.

Heller et al., [HeMD77] addressed the problem of estimating wiring space requirements. He modeled interconnection wires as independent two-point wires originating stochastically (with a Poisson distribution) at some cell, covering a random distance (an average interconnection length) and terminating at some second cell. Based on this model, he derived the probability of wiring completion of some number of cells in a limited number of wiring tracks. His model correctly predicts the relative difficulty of wiring completion in various designs. El Gamal [ElG81] refined Heller's model. His model assumes a regular two-dimensional array of cells. The generation and length of interconnecting wires are modeled as in Heller's work. The path traveled by each wire is established randomly, with the restriction that its endpoints be separated by a Manhattan distance which is equal to the path length. El Gamal derived from this model the minimum number of wire segments, and hence the minimum wiring area required for the square array of cells. He concluded that the overall minimum wiring area is of order $N^2 log^2 N$ where all cells have been placed in an $N \times N$ array.

Sastry and Parker [SaPa86] used a model very similar to El Gamal's. They modeled interconnections as independent two-point wires covering an average length and derived expressions for channel widths, probability of routing completion, and wire lengths. They showed that wire length distribution has the form of a Weibull distribution with location and shape parameters. These parameters must be computed based on the net lengths obtained from actual layouts. Kurdahi and Parker [KuPa89] presented an area estimator for standard cell layouts. They assumed rows of equal size, double entry cells, constant pin pitch, two-pin nets and minimal rectilinear connection paths. Their model assumes *birth* of a wire at pin slot $i$ and length of a wire $l$ are independent random variables with probabilities $p_B(i)$ and $p_L(l)$. They suggested uniform distribution for $p_B(i)$ and geometric distribution for $p_L(l)$. Based on these assumptions, the required routing area is estimated. This model, however, requires knowledge of average interconnection length which is computed by fitting curves to known data.

These models, although of great theoretical interest, are too general to be useful for specific design decisions. They require knowledge of empirical parameters or hypothetical wire length distributions. Assumptions about wire length distributions are either not verified in practice or require fitting curves to the actual layout data. Many area estimators require wire lengths as input. The accuracy of the area estimates is, therefore,

[1] For a more comprehensive review, refer to [Han88].

bounded by the accuracy of interconnection length estimates. To be useful for design work, however, estimates with 10% accuracy are needed. In order to achieve this level of accuracy, proper abstractions to model layout processes and physical structures as well as careful analysis of the interconnection structure of the design under consideration are necessary. Theoretical models lack this level of detail and therefore produce results that are not accurate enough for today's design work.

### 2.3 Procedural Models

Procedural models incorporate greater detail and a lower level of abstraction compared to other models. They rely on relations derived from knowledge of the actual design processes, interconnection structure of the design, physical layouts of the leaf cells and layout rules. These models extract interconnection characteristics of the design and combine them with abstractions of the placement and routing processes to give estimates without need for arbitrary wire length distribution assumptions or empirical parameters.

Sechen [Sec87] presented an interconnection length estimator which gives accurate estimates for small designs. He assumed square cells placed on an square, two-dimensional grid. For each size of net, the half perimeter of the smallest rectangle enclosing all pins on the net is computed. Various scenarios and a look up table are used to determine all possible arrangements of cells which establish a given bounding box. Total interconnection length is then computed by summing (over all nets) the half perimeter lengths of the rectangles enclosing pins on the nets. Sechen's abstraction of the layout surface make his model most applicable to "sea-of-gates" style. His approximation of total interconnection length for nets with large number of pins ($> 4$) is not accurate enough for our purposes.[2] We implemented an interconnection length estimator based on half perimeter lengths of net bounding boxes. For the circuits in our test suite, errors up to 30-40% were observed.

Chen and Bushnell [ChBu88] introduced an area estimator for random placement with the assumption that wires do not share tracks. They derived the expected number of wiring tracks, and feedthroughs in the central row, and thereby, estimate the chip width and height. The authors do not attempt to model global and detailed routing processes, and do not differentiate between designs based on their interconnection structures. Their estimated chip area for small designs is 40-70% over the actual chip area, and the number of wiring tracks is overestimated by a factor of 2-3. No data is presented for medium or large size designs.

## 3 Model Overview

We present two interconnection models: basic and improved. The basic model features a random placement but optimized global and detailed routing. Since the random placement process can be characterized accurately, we can separate the effects of placement and routing within the overall model. The improved model extends the basic model by including optimized placement and is used in production. Optimized global and detailed routing abstractions from the basic model are retained.

[2] Chung [ChHw79] showed that the worst case length of a minimal rectilinear Steiner tree connecting $d$ pins of a net tends to be $(\sqrt{d}+1)/2$ of the half perimeter length of the smallest rectangle enclosing pins of that net.

The basic model assumes independent interconnections, a random placement process, a minimum spanning tree global router such as those described in [Rob84] and [CoPr88], and a left edge channel router, for example [Deu76] and [HaSt71]. To obtain estimates within 10%, we incorporate knowledge of the placement, global and detailed routing processes. The average interconnection length, the expected channel span and the expected number of feedthroughs for nets as a function of net sizes (number of pins on the net) are computed by considering all pin configurations (various assignments of pins to rows) and averaging over all states.[3] Knowledge of the global and detailed routing processes is used to predict the number of pins (on a net) in a channel, the feedthrough locations and the wire lengths for different pin configurations. By summing over all nets, the total *metal*1 and *metal*2 lengths and the total number of feedthroughs can be computed. Then, the chip width is estimated from width of the cell row with largest number of feedthroughs . The chip height is determined from the sum of row heights plus channel heights.

The improved model approximates the effects of a purposeful placement by spatially restricting the possible positions of $d$ pins on a net to a $x \times y$ grid ($x$ and $y$ are functions of the interconnection structure of the design and the net size). It, then, computes the average interconnection length and the expected number of feedthroughs for each size of net.

Given knowledge of standard cell layouts and model assumptions, the model equations follow logically (without reference to any empirical or arbitrary parameters).

## 4 The Basic Interconnection Model

The inputs to our area estimation model are the logical design specification and the primitive cells included in the specification. Following the standard cell model, double entry cells are placed in rows and interconnected in routing channels among the rows. The outputs of the estimation model are the estimated total wire length, wire length distribution, the estimated total number of feedthroughs, the feedthrough distribution, chip width and height and chip area.

A standard cell layout is modeled as a regular $w \times n$ array, where $n$ is the number of rows and $w$ ($= numCells/n$) is the average number of cells per row. Wires follow rectilinear paths with horizontal segments on one layer (called *metal*1 or $M1$) and vertical segments on another (called *metal*2 or $M2$). The average cell width is computed from the cells actually used in the design. The basic model assumes a random placement but optimized global and detailed routing processes. The following important aspects of the algorithms have been incorporated. The placement process uniformly distributes cells on the $w \times n$ grid. The global router finds a minimum spanning tree to connect pins of nets. We assume that wiring for a net does not meander outside the bounding box defined by the pins on the net and that feedthroughs are placed at the intersections of cell rows and the edges in the spanning tree connecting pins on the net. We assume a channel routing paradigm. Over-the-cell routing is not considered.

The assumption of independent nets allows us to compute the wire length and feedthrough contribution of each net separately. The random placement assumption implies uniform pin distribution over the layout sur-

[3] We assume that pins on a net do not share cells with other pins on the same net.

face and is captured in the $M1Length$ and $M2Length$ equations. Consider a net with $d$ pins uniformly distributed on the $w \times n$ grid. We compute sum of the lengths of $metal1$ wires connecting all pins on the net ($M1Length(d)$ in units of cell pitch) as follows:

$$M1Length(d) = \sum_{i=1}^{Min(d,n)} (\frac{1}{n})^d \times \binom{n}{i} \times M1LengthContrib(i,d)$$

The first term gives the probability of placing $d$ pins on some subset of $n$ rows. The next term gives the number of ways we could select $i$ rows from among $n$ rows, and $M1LengthContrib(i,d)$ gives the contribution of a $d$-pin net occupying exactly $i$ rows ($i \leq d$) to the $M1Length(d)$.

In order to compute $M1LengthContrib(i,d)$, we examine all different configurations (groupings) of $d$ pins on $i$ rows. We must solve an integer equation of the form

$$\sum_{j=1}^{i} x_j = d \qquad 1 \leq x_j \leq w$$

The solution to this equation returns a list of *sets*. Each set represents a *distinct* pin configuration representing pins of net on a single row. (See Figure 2.) For example, if $i = 3$, $d = 6$ , $w = 60$, then solution to the integer equation is $((1,1,4),(1,2,3),(2,2,2))$. This equation is efficiently solved by a recursive procedure. The cardinality of the solution (list of sets) strongly affects the runtime of this model since the number of solutions grows rapidly with $d$ and $i$. Therefore, we approximate results for very large nets by dividing large nets into cliques of smaller nets. Now,

$$M1LengthContrib(i,d) = \sum_{sets} i! \times \frac{d!}{\prod_{k=1}^{d} rows[k]! \times \prod_{k=1}^{i} pins[k]!} \times M1ConfigLength(i,set)$$

where the first term gives the number of row permutations and the second term gives the number of distinguishable pin-to-row assignments. (We need not consider pin permutations within the same row.) Here, $rows[k]$ is the number of rows with $k$ pins, and $pins[k]$ is the number of pins on the $k$th row. For example, if $set = (1,1,4)$, then $rows[1] = 2$ and $pins[1] = 1$.

Next, we compute the $M1ConfigLength(i,set)$ which gives the expected length on the net if it assumes the configuration of pins described by a particular set. $M1ConfigLength(i,set)$ is our abstraction of the global router and assumes that pins (on a net) on rows do not share channels with pins on other rows. It is given by

$$M1ConfigLength(i,set) = \begin{cases} WireLength(pins[1]) & \text{if } i = 1 \\ \sum_{k=2}^{i} WireLength(pins[k]+1) & \text{if } pins[1] = 1 \\ WireLength(pins[1]) + \sum_{k=2}^{i} WireLength(pins[k]+1) & \text{otherwise} \end{cases}$$

where $pins[k]$ is sorted in increasing order.[4] $WireLength(m)$ gives the expected length of the net which has $m$ pins in a routing channel ($2 \leq m \leq 2w$).

In order to compute $WireLength(m)$, we must abstract the main features of the detailed routing process. We assume a channel router which finds the shortest path inside the channel to connect the $m$ pins on the net and does not zigzag or meander outside the box enclosing these pins. The equation for $WireLength(m)$ captures these aspects of the channel router and is given by

$$WireLength(m) = \frac{\sum_{l=m}^{w} 4(w-l+1) \times \binom{2l-2}{m-2} \times l}{\binom{2w}{m}}$$

The numerator is a sum over all possible spans of the $m$ randomly placed pins on a channel with $w$ cells on each side, and the denominator is the number of ways $m$ cells can be chosen from among $2w$ cells. The first term in the numerator is the number of ways spans of $l$ cell pitches can be established within the channel, the second term in the numerator is the number of ways the remaining $m-2$ pins can be placed on $2l-2$ cells, and $l$ is the cell span established by the pins.

If we assume a single wire per track (using unity track demand factor i.e. $\forall m\ WireLength(m) = 1$), then the equation for $M1Length(d)$ reduces to that of Chen and Bushnell [ChBu88]:

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

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

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

where $B(1) = 1$ and $B(i)$, which is defined recursively, gives the number of ways of placing $d$ pins on exactly $i$ rows. Our computational model, however, looks at detailed pin distributions on $i$ rows and computes the average wire length by averaging over all states.

We compute sum of the lengths of $metal2$ wires connecting all pins of the net ($M2Length(d)$ in units of channel height) as follows:

$$M2Length(d) = \sum_{i=1}^{Min(d,n)} (\frac{1}{n})^d \times \binom{n}{i} \times M2LengthContrib(i)$$

$$M2LengthContrib(i) = B(i) \times ChanSpan(i)$$

$$ChanSpan(i) = \frac{\sum_{l=i}^{n} (n-l+1) \times \binom{l-2}{i-2} \times (l-1)}{\binom{n}{i}}$$

[4] These equations model a minimum spanning tree global router. A similar but different set of equations is used to model minimum Steiner tree global routers. Same comment applies to the equation computing $NumFTs$.

where $ChanSpan(i)$ is the expected number of channels spanned by a $d$-pin net (occupying $i$ rows). The first term in the numerator is the number of ways spans of $l$ rows can be established within the chip, the second term in the numerator is the number of ways the remaining $i-2$ rows can be chosen from among $n-2$ rows, and $l-1$ is the channel span. The denominator is the number of ways $i$ rows can be chosen from among $n$ rows.

Next, we compute the expected number of feedthroughs contributed by a $d$-pin net:

$$FTHeight(d) = \sum_{i=1}^{Min(d,n)} (\frac{1}{n})^d \times \binom{n}{i} \times FTHeightContrib(i)$$

$$FTHeightContrib(i) = B(i) \times NumFTs(i)$$

where $NumFTs(i)$ is the expected number of feedthroughs added by a net which is occupying exactly $i$ rows and is given by

$$NumFTs(i) = \frac{\sum_{l=i}^{n}(n-l+1) \times \binom{l-2}{i-2} \times (l-i)}{\binom{n}{i}}$$

The expression for $NumFTs$ is identical to that for $ChanSpan$ with $l-i$ (number of feedthroughs) replacing $l-1$ (channel span). This is because the global router does not add a feedthrough to a row which contains a pin on the net.

The total interconnection length required to connect all the nets ($totalM1Length$ and $totalM2Length$ in $\mu$meters), and the total number of feedthroughs contributed by all the nets ($totalFTs$) are

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

$$totalM2Length = \sum_{nets} nets[d] \times M2Length(d) \times avgChanHeight$$

$$totalFTs = \sum_{nets} nets[d] \times FTHeight(d)$$

where $nets[d]$ represents the number of nets with $d$ pins. Note that $avgChanHeight$ can be estimated only after total number of wiring tracks required to accommodate all interconnections is computed. Distribution of wire lengths and feedthroughs as a function of the number ($nets[d]$) or size of nets ($d$) in the logic design are computed as well.

Now that we have the $metal1$ lengths and channel spans of nets, we can complete the model of the detailed router. We assume a left edge channel router that finds the shortest path to connect pins on the net inside the channel and does not zigzag or meander outside the bounding box enclosing these pins. A single trunk is used to connect pins of any net inside the channel. In addition, we assume that all branch layer conflicts can be resolved by adding horizontal jogs. Our abstraction of the channel routing process is composed of two components: the wire length abstraction captured by $WireLength(m)$ equation given previously and the segment packing into tracks abstraction described below.

The $metal1$ length for each net is divided equally into a number of segments as determined by the expected channel span of the net.[5] The number and lengths (in units of cell pitch) of all segments of a net with $d$ pins are given by

$$numSegments[d] = nets[d] \times \lceil M2Length(d) \rceil$$

$$segmentLength[d] = \frac{M1Length[d]}{\lceil M2Length(d) \rceil}$$

Now, the segment packing problem (in the absence of vertical constraints)[6] can be defined as follows: Find the minimum possible number of tracks that accommodate all net segments such that sum of the unutilized spans of individual tracks is minimized. Here we assume that all routing is completed in one big channel. This channel is then divided into $n-1$ actual routing channels. This problem can be formulated as a constrained linear optimization problem and solved by linear programming techniques. (See [PePr89].) A simpler and more efficient, but less accurate, approach is described in the next paragraph.

Instead of computing expected segment length for a net with $d$ pins ($segmentLength[d]$), we compute average segment length (over all nets) as follows:

$$avgSegmentLength = \frac{totalM1Length}{totalSegments}$$

$$totalSegments = \sum_{segments} numSegments[d]$$

Because of random placement process, we argue that the segments in the channel originate according to a Poisson distribution with parameter $\alpha$ where

$$segmentDensity = \frac{totalSegments}{w \times (n-1)}$$

and

$$\alpha = segmentDensity \times avgSegmentLength = \frac{totalM1Length}{w \times (n-1)}$$

From this and a confidence level of $c$ (= 0.999) for routing completion, the required number of wiring tracks per channel is approximated as follows:

$$\sum_{k=1}^{numTracks} \frac{e^{-\alpha} \times \alpha^k}{k!} \leq c$$

We have computed the total number of wiring tracks required by the detailed router ($numTracks$). Now we must compute the number of feedthroughs crossing the central row. We can compute the number of feedthroughs crossing each of the rows as follows. The probability of a feedthrough crossing row $i$ (rows are numbered from top to bottom starting from 1) is given by

$$PFTInRow(i) = \frac{\sum_{j=1}^{d-1}(\frac{i-1}{n})^j \times (\frac{n-i}{n})^{d-j} \times \binom{d}{j}}{2^d - d - 1}$$

[5] A spanning tree global router tends to divide the $metal1$ wire length of a net equally among the channels spanned by the net.

[6] Since the positions of net segments are unknown, horizontal constraints are absent as well.

From the $d$ pins on the net, assume that $j$ are placed in rows above the $i$th row and $d-j$ pins are placed in rows below the $i$th row. $(i-1)/n$ is, then, the probability that one pin is placed in rows above row $i$ and $(n-i)/n$ is the probability that another pin is placed in rows below row $i$. Note that if at least one pin on the net is placed on row $i$, then $PFTInRow(i) = 0$. This is because the global router does not add feedthroughs to a row which contains some pin on the net.

We know that, for randomly placed designs, the number of feedthroughs crossing the central row is the largest. To compute the probability that a $d$-pin net will contribute a feedthrough to the central row, we set $i = n/2$ in above. Now,

$$NumFTsInRow(i) = \frac{PFTInRow(i)}{\sum_{i=1}^{n} PFTInRow(i)} \times totalFTs$$

In the above, we set $d$ to be the average number of pins per net (although the results are not sensitive to $d$).

The $avgChanHeight$ is computed as follows:

$$avgChanHeight = \frac{numTracks \times trackSpacing}{n-1}$$

Due to uniform number of cells and non-uniform number of feedthroughs per rows, there is some dead channel space at the corners of the chip that cannot be utilized by the router (Figure 1). This area is accounted for as follows:

$$areaCorrection = 2 \times avgChanHeight \times \sum_{j=1}^{\lceil \frac{n}{2} \rceil - 1} (ChanWidth(\lceil \frac{n}{2} \rceil) - ChanWidth(j))$$

$$ChanWidth(j) = w \times avgCellWidth + NumFTsInRow(j) \times ftWidth$$

Finally, chip width and height are computed as follows:

$$chipWidth = w \times avgCellWidth + NumFTsInRow(\lceil \frac{n}{2} \rceil) \times ftWidth$$

and

$$chipHeight = n \times cellHeight + numTracks \times trackSpacing + \frac{areaCorrection}{chipWidth}.$$

# 5 The Improved Interconnection Model

The improved interconnection model builds on basic model by including the characteristics of placement optimization. We retain the global routing and the channel routing features of the basic model. The placement optimizer minimizes the sum over all the nets of the half perimeter of the rectangle enclosing pins of each net. Pins inside the placement bounding box for the net are not optimized for that net.

The effects of a purposeful placement on the interconnection structure are captured by *net neighborhood populations* which account for the local influence of the other nets over the net in question. For each size of net, the neighborhood population is the number of distinct primary input/outputs (IOs) and distinct cells which border the net. To a first approximation, placement of the net in question is mostly affected by cells in the immediate neighborhood of the net. Therefore, the placement characteristics of each size of net can be estimated by the size of its average neighborhood population. An overview of the interconnection length/area estimation techniques for optimized placement follows. See [PePr89] for more detail.

We compute the average interconnection length and the expected number of feedthroughs for nets with $d$ pins by spatially restricting the possible positions of the $d$ pins on the net to a $x \times y$ subgrid within the $w \times n$ grid. The size of this bounding box is different for each size of net and is computed from the average neighborhood population for that net. Due to the objective of the placement process which only 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 located inside this bounding box. By considering all feasible aspect ratios for this bounding box and all different pin configurations within the box, and averaging over all such states, we compute the expected interconnection length and the expected number of feedthroughs for the net. By summing over all nets, the total interconnection length and the total number of feedthroughs are computed. From these, the total width and height of the layout are computed.

# 6 Experimental Results

We implemented our interconnection length and circuit area predictor models in the Cedar language running on Xerox Dorado workstations (2-MIPS machines) and incorporated the model into the DATools system developed at Xerox, PARC [BaMS88]. Table 1 summarizes the examples used to test the predictions. The counters and the adders are synthesized by the DATools system. The *RSD* is part of a Reed-Solomon error correction circuit. The Primary1 is one of the benchmarks from the physical design workshop [Pre87].

Tables 2 and 3 summarize the experiments comparing our area and interconnection length estimates with the actual results. Our area estimates are within 10% accuracy. We also verified the model by collecting detailed statistics about the actual interconnection lengths and feedthrough heights as a function of the number of pins on the nets and comparing this information with our estimated values. The close agreement demonstrates the robustness of our computational model. Table 4 summarizes this comparison for Primary1. Figure 3 shows the predicted and actual shape constraint function ($chipHeight$ versus $chipWidth$) for Primary1. The runtime is relatively independent of the size of design but is strongly affected by the maximum number of pins per net. For this reason nets with more than 18 pins are divided into cliques of smaller nets. Each execution of the model requires 30-60 seconds for the examples run.

We briefly note some sources of error. A large portion of our 10% error budget arises from the fact that we operate on average behavior rather than worst case behavior. Although large scale features of the actual layouts (that is, layout area and aspect ratio, total $metal1$ and $metal2$ lengths and total feedthrough counts) remain relatively constant, the detailed wire length and feedthrough distributions as a function of number of pins on nets vary as much as 20-40% from one layout of the same circuit to the next. Our estimates of these two distributions are close to the average over several

layouts. Another source of error is the incomplete characterization of the physical design processes. We do not capture some aspects of the processes. For example, our random placement abstractions do not include the interdependence of nets, our global routing abstractions exclude an improvement global routing phase targeted toward reducing local congestions, and our channel router abstractions ignore vertical constraints among various net segments in the channel. Some aspects of the physical design processes we capture only partially. For example, $ConfigLength(i, set)$ does not completely capture a spanning tree global router and the statistical technique presented previously to compute number of tracks is only accurate to a first approximation.

# 7 Conclusions

We have developed an interconnection model that predicts interconnection lengths and layout areas for standard cell layouts. The procedural model abstracts the important features of the physical design processes for standard cell layout (placement, global and detailed routing). The equations that define this model are based on the functions performed by the design processes rather than on unsubstantiated statistical distributions or on arbitrary parameters. We extract the relevant features (interconnection structure and leaf cell layouts) of the logic design to provide parameters for the equations. The detailed information (wire length and channel span distributions) abstracted from the logic design allow us to transform the two-dimensional area estimation problem into two one-dimensional problems. Technology independence is achieved through parametrizing the layout design rules. These predictions are within approximately 10% of the actual lengths and areas over a wide range of layout aspect ratios and over a range of logic design characteristics. The prediction process is very efficient; it takes no more than 60 seconds to analyze each of the circuits posed as examples in this paper. 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


[BaMS88] Barth, R., Monier, L., Serlet, B., "Patchwork: layout from schematic notations," in *Proc. 25th Design Automation Conf.*, pp. 250-255, June 1988.

[ChBu88] Chen, X., Bushnell, M. L. "A module area estimator for VLSI layout," in *Proc. 25th Design Automation Conf.*, pp. 54-59, June 1988.

[ChHw79] Chung, F. R. K. and Hwang, F. K., "The largest minimal rectilinear Steiner trees for a set of n points enclosed in a rectangle with given perimeter," in *Networks*, vol. 9(1979) pp. 19-36.

[CoPr88] Cong, J. and Preas, B. T.' 'A new algorithm for standard cell global routing" in *IEEE Inl. Conf. on Computer-Aided Design*, pp. 176-180, November 1988.

[Deu76] Deutsch, D. N., "A 'dogleg' channel router," in *Proc. 13th Design Automation Conf.*, pp. 425-433, 1976.

[Don69] Donath, W. E. "Hierarchical structure of computers," in *IBM T. J. Watson Research Center Report RC 2392*, March 1969.

[Don70] Donath, W. E. "Stochastic model of the computer logic design process," in *IBM T. J. Watson Research Center Report RC 3136*, November 1970.

[Don79] Donath, W. E. "Placement and average interconnection lengths of computer logic," in *IEEE Trans. on Circuits and Systems*, vol. CAS-26, no. 4, pp. 272-277, April 1979.

[ElG81] El Gamal, A. "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.

[Han88] Hanson, D. "Interconnection Analysis", in *Physical Design Automation of VLSI Circuits*, Benjamin/Cummings Publ., Preas, B. and Lorenzetti M., editors, 1989, pp. 47.

[HaSt71] Hashimoto, A., and Stevens, J., "Wire routing by optimizing channel assignment within large apertures," in *Proc. 8th Design Automation Workshop*, pp. 155-163, 1971.

[HeMD77] Heller, W. R. Mikhail, W. F. and Donath, W. E. "Prediction of wiring space requirements for LSI," in *Proc. 14th Design Automation Conf.*, pp. 32-42, June 1977.

[KuPa86] Kurdahi, F. J. and Parker, A. L. "Techniques for area estimation of VLSI layouts," in *IEEE Trans. on Computer-Aided Design*, vol. CAD-8, no. 1, pp. 81-92, January 1989.

[LaRu71] Landman, B. S. and Russo, R. L. "On a pin versus block relationship for partitions of logic graphs," in *IEEE Trans. on Computers*, C-20, no. 12, pp. 1469-1479, December 1971.

[PePr89] Pedram, M. and Preas, B. T. "Interconnection length estimation for optimized standard cell layouts," to be published in *IEEE Inl. Conf. on Computer-Aided Design* (1989).

[Pre87] Preas, B. T. "Benchmarks for cell-based layout systems," in *Proc. 24th Design Automation Conf.*, pp. 319-320, 1987.

[Pre89] Preas, B. T. "An algorithm to resolve cyclic vertical constraints in channel routing," to appear.

[Rob84] Roberts, K. A., "Automatic layout in the Highland system," in *Proc. of ICCAD* (1984) pp. 224-226.

[SaPa86] Sastry, S. and Parker, A. C. "Stochastic models for wirability analysis of gate arrays," in *IEEE Trans. on Computer-Aided Design*, vol. CAD-5, no. 1, pp. 52-65, January 1986.

[Sec87] Sechen, C. "Average interconnection length estimation for random and optimized placements," in *IEEE Inl. Conf. on Computer-Aided Design*, pp. 190-193, November 1988.

[SeSa86] Sechen, C., Sangiovanni-Vincentelli, A. "TimberWolf 3.2: a new standard cell placement and global routing package," in *Proc. 23rd Design Automation Conf.* pp. 432-439, June 1986.

| Ex. | # cells | # IOs | # nets | # pins |
|---|---|---|---|---|
| 8-bit adder | 72 | 26 | 89 | 210 |
| 16-b counter | 121 | 53 | 153 | 438 |
| RSD | 210 | 89 | 211 | 670 |
| 64-b adder | 576 | 194 | 707 | 2178 |
| Primary1 | 750 | 73 | 903 | 2801 |

Table 1: A summary of the example circuits used to verify interconnection length and layout area estimator.

| Example | # rows | predicted | | | actual | | | area |
|---|---|---|---|---|---|---|---|---|
| | | width | height | aspect ratio | width | height | aspect ratio | error |
| 8-bit adder | 4 | 672 | 1112 | 1.67 | 656 | 1140 | 1.74 | 0.77% |
| 8-bit adder | 6 | 594 | 1542 | 2.60 | 584 | 1584 | 2.65 | 0.01% |
| 16-bit counter | 6 | 1193 | 1748 | 1.47 | 1160 | 1692 | 1.46 | 6.24% |
| 16-bit counter | 8 | 1085 | 2064 | 1.90 | 1000 | 2156 | 2.16 | 3.87% |
| RSD | 4 | 2080 | 1898 | 0.86 | 2160 | 1972 | 0.91 | 7.32% |
| RSD | 8 | 1568 | 2833 | 1.81 | 1576 | 2844 | 1.80 | 8.9% |
| 64-bit adder | 6 | 4729 | 5683 | 1.20 | 4728 | 6116 | 1.29 | 7.06% |
| 64-bit adder | 18 | 4140 | 8568 | 2.07 | 3880 | 9444 | 2.43 | 3.19% |
| Primary1 | 4 | 22228 | 6054 | 0.29 | 22270 | 6580 | 0.30 | 8.17 % |
| Primary1 | 14 | 10390 | 10821 | 1.04 | 10060 | 10600 | 0.99 | 5.43% |

Table 2: A comparison of predictions of chip width and height length ($\mu$), and aspect ratio versus the actual data for random placement layouts.

| Example | # rows | predicted | | | actual | | |
|---|---|---|---|---|---|---|---|
| | | M1 length | M2 length | FT count | M1 length | M2 length | FT count |
| 8-bit adder | 4 | 35914 | 36466 | 44 | 32964 | 41609 | 39 |
| 8-bit adder | 6 | 36355 | 44502 | 109 | 32248 | 49513 | 113 |
| 16-bit counter | 6 | 91007 | 82873 | 184 | 82638 | 80718 | 175 |
| 16-bit counter | 8 | 94038 | 95672 | 298 | 88810 | 104058 | 321 |
| RSD | 4 | 263324 | 195069 | 103 | 250632 | 222469 | 127 |
| RSD | 8 | 229644 | 216586 | 419 | 214656 | 232082 | 441 |
| 64-bit adder | 6 | 2106909 | 1990335 | 868 | 1963184 | 2257585 | 950 |
| 64-bit adder | 18 | 2149961 | 2538525 | 4615 | 1932062 | 3027954 | 4652 |
| Primary1 | 4 | 11623500 | 3445901 | 439 | 10764122 | 3829880 | 447 |
| Primary1 | 14 | 8607196 | 3799353 | 4017 | 7804133 | 4053470 | 4113 |

Table 3: A comparison of predictions of total *metal*1 and *metal*2 lengths ($\mu$) and total number of feedthroughs versus the actual data for random placement layouts.

| # pins | predicted | | actual | |
|---|---|---|---|---|
| | M1 length | FT count | M1 length | FT count |
| 2 | 19.0 | 3.71 | 19.49 | 3.60 |
| 3 | 35.96 | 5.17 | 35.05 | 5.10 |
| 4 | 52.99 | 5.76 | 46.76 | 5.12 |
| 5 | 69.21 | 5.93 | 67.34 | 6.19 |
| 7 | 100.11 | 5.75 | 96.61 | 6.08 |
| 12 | 168.29 | 4.46 | 142.614 | 3.78 |
| 17 | 225.58 | 3.22 | 193.32 | 3.09 |

Table 4: Detailed net *metal*1 length (in units of cell pitches) and feedthrough count comparison for random placement of Primary1 benchmark with 14 rows.

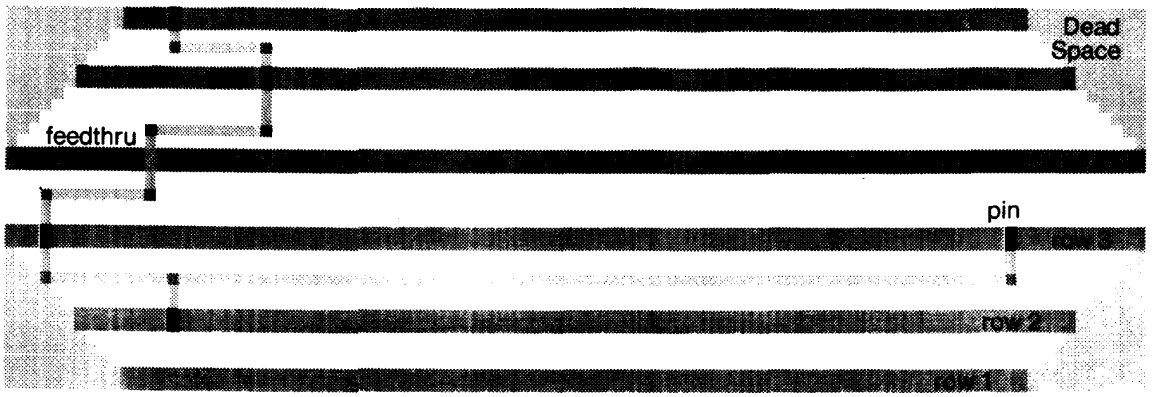


Figure 1: This terminology is used in the basic and improved interconnection models. *M1Length* of the net is sum of the *metal1* wire lengths in units of cell pitches, *M2Length* of the net is sum of *metal2* wire lengths in units of channel spans and *FTHeight* of the net is the number of feedthroughs on the net.

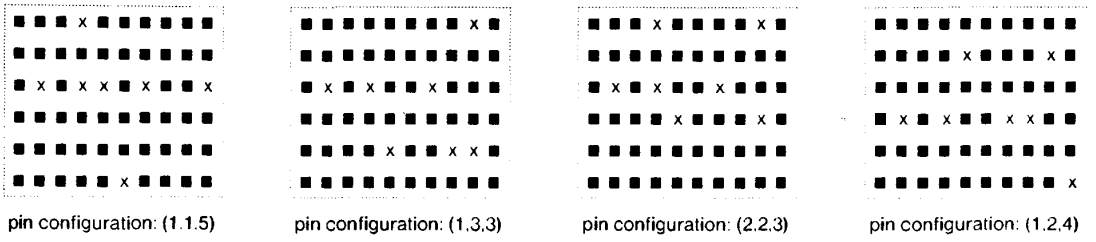


Figure 2: The allowable pin configurations for a seven pin net lying on three rows are shown here. The crosses represent pins of the net.

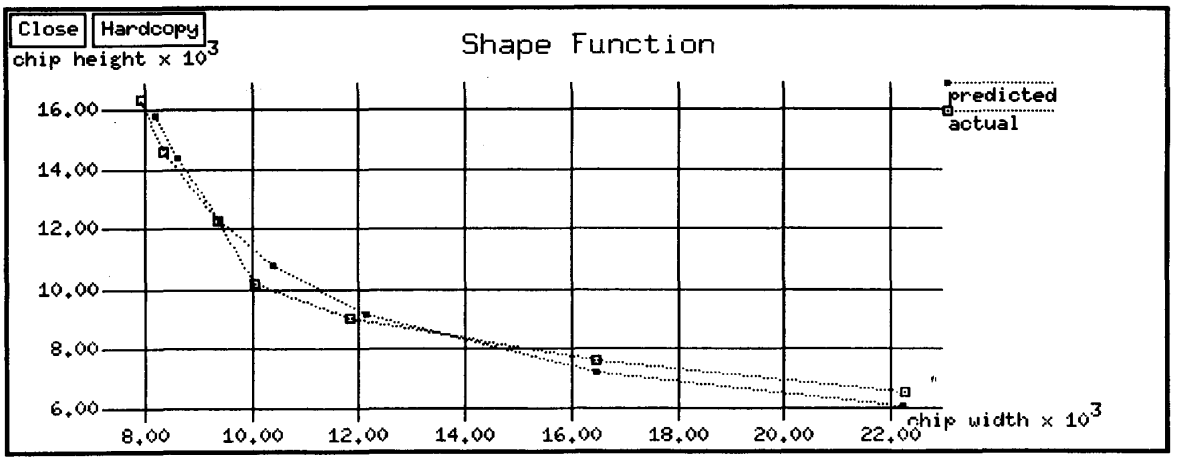


Figure 3: The predicted and actual shape constraint functions for Primary1 are shown. The procedural model produces corresponding results for other designs.