Estimating and Optimizing Routing Utilization in DSM Design

Philip Chong and Robert K. Brayton
EECS Department, UC Berkeley
(pchong|brayton)@eecs.berkeley.edu

Abstract

A method for estimating and optimizing routing utilization for digital integrated circuits is described. A wirelength distribution model based on Rent's Rule is used to estimate wiring requirements in this paper, although our method is independent of the model used. This method accounts for interconnect performance requirements, and wire sizes on the various metal layers are optimized as part of the computation. Practical application to areas outside of routing are discussed.

1 Introduction

Integrated circuit designs continue to increase in both size and complexity. With the coming of deep-submicron (DSM) feature sizes, we expect to see circuits with several tens of millions of gates within the next decade [1]. This unprecedented scale of integration will certainly bring along changes in the needs of IC designers and corresponding changes in the CAD tools which they use. To anticipate these needs, we must predict the style of such large designs and how these designs can be physically accommodated on a die.

Routing will become an increasingly important resource on a chip; a die has a finite area, and thus can accommodate only a limited number of wires. Bohr [2] predicts that interconnect will become a significant limiting factor for future designs. As evidence, he notes that as design sizes increase, the number of metal layers typically increases as well. This indicates that the wiring demands for designs are increasing faster than the design size itself, where additional metal layers are used to handle wires which cannot be accommodated on the lower layers. Since the number of metal layers must be kept low (e.g. for cost reasons), this means there will be increasing difficulty accommodating the routing for larger designs on a die.

Performance requirements on designs are becoming more stringent as well, thus putting further demands on the wiring. Long wires must be treated specially, since they suffer the most delay due to interconnect resistance and capacitance. These wires are typically routed on the upper layers of metal, where larger wires (i.e. wider and taller "fat wires") minimize the additional RC delay. However, the very nature of these wires limits their number; the wider a wire, the more area it takes up, and so large wires are a scarce on-chip resource.

The increasing importance and scarcity of interconnect in IC designs motivates us to examine how the routing resources on a chip are typically used. In this work, we describe a method to estimate the utilization of the interconnect on a die. First, we take a stochastic wirelength distribution model based on Rent's Rule [3] to estimate the wiring requirements imposed on a die. We show how to estimate the wiring capacity for each metal layer on the die, and then use *layer assignment* [4] to distribute the wires in the design among the metal layers, while simultaneously optimizing the wire sizes for each layer of metal. This layer assignment considers the performance requirements for the interconnect as well. In this manner, we obtain an estimate for the utilization of the routing resources on a chip as well as an optimized size for the wires on each layer of metal.

In addition to improving understanding of how wires are used in designs, this estimation technique can be used in more practical settings. In Section 5, we discuss how to apply this procedure in designing a fabrication process, and examine how layout/placement algorithms may benefit from this estimation technique.

2 Definitions

2.1 Some Notation

For consistency, the same terminology is used here as presented in [3]. An *interconnect* is a connection from the output of one gate to the input of another. A *net* is the collection of interconnects driven by a single gate.

Typically, metal layers are used in pairs; one layer primarily routes signals in the horizontal direction, the other in the vertical direction. Thus it makes sense to treat the metal layers in each pair uniformly. For this computation, we group a pair of metal layers into a *tier*; from here on, we use the term *tier* to mean a pair of related metal layers.

We use λ in the usual sense; that is, all design rule lengths are expressed as multiples of λ.

Let N be the number of gates in a design, let A_g be the average area of a gate, let p_g be the average pitch (average of height and width) of a gate, let F_o be the average fanout of a gate, and let M be the number of tiers in the design. We assume each of these quantities are known constants for a given design. Note that p_g is only indirectly used in this study; occasionally, wire lengths are converted between λ units and gate pitches.

Each tier on the chip has a number of parameters associated with it; we treat these as unknowns to be computed using the procedure described in Section 3. Let w_m be the width of wires on tier m, let I_m be the total number of interconnects that are routed on tier m, let ℓ_m be the length of the longest interconnect appearing on tier m, and let K_m be the total length of the available wiring (i.e. the wiring capacity) on tier m. This last parameter is described in detail in Section 3.1.

2.2 A Stochastic Wirelength Model

Davis et al. [3] give a stochastic model for the wirelength distribution in an IC design, based on the Rent parameter for the design. In particular, they give a formula for $i(\ell)$, the *interconnect density function* such that the number of interconnects with length between $\ell = a$ and $\ell = b$ is

$$I(a < \ell < b) = \int_a^b i(\ell) d\ell$$

The function

$$I(\ell) = \int_1^\ell i(\zeta) d\zeta$$

is the total number of interconnects with length ℓ or less. Similarly, the function

$$L(\ell) = \int_1^\ell \zeta i(\zeta) d\zeta$$

is the total length of all interconnects each with length ℓ or less.

In this paper we use the function $i(\ell)$ from [3] as the basis of our wiring requirements.

3 Method

The *layer assignment problem* [4] is the assignment of interconnects to metal layers (or equivalently, tiers) on a chip. This problem is important because long wires require larger (wider and taller) metal traces (found on upper metal layers) to avoid incurring excessive delay due to the resistance and capacitance of the interconnect. However, there can only be a limited number of wide interconnects due to area considerations, so the wide wires become a limited resource.

For simplicity, we impose the restriction that an interconnect resides on exactly one tier, except for the associated vias on lower tiers. Shorter interconnects are assigned to lower tiers, longer to higher tiers; this packs as many interconnects onto the lower tiers as possible, minimizing the number of vias required to pass signals up to higher tiers.

We solve the layer assignment problem bottom-up; we first estimate the total length of wiring that can fit on the bottom tier, and then we find the set of shortest wires of length $\leq \ell_1$ (where ℓ_1 is to be determined) whose total length does not exceed the capacity of the lowest tier. This set is assigned to the lowest tier, and we repeat for the remaining wires on successive tiers until the topmost tier is reached.

Two additional concerns complicate this otherwise simple formulation. First, some wires may be too long to assign to certain tiers; if the wires on a tier are not large enough, long interconnects may have excessive RC delays, violating the performance requirements for the design. We show how to deal with delay constraints in Section 3.3.

Second, the wire sizes w_m for each of the tiers will impact the estimates; if a tier has large wires, it will naturally have a smaller capacity in terms of the total length of interconnects which may be routed on that tier. Traditionally, the minimum wire size for each layer is dictated by the fabrication process used. If we do have predetermined wire sizes, then we have no additional degrees of freedom in our problem and hence no additional complications. On the other hand, we may want to optimize the minimum wire sizes for each metal layer to best accommodate a specific design or family of designs; this is particularly useful when designing the fabrication process itself. We thus have the problem of optimizing the wire sizes per tier while simultaneously packing wires onto the die area. For now, assume the wire sizes w_m are given constants; we show in Section 3.4 how to extend our technique to optimize the wire sizes as well.

3.1 Available Routing Capacity

If gates are densely packed on the die, then the raw die area required by a design is simply NA_g. If gates are not densely packed, we increase A_g (and p_g) to include the inter-gate spacings. The total area available for routing on tier m is thus $2NA_gF$, where $0 < F < 1$ is a factor to account for area which cannot be routed (e.g. due to congestion). The factor of 2 arises because the tier is composed of two metal layers.

This raw routing area will not be completely available for routing wires. We must also account for the area consumed by vias which either pass signals through tier m or connect signals to tier m. If an interconnect is routed on a tier above m, then there must be two sets of vias passing through tier m, one carrying the signal from the substrate/transistor layer up to the higher tier, and one carrying the signal back down. The upgoing signal requires two vias on tier m, one per physical layer in the tier, and likewise for the downgoing signal. Thus an interconnect routed on a tier above m will contribute 4 vias to tier m itself. For simplicity, we assume the I_m interconnects which are routed on tier m itself require 4 vias per interconnect as well. We thus have $4(NF_o - \sum_{i=1}^{m-1} I_i)$ vias on tier m, since there are NF_o interconnects in total, and the interconnects which do not contribute vias to tier m are the ones assigned to tiers below m. If each via takes A_v^m area, then the available routing area accounting for vias is

$$2NA_gF - 4A_v^m(NF_o - \sum_{i=1}^{m-1} I_i)$$

Typically, the allowed spacing between wires is the same as the width of the wires themselves [5]. Thus, for densely-packed wires, the total wiring capacity (i.e. total allowed length of wires) on tier m is

$$\begin{aligned} K_m &= \frac{2NA_gF - 4A_v^m(NF_o - \sum_{i=1}^{m-1} I_i)}{2w_m} \\ &= \frac{NA_gF - 2A_v^m(NF_o - \sum_{i=1}^{m-1} I_i)}{w_m} \end{aligned}$$

For sparsely-packed wires, we can use the factor F introduced above to account for the area wasted due to sparsity.

For our study, we chose $A_v^m = (w_m + 3\lambda)^2$. This is similar to the design rules for the MOSIS SCMOS technology [5], where the additional 3λ accounts for metal overhanging the via and spacing requirements.

Note that if layer assignment has been performed for tiers 1 through $m-1$, as in our bottom-up fashion, the values of I_i for $1 \leq i < m$ are known, and hence K_m is a known quantity for a fixed w_m.

3.2 Utilizing Available Capacity

Since K_m is a limit on the total wirelength on tier m, we have

$$\chi L(\ell_m) - \chi L(\ell_{m-1}) \leq K_m$$

where $\chi = 4/(F_o + 3)$ is a factor which accounts for sharing of wires between interconnects on the same net [3].

Our procedure performs layer assignment bottom-up, hence for a fixed w_m, both $L(\ell_{m-1})$ and K_m are known quantities. Thus the value $(K_m/\chi) + L(\ell_{m-1})$ is known, and this is an upper bound for $L(\ell_m)$. Since $L(\ell)$ is monotonically increasing in ℓ, we can easily find the largest value for ℓ_m which satisfies this bound. This value of ℓ_m thus defines the set of wires which can be assigned to tier m without overflowing the tier capacity.

3.3 Accounting for Delay Constraints

We must also account for the delay constraints imposed on the design; long interconnects cannot be assigned to layers with narrow wires. In order to determine quantitatively the interconnects which can be assigned to a layer, we use the concept of *critical length*, introduced in [4].

Consider buffered wires on each metal layer; the buffering is done optimally to minimize delay. As per [4], the delay per segment (between buffers) along this line is constant (independent of the layer), hence a delay constraint on a wire translates directly to an upper bound on the number of segments for that wire. However, the length of each optimally buffered segment is inversely proportional to $\sqrt{rc}$, where r and c are the distributed resistance and capacitance of the line. Thus the translation from interconnect delay constraints to wirelength constraints depends on the physical characteristics (i.e. the width and height) of the wires.

Let τ_{crit} be the *critical delay* for all wires in the design; that is, we allow no interconnect to have a propagation delay greater than τ_{crit}. Generally, τ_{crit} depends on the target clock period for the design, the depth of logic (i.e. number of logic stages) between latches, and the total gate delay along the critical path for the circuit. That is,

$$\tau_{crit} = \frac{\text{clock period} - \text{total gate delay}}{\text{logic depth}}$$

where the total gate delay and logic depth are measured along the critical path of the circuit.

The use of this τ_{crit} abstraction is not completely realistic; generally the distribution of slack among the wires will not be uniform. However, without more detailed knowledge of the circuit we cannot provide a more accurate model for the allocation of delays of the wires in the circuit. For the purposes of estimation, we can give τ_{crit} a value higher than the actual critical delay of the circuit. This simulates the allocation of slack, since some wires will have additional delays above the actual critical delay.

Let $\ell_{crit,min}$ (the *minimum critical length*) be the length of a minimum-sized wire which still meets the delay constraints. Note that this wire may consist of several optimally buffered segments. (This may introduce additional vias on the lower layers, but this is not accounted for in this paper. Additionally, the buffers occupy area not accounted for.) That is,

$$\ell_{crit,min} = \frac{\text{optimum segment length}}{\text{segment delay}} \times \tau_{crit}$$

Note that the optimum segment length and segment delay both depend on the distributed rc characteristics of the wire. Thus $\ell_{crit,min}$ neatly represents, in a single parameter, both the target clock frequency for the design (as part of τ_{crit}), as well as the physical characteristics of the process used.

To see how the critical length scales with the wire size, we consider a first-order model for the wires, as presented in [2]; more complex models, such as presented in [6], will give different results, but do not change our general methodology. We assume the horizontal spacing s_m between wires is proportional to their width w_m, the vertical spacing v_m (i.e. the oxide thickness between metal layers) is proportional to the height h_m of the wires, and the aspect ratio (h_m/w_m) is some fixed constant ([2] suggests an optimal aspect ratio of 2, though the exact ratio is not relevant for this study). The distributed resistance r is thus inversely proportional to $h_m w_m \propto w_m^2$. However, the distributed capacitance between a wire and its neighbour on the same layer is proportional to h_m/s_m, which is constant. Likewise, the distributed capacitance between a wire and a wire above/below is proportional to w_m/v_m, which is constant as well.

Thus the rc product of the wires is inversely proportional to w_m^2, and since the length of each buffered segment is inversely proportional to $\sqrt{rc}$, the upper bound on wire length is proportional to w_m. Under the first-order model described above, for tier m the longest wire length ℓ_m must therefore satisfy

$$\ell_m \leq \ell_{crit,min} \times \frac{w_m}{w_{min}} \tag{1}$$

We chose $w_{min} = 2\lambda$ for the purposes of this study.

There are now two upper bounds on the length of the wires assigned to each tier. The first, from Section 3.2, bounds ℓ_m based on the capacity of the tier, while the second, given in (1) above, bounds ℓ_m based on the delay of wires on the tier. Choosing the minimum of the two bounds gives a maximal set of wires which may be validly assigned to tier m.

3.4 Wire Width Optimization

So far we have assumed w_m to be predetermined constants, not variables to be optimized. We wish to treat these parameters as variables, especially if we are designing the fabrication process to best accommodate our target designs.

If w_m is restricted to multiples of λ, then we can easily try values of $w_m = 2\lambda, 3\lambda, \ldots$ to find the value of w_m that maximizes the number of interconnects for each tier in turn. Since we proceed in the layer assignment bottom-up, the number of vias extending to the higher layers is minimized, and thus the available routing capacity on the higher layers is maximized. Note K_m is monotonically decreasing in w_m, hence we can terminate the search for w_m when we see that increasing w_m will give no improvement in ℓ_m.

4 Results

To demonstrate how this procedure can be used, we chose a set of default parameters to reflect a projected typical DSM design. We show how the wiring utilization varies as we change these parameters from the defaults. We chose $p = 0.6$ and $F_o = 3$ as typical for a random-logic style of design. $\lambda = 0.05$, $N = 10M$, and $M = 5$ are projections for a design in a hypothetical $0.1\mu m$ process. $p_g = 40\lambda$, $A_g = 1600\lambda^2$ and $F = 0.65$ are chosen as an estimation of the size of a typical gate and the density of wiring in a design; the actual area of a 3-input NAND gate (a typical gate for this design) is certainly smaller than $1600\lambda^2$, but we account for some sparsity of the arrangement of gates on the die. Finally, we chose $\ell_{crit,min} = 14000\lambda$. This is in agreement with the study presented in [4], with an allowance of $80ps$ for τ_{crit}, the interconnect delay. As described in Section 3.3, this value depends on the target clock frequency for the design.

We implemented the algorithm described above as a Java applet. This allows us to interactively adjust the various design parameters while observing the effects on the overall routability.

Table 1 shows the effect of varying the design size. *Longest Wire* indicates the length of the longest interconnect in the design, in gate pitches. *Metal Widths* gives

Design Gates	Longest Wire	Metal Widths	Longest Metal	Unrouted
4M	3457	2/3/8/20/23	3457	0
5M	3883	2/3/7/16/26	3883	0
6M	4270	2/3/7/14/28	4270	0
7M	4626	2/3/7/13/31	4626	0
8M	4958	2/3/7/13/33	4958	0
9M	5271	2/3/7/12/26	4404	34
10M *	5567	2/3/7/12/22	3850	256
11M	5850	2/3/7/12/20	3448	862
12M	6120	2/3/7/12/19	3150	1995
13M	6379	2/3/7/11/18	3109	2736
14M	6628	2/3/7/11/18	3017	3886
15M	6870	2/3/7/11/17	2975	4957
16M	7103	2/3/7/11/17	2887	6546

Table 1: Default Parameters with Variation of Design Size

Rent Param.	Longest Wire	Metal Widths	Longest Metal	Unrouted
0.55	5425	2/5/12/37	5425	0
0.56	5455	2/4/10/25/37	5455	0
0.57	5485	2/4/9/19/37	5485	0
0.58	5513	2/4/8/14/37	5513	0
0.59	5541	2/3/7/13/31	5425	2
0.60 *	5567	2/3/7/12/22	3850	256
0.61	5593	2/3/6/11/17	2843	2180
0.62	5618	2/3/5/9/13	2275	6533
0.63	5642	2/2/5/8/12	1995	11529
0.64	5665	2/2/5/7/11	1775	18324
0.65	5687	2/2/4/7/10	1586	27673

Table 2: Defaults with Variation of Rent Parameter

Critical Length	Longest Wire	Metal Widths	Longest Metal	Unrouted
8k	5567	2/4/8/12/17	1675	12738
9k	5567	2/4/7/12/17	1907	8910
10k	5567	2/3/7/11/17	2098	6625
11k	5567	2/3/7/11/17	2338	4520
12k	5567	2/3/7/12/18	2700	2453
13k	5567	2/3/7/12/20	3137	1092
14k *	5567	2/3/7/12/22	3850	256
15k	5567	2/3/7/12/26	4875	18
16k	5567	2/3/7/12/31	5567	0
17k	5567	2/3/7/12/30	5567	0
18k	5567	2/3/6/12/29	5567	0
19k	5567	2/3/6/12/27	5567	0
20k	5567	2/3/6/12/26	5567	0

Table 3: Defaults with Variation of Critical Length

the optimal metal layer sizings for each tier found by our procedure, in λ units. *Longest Metal* gives the length of the longest interconnect physically achievable on the topmost metal layer on the die, in gate pitches. *Unrouted* shows the number of interconnects which could not be accommodated; a number greater than 0 indicates the procedure gives a layer assignment which exhausts all wiring capacity on the die. (There may be unused wiring area on lower layers for such designs, due to the performance requirements forcing interconnects to higher layers. However, since our procedure packs as many wires as possible onto each layer bottom-up, there is no way to avoid such waste.) An asterisk (*) denotes the default parameters used without variation.

An interesting note is that as the designs get larger, the wiring demands grows at a greater rate than the wiring capacity; as the size of design size grows (in number of gates), we start to see more unroutable wires. This agrees with the predicted trends.

Table 2 shows the effect of varying the Rent Parameter, all other parameters being held at their default values. As expected, we see a high sensitivity of the wire utilization to the Rent Parameter. With small increases in the Rent Parameter, the number of unrouted interconnects increases rapidly. With decreases in the Rent Parameter, we may turn an unroutable design into a routable one. Note that reducing p to 0.55 allows all interconnects to be assigned to the first four metal tiers; the topmost metal tier is completely unoccupied.

Table 3 shows the effect of varying the minimum critical length $\ell_{crit,min}$. Increasing $\ell_{crit,min}$ allows more interconnect to be packed on the lower tiers which would otherwise be underutilized.

5 Discussion

There are a few considerations which are not directly covered in the description of our method above. We deal with these considerations here.

The use of λ-based design rules comes into play only during the search process for w_m; recall that values of $w_m = 2\lambda, 3\lambda, \ldots$ are tried here. If the design rules used are not λ-based, then any suitably small increment can be used instead to determine the optimal w_m.

The estimate for the number of vias per tier per interconnect is fairly simplistic; recall we assume each interconnect contributes four vias on each affected tier. This does not account for effects such as vias shared between interconnects on the same net, nor does it include vias required for buffering along the interconnect. Long interconnects need to be buffered to avoid excessive RC delay, and these buffers require vias to carry the signal down to the transistor layer and back up to the upper metal layers. One could consider adding an extra number of vias based on the length of the interconnects involved. However, we note that the area utilized by the vias is only a small portion of the total routing area, hence for estimation purposes our simple via model should be sufficient.

A technique generally practiced in IC design is *wire sizing*, i.e. creating wires wider than the minimum width allowed for the associated metal layer. This enables longer wires to be placed on metal layers lower than normally allowed. As well, *tapering* wires has a similar effect. In contrast, our procedure assumes all wires on a metal layer are sized uniformly at w_m. We can emulate wire sizing or tapering by using higher values for $\ell_{crit,min}$, and reducing the value of F. Intuitively the higher value of $\ell_{crit,min}$ allows long wires which would normally reside on higher layers to be assigned to lower layers. The decrease in F takes care of the increase in utilized area due to the increased wire widths.

Since we assume a fixed aspect ratio (height/width) for all wires, our algorithm not only optimizes the widths of the wires on each layer, but also the height of these wires. This can be useful when designing a fabrication process, when the height of each metal layer is yet to be determined. We can use a stochastic wirelength model to find a set of wire heights that best accommodates a set of target designs, for example.

Note that our method is not directly tied to one model of wirelength distribution. This in itself is advantageous, as more accurate models may be substituted as needed. Additionally, this flexibility suggests a means of using our estimation technique to aid in the placement process. In particular, given a physical placement of gates, we can derive the wirelength distribution based on the connectivity of those gates. Our method can then be used to generate a measure of the routability for that particular layout, and this can be used as a metric for layout optimization. Our algorithm is fast, and thus suitable for the inner loop of a placement algorithm.

6 Conclusions

We presented a method of estimating the wiring capacity on a die and the utilization of this wiring, accounting for via areas and design performance requirements. Here we focused on a wirelength distribution model based on Rent's Rule to drive our algorithm, though our method is flexible and is not restricted to any particular model. As part of the procedure, our algorithm also optimizes the wire sizes for each metal layer. We hope to use this method to gain insight into the wiring requirements for IC designs in the deep-submicron era. Additionally, our technique seems promising for use in areas outside of routing itself, such as IC process design and placement.

Acknowledgements

Thanks to Kurt Keutzer for motivating this study. This work was supported in part by the California MICRO program, NSERC Canada, and the MARCO/DARPA Design Focus Center at Berkeley.

References

[1] Semiconductor Industry Association. National technology roadmap for semiconductors: Technology needs, 1997.

[2] Mark T. Bohr. Interconnect scaling — the real limiter to high performance ULSI. In *International Electron Devices Meeting*, pages 241–244, 1995.

[3] Jeffrey A. Davis, Vivek K. De, and James D. Meindl. A stochastic wire-length distribution for gigascale integration (GSI)—Part I: Derivation and validation. *IEEE Transactions on Electron Devices*, 45(3):580–589, March 1998.

[4] Ralph H.J.M. Otten and Robert K. Brayton. Planning for performance. In *35th ACM/IEEE Design Automation Conference*, pages 122–127, 1998.

[5] The MOSIS Service (USC/ISI). MOSIS Scalable CMOS (SCMOS) design rules, revision 7.2.

[6] Jue-Hsien Chern, Jean Huang, Lawrence Arledge, Ping-Chung Li, and Ping Yang. Multilevel metal capacitance models for CAD design synthesis systems. *IEEE Electron Device Letters*, 13(1):32–34, January 1992.