Web-Based Tools for System-Level Interconnect Prediction

Thomas Grund
Department of Mathematics
Chemnitz University of Technology
09107 Chemnitz, Germany
Thomas.Grund@mathematik.ti-chemnitz.de

Phillip Christie and Mark D. Butala
Electrical & Computer Engineering Department
University of Delaware
Newark DE 19716, U.S.A.
{Christie,Butala}@ece.udel.edu

Abstract

More accurate floor-planning and performance projections require a better understanding of the statistical relationships linking the logical description provided by the net-list with the physical layout of the optimized circuit. This can only be achieved by the analysis of a large number of diverse net-lists optimized under a wide variety of conditions. Since the analysis is performed on the local client machine web-based analysis tools offer a mechanism for overcoming the commercial intellectual property concerns and distribution issues that have limited previous investigations. This paper describes the use of three web-based java applets for conducting experiments in system-level interconnect prediction. The first is a general purpose applet for assessing the effects of different optimization strategies on interconnect statistics. The second applet implements renormalization group algorithms for estimating interconnect parameters of large net-lists by sampling the properties of small, 4×4 *wiring cells. The third applet examines the sensitivity of the cycle time to changes in statistical interconnect parameters.*

1. Introduction

It is reasonable to assume that there can be no single value of the Rent exponent appropriate for all optimization strategies. Similarly, the same optimization strategy applied to two different net-lists with different statistical wiring "signatures" (e.g., differences in the distribution of 2-pin and higher order n-pin nets or in the average numbers of nets per gate) will also result in different Rent exponents.

These issues can, in principle, be investigated by applying a variety of optimization and statistical assessment algorithms to benchmark net-lists. This approach, however, suffers from several practical difficulties: widely available algorithms employ different methods for representing net-list information leading to ambiguities in the determination of Rent exponents and other statistical parameters; the portability of algorithms between platforms severely limits comparative analysis; access to net-lists representing state-of-the-art designs is restricted due to intellectual property considerations.

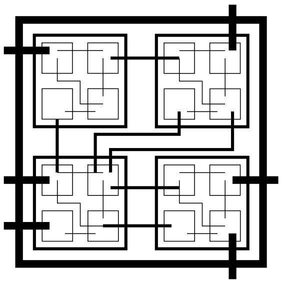


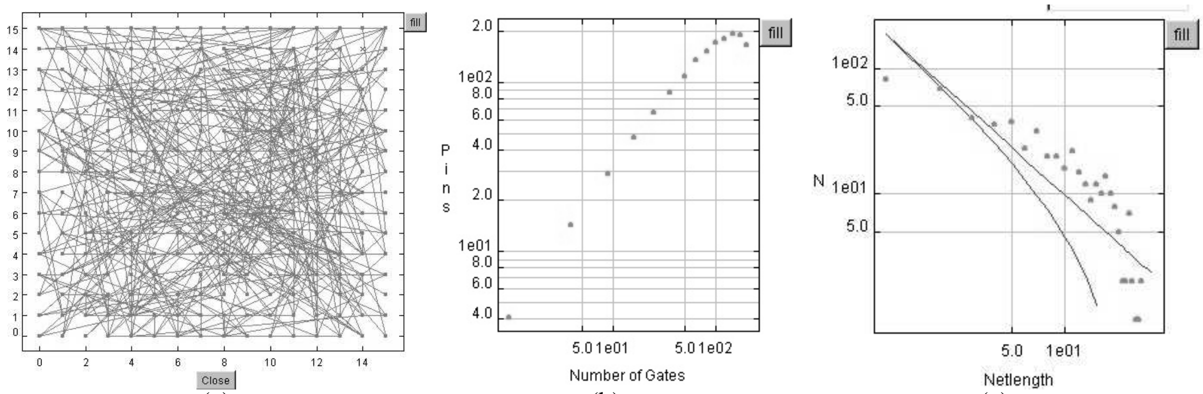
Figure 1. Hierarchical assignment of nets to a small section of a quadratically partitioned gate array. The thickness of the line indicates the assigned level.

To address these issues we have developed a series of System-Level Interconnect Prediction (SLIP) tools implemented using java applets which can be accessed using the World Wide Web. Due to space limitations it is not possible to explore all the features of the applets and only the basic functionality will be described. The initial release of the tools (designated 1.0) and their associated documentation can be found at `www.ece.udel.edu/irl/papers/slip99`.

2. System-Level Interconnect Prediction Applets

2.1. Placement Optimization Applet

The placement optimization applet consists of three main parts: net-list selection, optimization, and statistical analy-



(a) (b) (c)

Figure 2. Sample output from placement optimization applet showing (a) net-list connections optimized by minimum boundary crossing algorithm, (b) Rent analysis, and (c) length distribution analysis. The solid lines in the length analysis plot represent predictions made by two mathematical models: the lower line represents the Davis function and the upper line represents the Donath function (see the section on the clock-rate applet).

sis. Net-list selection can be performed by specifying the URL of a net-list (in YAL format in version 1.0), by creating an artificial randomly allocated net-list , or by creating a two-dimensional mesh net-list (used for benchmarking the placement algorithms).

Three placement optimization procedures have been implemented:

Minimum Wire Length Based on the classic simulated annealing algorithm [10] where the cost of each net is proportional to its length (as determined by half the perimeter of its bounding box). The algorithm attempts to find the global minimum wire length placement by adopting a (user defined) annealing schedule.

Minimum Boundary Crossing A variant of the simulated annealing algorithm which allocates a cost to each net by first imposing a recursive quadratic partition on the circuit, the cost associated with each net being determined by the highest level partition that the net crosses, as indicated by the thickness of the lines in Fig. 1.

Shifted Node Algorithm for Placement (SNAP) An iterative improvement algorithm which attempts placement based on the concept of positioning a gate at the center of mass of the gates connected to it. The algorithm is based on a two-step procedure and is presented here for the first time. It begins my shifting the position of the first gate in a Manhattan array to a location which minimizes the sum of the square of the lengths of all the gates connected to it. The next gate in the array is then shifted to its center of mass and so on, until all the gates in the array have been shifted. This process results in a highly distorted non-Manhattan arrangement of gates which is then snapped back to a Manhattan grid in the second phase of the algorithm. This process of sequentially placing gates at their center of mass and then snapping to a Manhattan grid is then repeated for some predetermined number of cycles. Typically 50-100 cycles are sufficient. This algorithm performs reasonably well when compared to other widely used algorithms but its main advantage is that runs in $\mathrm{O}(n \log n)$ time resulting in short optimization times and quick net-list selection, optimization, and analysis cycles.

It should be noted that the object oriented nature of the java programming language make the applets very extensible. It is anticipated that a wide range of algorithms for placement and routing will be added in he near future.

Fig. 2 shows three images captured from the placement optimization applet indicating the various modes for displaying optimization information. In this particular example a 16×16 array of gates with 512 randomly allocated 2-pin nets was optimized using the minimum boundary crossing algorithm. The measured Rent exponent was $p = 0.77$ and the average wire length was 7.37 gate pitches. The optimization time was approximately 5 seconds using the Netscape 4.5 virtual machine running on a 200 MHz Pentium MMX processor (this time is, of course, very sensitive to the parameters chosen for the annealing schedule).

Fig. 3 shows a comparative analysis of Rent exponents extracted from net-lists optimized by the three different algorithms available in the placement optimization applet. Artificial net-lists were used in order to vary the average number of nets per gate, with values ranging from $c = 2$ to $c = 8$.

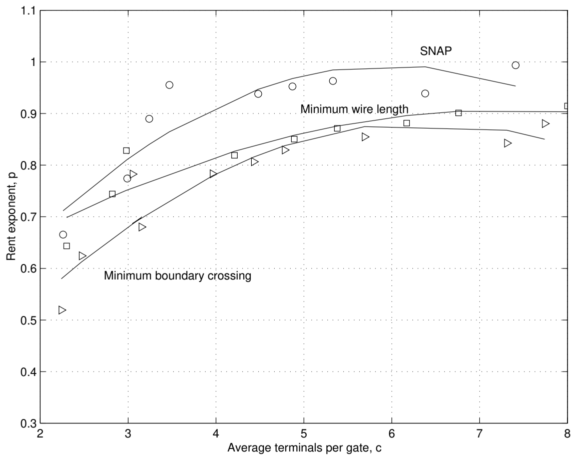


Figure 3. Rent exponent as a function of average number of nets per gate for three different placement algorithms: squares— minimum wire length, triangles— minimum boundary crossing, circles—SNAP. The solid lines represent second order polynomial functions fitted to the data to minimize the square of the error.

2.2. Rent exponent estimator applet

Finding globally optimum solutions to the placement problem is known to be NP-hard. It is therefore inevitable that the optimization of large net-lists is time-consuming; the interchange of a pair of gates to reduce the length of one of the associated nets might well increase the length of several other nets. Moreover, the length of these nets may span length scales ranging from nearest-neighbor to critical paths spanning the length of the chip.

The fact that all length scales must be considered when attempting to estimate interconnect parameters does not imply that estimation algorithms must be applied to an entire circuit in order to estimate the Rent exponent. The intrinsic self-similarity of circuits characterized by Rent's rule enables global system properties to be estimated from the analysis of small representative sub-sections of the system. This assumption of intrinsic self-similarity [3] is based on the observation that the optimal arrangement of nets at the lowest design level, e.g., the arrangement of nets between transistors to form a logic gate, will be similar to the optimal arrangement of nets at the next design level, e.g., the arrangement of gates in an adder circuit, and so on, up through the design hierarchy. This principle of "dilational invariance" enables global system properties to be inferred from small numbers of optimized gates if the wiring statistics are suitably "renormalized" to take into account the change in length scale. The Rent exponent estimator applet implements these dilational invariance assumptions in order to estimate global system properties from the properties of a small 4×4 wiring cell. This enables extremely rapid analysis of a wide range of sample net-lists.

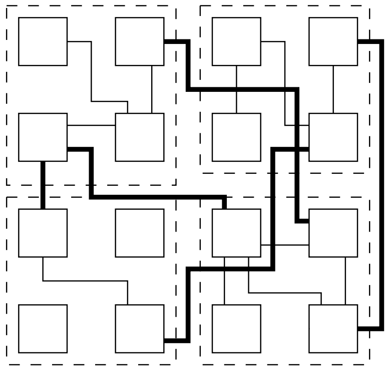


Figure 4. Hierarchical assignment of nets to 4×4 wiring cell.

The algorithm is mathematically defined within the partitioning structure originally proposed by Donath [9]. A partitioning tree is represented by idealized quadratic partition sequence of a circuit comprised of 4^H gates, generating $h = 1, \ldots, H$ hierarchical wiring levels, see Fig. 1. Nets at the lowest level ($h = 1$) are routed across partitions containing individual gates with probability q_1 and are represented by the thinnest lines in Fig. 1. Nets at level $h = 2$ are routed across partitions bounding 4 gates with probability q_2 and are represented by lines of intermediate thickness in Fig. 1. The thickest lines in Fig. 1 represent nets assigned to the $h = 3$ level with probability q_3. As the partition hierarchy is ascended gates are connected across partitions with probability q_h, the number of gates in each partition quadruples, and the expectation length of a net crossing the partition doubles.

The sampling procedure is illustrated by considering the arrangement of wires produced by an optimization procedure applied to a small 4×4 wiring cell. For the example of Fig. 4 the cell was optimized using the minimum boundary crossing algorithm described in the previous section which favored nets between small groups of gates over nets between large groups of gates and was designed to minimize the system Rent exponent. The trial net-list was constructed by randomly allocating 16 2-pin nets. The optimization procedure resulted in a total of 11 level 1 nets (thin lines) and 5 level 2 nets (thick lines). Since the maximum number of level 1 nets is 24 and the maximum number of level 2 nets is 96 (for a total of 120 nets in the fully connected cell), q_1 and q_2 are estimated to be 0.458 and 0.052, respectively.

Using the theory developed by Donath [7] for the quadratic partitioning structure it is straightforward to show that the ratio of connection probabilities is simply related to the Rent exponent by

$$p = 1 + \log_4 4\frac{q_2}{q_1}, \qquad (1)$$

and that the Rent exponent for the data of Fig. 4 is therefore estimated to be $p = 0.431$.

This value of p represents a single sample of the relationship between q_2 and q_1 that characterizes both the optimization algorithm and the wiring signature of the net-list. The structure of the entire relationship between q_1 and q_2 is deduced by repeating the two-step process of optimization and level assignment for a large number of artificially generated cells, ranging from cells with a single allocated net to fully connected cells containing 120 nets. This range of net allocations ensures that the full range of connection probabilities is sampled. Fig. 5 plots the relationship between q_1 and q_2 obtained by sampling the wire arrangements obtained by optimizing 4×4 wiring cells with the minimum boundary crossing algorithm.

Within the framework of renormalization group theory [2, 6] the relationship between q_1 and q_2 is interpreted as a Monte Carlo renormalization group transformation

$$q_2 = \mathcal{R}[q_1] \qquad (2)$$

and it provides the basic mechanism for estimating global statistical properties. With the assumption of the self-similarity of systems characterized by Rent's rule, this transformation can be applied recursively to all wiring levels and Eq. (2) is more generally written as

$$q_{h+1} = \mathcal{R}[q_h] \qquad (3)$$

so that

$$q_h = \mathcal{R}^h[q_1] \qquad (4)$$

As q_1 increases from zero, we note that q_2 will also start to increase, but at a slightly lower rate due to the small gradient of the tangent to $\mathcal{R}$ near $q_1 = 0$. By applying $\mathcal{R}$ to q_2 we note that q_3 will increase even more slowly, and q_4 at a still lower rate. Using this recursive process we conclude that for small values of q_1 the probability of long range communication using high level nets is quite low and that this corresponds to a small Rent exponent. Conversely, for larger values of q_1 we will eventually reach a value where the number of nets in each level increases rapidly and long range communication will be favored over short range communication implying a large Rent exponent. The value of q_1 which initiates this avalanche process is the critical point of the system and corresponds to the critical Rent exponent of $p_{crit} = 1/2$. From Eq. (1) this occurs when

$$\frac{q_2}{q_1} = \frac{1}{8} \qquad (5)$$

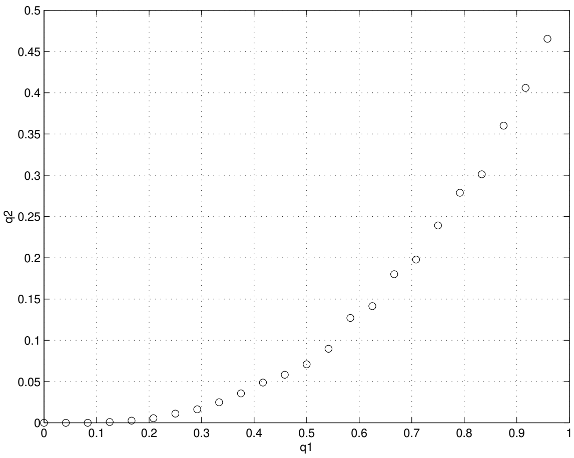


Figure 5. Sampled relationship between q_1 and q_2 for minimum boundary crossing algorithm.

and for the renormalization group transform defined by Fig. (5) this equation is satisfied for $q_{1_{crit}} = 0.4429$.

The estimation of the Rent exponent for the small wiring cells as a function of optimization strategy is interesting but not especially useful since we are primarily concerned with the Rent exponents of much larger systems. However, the self-similarity of optimized circuits allows us to extrapolate the results from the wiring cells to circuits of arbitrary size and also to infer the relationship between the Rent exponent as a function of net-list parameters which are known prior to layout. In particular we can determine the relationship between the Rent exponent and the average number of nets per gate, c.

Consider the expression for the expectation number of level 1 nets per gate, n_1 (i.e., nearest-neighbor connections). This is given by the product of the level 1 connection probability with the number of available wire sites at the first level. For the quadratic partition model each gate at level 1 can connect to a maximum of three other gates so that $n_1 = 3q_1$. Again, due to the quadratic structure of the partition, we also have $n_2 = 4n_1$ and, in general, we may write $n_h = 4^h n_1$. The the expectation number of nets per gate at all levels, which is defined as the connectivity, c, is then given by the geometric series

$$\begin{aligned} c &= \sum_{h=1}^{H} n_h \\ &= n_1 \frac{(1-\xi^H)}{1-\xi} \qquad (6) \end{aligned}$$

where $\xi = n_2/n_1 = 4\mathcal{R}[q_1]/q_1$ and H is the total number

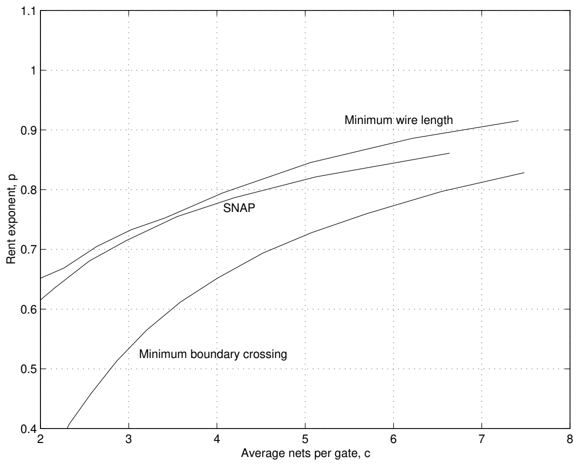


Figure 6. Rent exponent as a function of average number of nets per gate for three optimization strategies. The data were collected from 4×4 cells and extrapolated to the limiting case of infinite system size using appropriate renormalization. These results should be compared with the data shown in Fig. 3

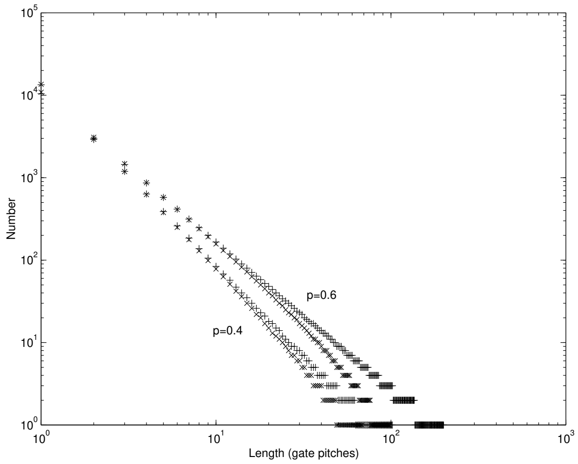


Figure 7. Length distribution of wires defined by the models of Donath (+) and Davis et al. ($\times$). For all distributions: number of gates = 10^6, fan-out = 1, connectivity = 4

of wiring levels and is related to the total number of gates by $G = 4^H$.

We observe that the expression for the Rent exponent, Eq. (1), and the expression for the connectivity, Eq. (6), are completely determined if q_1 is known. Thus, as q_1 is varied from zero to one, it will simultaneously evaluate both the Rent exponent and the net-list connectivity. Fig. 6 plots the Rent exponent as a parametric function of connectivity, generated as q_1 is varied between zero and one, for the three optimization strategies analyzed using Monte Carlo renormalization group algorithms.

2.3. Clock rate applet

The clock rate applet estimates the maximum clock rate (minimum cycle time) of an idealized chip as a function of the geometrical dimensions and material composition of the interconnect, which consists of six wiring layers (arranged in three pairs), and the statistical distribution of wire lengths. The clock rate is estimated from [1, 12]

$$\text{Clock rate} \approx [\mathrm{m}ax(DT_G, T_C)]^{-1}$$

where D is the logic depth, T_G is the delay along a wire of average length connecting two gates and T_C is the delay along the longest wire on the chip, the critical path delay.

The average and longest delays are determined from a mathematical model of the distribution of wire lengths, which is determined, in part, by the Rent exponent. Two length distribution models are defined within the applet: the first is based on the model proposed by Donath in 1981 [8] and second is based on the model proposed by Davis et al. in 1998 [4]. These distributions are shown in Fig. 7 for two different Rent exponents.

In order to determine the relevant delays, wires have to be allocated to the wiring layers so that the appropriate geometrical and materials parameters of the interconnect and devices can be used to calculate the Elmore delay [11]. Starting with the lowest wiring layers, wires are allocated in order of increasing length. When the capacity of each layer is reached wires are added to the next highest layer until all the wires are allocated [5]. Once the wires are allocated the Elmore delay associated with the wire of average wire length is calculated using the geometrical and material parameters of the lowest pair of layers. The critical path delay is determined by calculating the delay of the longest wire in each layer, based on the geometrical and material parameters of the layers (since the critical path delay is dominated by the interconnect delay device parameters are not used). The critical path delay is then defined as the largest of these delays. Fig. 8 shows the output of the applet as the Rent exponent is changed from $p = 0.50$ to $p = 0.55$.

The estimated clock rates indicated in Fig. 8 are very sensitive to the choice of Rent exponent and this underscores the importance of accurate Rent exponent prediction methods. However, the statistical model of the chip is closely coupled to the electromagnetic properties of the interconnect through the geometry and material composition of the wires and dielectrics in each layer. The clock rate data were calculated with all wire geometries set to a

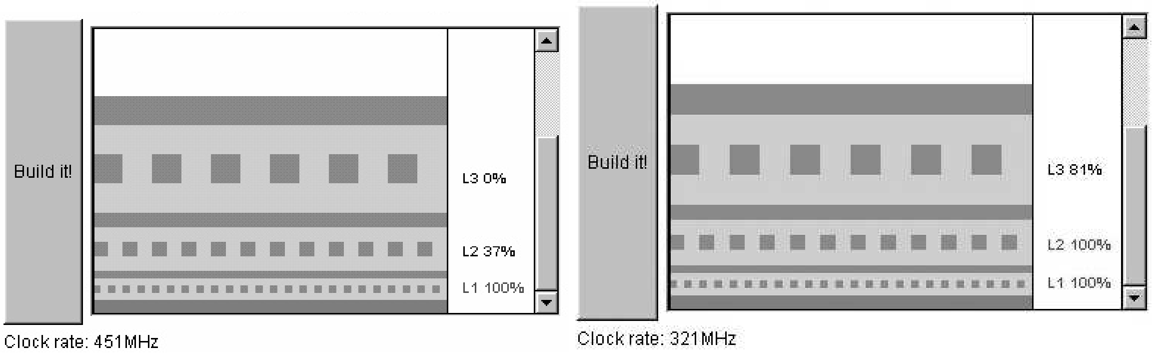


Figure 8. Sample geometrical display from clock rate applet.

simple “fat” wiring scheme in which all dimensions were doubled in each wiring layer.

These settings, however, do not represent the optimum geometrical arrangements in each layer. Extensive experimentation with the applet confirms that the optimal “fat” wiring scheme requires modification of the simplistic scaling indicated in Fig. 8. Furthermore, the optimal geometries appear to be dependent on the choice of Rent exponent. The investigation of the coupling between the statistical and electromagnetic models of the interconnect is an vitally important issue and requires a more methodical investigation of the design space. The applet is therefore equipped with an simulated annealing algorithm which searches the design space for the geometries which result in a globally optimum clock rate as a function of the statistical interconnect parameters. The results of these investigations will be reported at a later date.

3. Conclusions

In order to improve our understanding of the complex interactions which determine interconnect performance it is necessary to study a wide variety of optimization strategies and net-lists. The applets previewed in this paper represent initial attempts at web-based design and many further modifications and improvements are already in the pipeline.

The object oriented nature and inheritance mechanisms of the java programming language makes the applets high extensible. Considerable effort has gone into making the class interfaces and GUI very simple and programmer-friendly. The incorporation of new optimization algorithms and length distribution models by other interested researchers can therefore be managed in a straightforward manner. It is hoped that the evolution of this project will help both new students and practicing design engineers visualize the complex issues governing the optimal allocation of interconnect resources.

References

[1] H. B. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, 1990.

[2] J. J. Binney, N. J. Dowrick, A. J. Fisher, and M. E. J. Newman. *The Theory of Critical Phenomena*. Oxford University Press, 1992.

[3] P. Christie. A fractal analysis of inerconnection complexity. *Proceedings of the IEEE*, 81(10):1492–1499, Oct. 1993.

[4] J. A. Davis, V. De, and J. Meindl. A stochastic wire-length distribution for gigascale integration (gsi)—part I: Derivation and validation. *IEEE Trans. Electron Devices*, 45(3):580–589, March 1998.

[5] J. A. Davis, V. K. De, and J. D. Meindl. A stochastic wire length distribution for gigascale integration (gsi)—part II: Applications to clock frequency, power dissipation, and chip size estimation. *IEEE Trans. Electron Devices*, 45(3):590–597, March 1998.

[6] C. Domb. *The Critical Point*. Taylor & Francis, 1996.

[7] W. Donath. Placements and average interconnection lengths of computer logic. *IEEE Trans. Circuits and Systems*, CAS-26(4):272–277, 1979.

[8] W. Donath. Wire length distribution for placements of computer logic. *IBM J. Res. Develop.*, 25(3):152–155, May 1981.

[9] W. E. Donath. On the equivalence of memory to random logic. *IBM J. Res. Dev*, 18:401–407, 1974.

[10] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.

[11] L. T. Pillage, R. A. Rohrer, and C. Visweswariah. *Electronic Circuit and System Simulation Methods*. McGraw-Hill, 1995.

[12] G. A. Sai-Halasz. Performance trends in high-end processors. *Proc. IEEE*, 83:18, 1995.