An Information Theoretic Rent's Rule

Hema Narasimhadevara

B.E, Electronics and Communications, Osmania University (2001)

A thesis presented to the faculty of the

OGI School of Science & Engineering

at Oregon Health & Science University

in partial fulfillment of the

Requirements for the degree

Master of Science

In

Electrical Engineering

Under the guidance of

Dr. Dan Hammerstrom

April

2004

The dissertation "An information theoretic Rent's rule" by Hema Narasimhadevara has been examined and approved by the following examination committee.

> Dr. Daniel Hammerstrom Professor Thesis Research Adyisor

Dr. Todd Leen Professor

John D. Lynch Instructor

Acknowledgements

I would like to express my deepest gratitude to my thesis advisor Dr. Dan Hammerstrom, Professor, Department of Computer Science and engineering without whose patient guidance this work would not have been completed.

I would like to thank John Lynch, Instructor, Department of Computer Science and Engineering, Dr. Todd Leen, Professor, Dept of Computer Science and Engineering and Roy Kravitz, Instructor, Department of Computer Science and Engineering for their support and encouragement.

My special thanks to my friend Sunil and my family who always had trust in me.

Table of Contents

ABSTRACT	4
An Information Theoretic Rent's Rule	5
CHAPTER 1	7
INTRODUCTION 1.1 Interconnect 1.2 Contribution Of This Thesis 1.3 Thesis Outline	7 7 13 14
CHAPTER 2	15
DEFINITIONS 2.1 Definitions 2.2 Interconnect and Information Theory 2.3 Information Theoretic Rent's Rule	15 15 19 26
CHAPTER 3	28
DESCRIPTION OF THE EXPERIMENTAL SYSTEM 3.1 Experimental System 3.2 Rent's Rule and A Priori Interconnect Estimation 3.3 Assumptions 3.4 Static Measurements 3.5 Dynamic Measurements	28 28 32 32 33
CHAPTER 4	36
 RESULTS AND CONCLUSIONS 4.1 Results From Calculating The Rent's Parameters (Static) 4.2 Results From Calculating The Rent's Parameters (Dynamic) 4.3 Implications 4.4 Virtual Wires 4.5 Costs-Performance Evaluation and Future Work 4.6 Summary 	36 37 38 39 39 40 42
REFERENCES	48

ĺ

List of Figures

Figure 1.1 A directed Graph	.16
Figure 4.2 4 - Quadrant representation of a circuit graph structures	. 19
Figure 4.1 Static and Dynamic Rent's Parameters	.43
Figure 4.2 Static Rent's Parameters on a log- log Scale	.44
Figure 4.3 Dynamic Rent's Parameters on a log-log plot	.45
Figure 4.4 Static and Dynamic Rent's Parameters on a log- log Scale	.46

List of Tables

Table 1.1 MOSFET and Interconnect Latency for $1.0-\mu m$, 100-nm, and 35-nm Processe	es 10
Table 1.2 The ITRS projections for switching delay, switching energy ,clock frequency total chip current drain, maximum number of wiring levels, maximum total wire length per chip, and chip pad count for 1.0-um,100-nm, and 35-nm- technology generations	r, 12
Table 4.1 A table comparing Rent's parameters for various systems obtained from various Rent's curves of standard designs	37
Table 4.2 Routing for a 50nm microprocessor	41

ABSTRACT

An Information Theoretic Rent's Rule¹ Hema Narasimhadevara OGI School of Science & Engineering, 2003 Supervising Professor : Dan Hammerstrom

In complex VLSI with circuits approaching one billion transistors, the major limiting factor to performance is interconnect. With continued scaling of semiconductors this problem is becoming ever more serious. Potential new technologies such as molecular scale implementation have very limited interconnect capacity making it a bottleneck to performance.

The purpose of interconnect is communication, more specifically communication between distant points with a small latency. This thesis looks at the interconnect requirements of a circuit from an information theory perspective, considering the information rate of each connection in the circuit as opposed to the physical connectivity.

We have built a system that evaluates the physical connectivity of circuits by using a traditional static version of Rent's Rule. We also define a new, dynamic Rent's Rule. The physical connectivity of the circuit is initially calculated as the number of

¹ This research was supported in part by the Semiconductor Research Corporation, Cross-disciplinary Semiconductor Research Program (CSR), "Computing with Nano-scale Devices – Looking at Alternative Models," December 2001.

logic primitives and their interconnections as parameters, then these data are compared to the connectivity requirements of the circuit obtained taking the entropy of the dynamic behavior of each connection as a parameter. This paper also discuses the possibility of using virtual wires with multiplexers to emulate the physical connections in a system for more effective implementations.

A primary motivation for this work is to better understand the nature of interconnect and then to apply what we have learned to nano- and molecular-scale devices where it appears that connectivity will be far more limited than with traditional semiconductors.

CHAPTER 1

Introduction

1.1 Interconnect

Information technology is driving the silicon industry, which, thanks to Moore's law has grown exponentially in the past forty years. Several strategies, such the scaling of the minimum feature size, increasing die size and enhancing packaging efficiency, have contributed to these advances.

Scaling has two aspects: The scaling of devices and the scaling of the wires. There is concern about our ability to scale both these components and at the same time continue to improve performance. Device scaling is getting more difficult. Issues such as the gate oxide tunnel currents and punch through have become major limitations. Small devices such as the Fin Field Effect Transistor (FinFETS¹) are being built to overcome size limitations and the problems associated with extremely small devices. The speed of operation increases linearly with the reduction of device size. The power supply needs to be scaled for power consumption of the entire circuit, but this is difficult because the threshold voltage of the device is not being reduced in comparison with device scaling. However, with lower power levels, driving devices and wires becomes more difficult and expected speed gains may not be realized [Hor03].

Wire scaling has several issues associated with it. For fixed-length wire scaling, the resistance of the wire increases with scaling, and as the wires get closer together, the capacitance also increases. The increase in resistance alone increases the wire delay relative to the gate delay by a factor of 2 per scaling generation. For wire scaling, the

¹ The FinFET relies upon a thin vertical silicon "fin" to help control leakage of current through the transistor when it is supposed to be off.

resistance is constant but the capacitance decreases linearly with scaling. The picture is different when the wire length decreases, the scaled wire delays stay more or less constant relative to gate delay.

With this kind of reverse scaling, the performance of the wires can avoid being a problem if the width of the wires increases relative to the device features and retains the same basic design during process scaling. However, the complexity of the design is never the same for each generation. The ratio of the wire delay to gate delay will continue to increase slowly. Currently they both scale by roughly the same scaling factor but the wires are becoming slower relative to the gates with each generation. The key is to keep the logical span or fan-out of the wire constant. The problems start when the chip design becomes more complex, which is inevitable. In this case the wires need to connect to more gates and the logical span increases making them slower. Using repeaters and pipelining helps to some degree, but it is not enough. The main interconnect problems arise because of this increase in complexity. When, for each generation, we compare the chip size to that of the wire and its logic span, the chip looks small to a wire in the old view but large to a wire in the new view [Hor03].

Increased performance has been an important part of Moore's law. Processor performance has been increasing rapidly with increased clock rates. In addition to faster clocks that are enabled by smaller transistors, computer performance improvements are also due to more complex micro architectures that are enabled by the availability of more transistors, moving to pipelined superscalar architectures that leverage multiple port memories, multiple function units and out of order execution. With each design improvement there has been an increase in logic complexity. The direct consequence of this increase in complexity is an increase in the interconnectivity costs.

Another result of Moore's law is that clock frequency doubles with each generation. There are two factors which contribute to this, the technology (1.4X/gen) and the circuit design itself. The clock speed has been scaling faster than the base technology (transistor speed). Again architecture plays a roll, utilizing more pipelining, where the number of gate delays in a clock cycle is decreasing. In addition, faster circuits are due to better design techniques and CAD tools. All these techniques require more transistors, increasing the complexity of the circuit and hence higher wire costs.

The main problem with more complex designs rests with the wires and not the devices. Until a decade ago, the delay caused by interconnects in microprocessors was less than the switching time of transistors, so interconnect delay was not a big factor in microprocessor design. The scaling of interconnect reduces silicon area, but increases latency (response time) in absolute value and energy dissipation relative to that of the transistors [JM95]. As wires become thinner and move closer together, capacitive coupling increases. Another factor is increasing current density as interconnect frequency increases.

To quantify the exploding disparity between the latency of interconnects and switching speed of transistors, consider the comparisons illustrated in Table1.1 [MDZ01]. For the 1- μ m-generation technology of the late 1980s, the "*CV/I*," or intrinsic switching delay of a MOSFET [BAA96] (when not loaded with parasitic or wiring capacitance), is approximately 20 ps.

9

Technology Generation	MOSFET Switching delay (td=CV/I) (ps)	RC response time (Lint= 1mm) (ps)	Time of flight (Lint = 1mm) (ps)
$1.0\mu m (Al, SiO_2)$	~20	~1	~6.6
100nm (Cu, k(dielectric constant)=2.0)	~5	~30	~4.6
35nm (Cu, k=2.0)	~2.5	~250	~4.6

Table 0.1.1 MOSFET and Interconnect Latency for 1.0-µm, 100-nm, and 35-nm

Processes

However, for the same generation, the total resistance-capacitance product or RC delay of a "benchmark" 1.0-mm-long interconnect is about 1.0 ps. In comparison, for the 90-nm generation that is currently being deployed, the CV/I delay of a MOSFET decreases to 5 ps, while the RC latency of a 1.0-mm-long wire increases to 30 ps. The key observation is that as semiconductor technology is advancing from the 1.0- μ m to the 90-nm generation, the RC delay or response time of a benchmark 1.0-mm-long interconnect is devolving from 20 times faster to six times slower than transistor switching delay. Furthermore, the 1999 International Technology Roadmap for Semiconductors (ITRS) projection for 35-nm technology in 2014 suggests a 2.5-ps transistor delay and a 250-ps RC latency for a 1.0-mm-long interconnect. For completeness, the time of flight (ToF) of a 1.0-mm-long interconnect is included in Table1.1. As indicated, Time of flight delay is independent of scaling but does depend on the value of the relative permittivity of the interconnect dielectric.

The numerical values for RC delay in Table1.1 represent simple best-case calculations. For example, the results do not account for the adverse results of surface

scattering, high-frequency skin effect, liner thickness for copper interconnects, or temperature increases in a multilevel wiring network.

Not only is latency an issue with interconnects, but they also present an energydissipation problem as illustrated in Table 1.2. Energy dissipation will increasingly limit the performance of designs as a consequence of practical constraints on the heat-removal capabilities of chip packaging. The energy dissipation associated with a binary transition of a minimum-geometry MOSFET versus a 1.0-mm-long interconnect is respectively 1/3, 5, and 30 times larger for interconnect of the $1.0-\mu m$, 100-nm, and 35-nm-technology generations. These imbalances clearly indicate that interconnect contributes significantly to the power dissipation of chips.

Historical records and ITRS projections of clock frequencies for highperformance microprocessors are summarized in Table 1.2. These data assume 30 MHz, 3.0 GHz, and 13 GHz as the respective nominal clock frequencies for the $1.0-\mu m$, 100nm, and 35-nm-technology generations. These rapidly escalating requirements place enormous new demands on the interconnect that implements clock distribution networks of large chips. Power dissipation, and variation in the time of arrival of a clock pulse (skew) at different points on a chip, and fluctuation in clock edge arrival time (jitter) represent increasingly complex design problems.

Although the signal and clock distribution problems are daunting in large designs, power distribution is also an important issue. As noted in Table 1.2, estimated maximum chip current drain is respectively 2.5 A, 150 A, and 360 A for the $1.0-\mu m$, 100-nm, and 35-nm-technology generations. Concurrently, power-supply voltage scales down from 5.0 V to 1.0 V to 0.5 V for the corresponding generations. These aggressive expectations for high-current, low-voltage power distribution impose utterly unprecedented demands on interconnect networks.

	1.0 um	100 nm	35nm
MOSFET Switching delay(ps)	~20	~5	~2.5
Interconnect RC response	~1	~30	~250
time(ps)(Lint=1mm)			
MOSFET switching energy(fJ)	~300	~2	~0.1
Interconnect Switching energy(fJ)	~400	~10	~3
Clock Frequency	~30MHz	~2-3.5GHz	~3.6-13.5GHz
Supply Current(A)	~2.5	~150	~360
$(V_{dd} = 5.0, 1.0, 0.5 \text{V})$			
Maximum number of wiring levels	3	8-9	10
Maximum total wire length per chip	~100	~5000	-
Chip pad count	~200	~3000-4000	4000-4400

Table 1.0.2 The ITRS projections for switching delay, switching energy ,clock frequency, total chip current drain, maximum number of wiring levels, maximum total wire length per chip, and chip pad count for 1.0-um,100-nm, and 35-nm- technology generations.

In order to meet these demands, semiconductor manufacturers have responded by switching from aluminum to copper, a superior conductor of electricity. They also have resorted to employing ever-increasing numbers of wiring layers, with some of the latest microprocessors employing up to 9 metal layers. Manufacturers have also looked closely at improving low-k dielectric materials between the wiring levels. Porous dielectric materials, which could help the situation, are fragile and contaminate easily, and electromigration is an issue. Adding more layers of wiring to microprocessor processes would increase costs substantially, since each new layer adds more complexity to wafer processing and requires more masks, which are also becoming more expensive. Interconnect problems are not limited to CMOS, but appear as if they will also have wider implications for technologies at the molecular scale. Since these technologies appear to be even more limited in providing interconnect between the computing devices.

As a result, understanding connectivity, modeling it effectively, and finding alternatives to traditional connection architectures are becoming ever more important.

1.2 Contribution Of This Thesis

Interconnect allows communication between different points in a circuit. Obviously we would like to do this with minimal area, delay and power. Traditionally *a priori* wire length estimations and connectivity requirements have been based on Rent's Rule, a heuristic that relies on the number of physical connections between entities. However, not all connections are used at all times. Consequently, in this thesis we address the issue of whether it is possible and useful to consider other measures of communication at the chip level. The approach chosen here is to measure the dynamic information content on each wire, which can be evaluated from the bit rates on the wires. This information can then be used to compute an information theoretic version of Rent's Rule, which takes into account the activity and utilization of the wires as opposed to just their physical presence. The resulting rule determines the connectivity requirements based on the dynamic information rate of each wire.

For this research we developed a system that calculates, for a given circuit under standard operating conditions, the entropy for each wire and uses these data to obtain the connectivity requirements by defining a new set of Rent's parameters. The key result of this work is that the data indicate that the information content of most wires is very low most of the time.

13

In the last part of this thesis we look at ways to leverage these results by examining the tradeoffs involved with a virtual wiring scheme, where several *virtual* wires are multiplexed over a smaller number of *physical* wires.

1.3 Thesis Outline

Chapter 2 defines the problem, reviews Rent's Rule and presents dynamic and information theoretic versions of the rule. Chapter 3 describes the system we built for calculating the entropy and the information theoretic Rent's Rule parameters from simulated circuits. And finally, Chapter 4 discusses the results obtained, the implications, the tradeoffs involved in using virtual wires, and ends with the conclusions and suggestions for future research.

CHAPTER 2

Definitions

2.1 Definitions

This chapter defines the various terms and relationships used in the remainder of this dissertation. In order to calculate the data flow in a wire the circuit has to first be converted into an abstract data structure that allows us to capture the necessary data. The best candidate is a graph. In this chapter, we discuss this graph structure, the modeling of a wire as a communication channel, the calculation of the entropy of a wire, and the definition of the information theoretic Rent's Rule.

This work focuses on a key problem of a large circuit, its intercommunication requirements. Initially, the computation actually performed by each unit is ignored for the Rent's Rule value calculations. In a later step, the output of each interconnection is used in the calculation of the entropy of signal on each wire. Ignoring the actual computations being performed by the circuit allows us to concentrate on the interconnect requirements of the design. The major constraint on the interconnection architecture is the number and physical placement of the inputs to, or outputs from each node. This interconnection pattern can be represented as a directed graph with the primitives as the vertices and the wires between them as edges.

Definition 2.1: A directed graph G(N, E) is a finite set of vertices N and a set of edges E, where $E \subseteq N \times N$. Thus each element of E is an ordered pair (i, j) where $i, j \in N$. Edges are directed so that the edge (i, j) is different from the edge (j, i) and is said to originate with vertex *i* and terminate with vertex *j*. This paper will use the term "node" interchangeably with "vertex," and "connection" with "edge."

For an example see the Figure 2.1, where the set of nodes is $\{a,b,c,d\}$ and the set of connections is $\{(a,b),(a,c),(a,d),(b,d),(c,a)\}$. The arrow on each connection indicates its direction. For example, (a,d) originates with node a and terminates with node d. If connections are present in both directions between a pair of nodes, both need to be explicitly shown, as are in (a,c) and (c,a).



Figure 1. 0.1 A directed Graph

Definition 2.2: A connection matrix is a $N \times N$ boolean matrix that describes the connections of a graph. Column indices represent the terminating nodes in the related directed graph. A "0" in a given position indicates the absence of a connection, a "1" indicates its presence. Since the graphs are directed, the connection matrix is not generally symmetric.

For the static Rent's Rule calculation the connection matrices are binary matrices. It is possible to have connection matrices with values other than "0" or "1". For such non-binary matrices, the magnitude of an entry indicates the value of some associated attribute of the edge.

Definition 2.3: A <u>communication graph</u> (*c-graph*) is the connectivity graph of a computation being performed. The values associated with the nodes represent bit transformations on the wires, and connections are the required communication between transformation "units" or "primitives."

Definition 2.4: The physical graph (p-graph) is the physical realization of the computation described by the communication graph, vertices represent compute elements such as logic gates, and edges the physical connections between the hardware components.

When synthesizing a logic network to a set of standard cells, which are placed into silicon design, the c-graph and p-graph will generally be isomorphic. But this is not always so. When the two graphs differ; it is generally because there is a sharing of physical resources by virtual resources. Generally, the p-graph is smaller and sometimes it can be a single node. For example, when synthesizing to an FPGA where the physical

17

connections are configured dynamically, the c-graph and the p-graph are not necessarily the same.

In a computer cluster, the p-graph is the physical network of processors and their interconnections. A single p-graph vertex will often emulate many c-graph vertices. Likewise each p-graph edge (e.g. a bus) may contain many "virtual" c-graph edges.

The research described here concentrates primarily on the p-graph. The p-graph of a circuit can further be classified based on the type of nodes and connections.

Definition 2.4: <u>Static nodes</u> represent the logic units or circuit "primitives" such as multiplexers, adders, de-multiplexers, etc., they are a direct physical realization of the actual circuit These nodes are weighted with the silicon area required to implement them.

Definition 2.5: Dynamic nodes represent a property of the physical nodes, they are virtual nodes, which are actually the bit transformations at the output of the logic unit. For instance a particular logic gate might not be active all the time, and its dynamic node will be weighted by its percentage activity.

Definition 2.6: <u>Static connections</u> represent the physical connections between the logic units, i.e., the actual wires in the circuit. These connections are weighted by their physical interconnections which is a static measure.

Definition 2.7: <u>Dynamic connections</u> represent a property of the physical connections in a circuit. Each of the connections has the value of the bit rate of the wires constituting the connection.

Figure 2.1 shows the various formulations that are possible given the above definitions. There can be four different kinds of p-graphs for a given circuit

- 1. Completely Static: In this representation the graph of the circuit is its physical realization. The nodes and the connections represent static information. This is used in the experimental model for calculating the static Rent's parameters.
- Hybrid Static-Dynamic: Here the nodes are still static, i.e., weighted by their physical area, but the connections are dynamic, i.e. weighted by their entropy. This model is used to calculate the dynamic Rent's parameters.
- 3. Completely Dynamic: In this category, the nodes and the connections are weighted by their dynamic measurements of bit transformation rates.
- 4. Hybrid Dynamic-Static: Here the nodes have been weighted by their bit transformations but the connections represent the actual hardware.

These last two are not addressed in this paper.

STATIC	STATIC-DYNAMIC
Pure P-Graph Representation	Hybrid P-Graph Representation
(Physical Nodes, Physical Edges)	(Physical Nodes, Virtual Edges)
DYNAMIC	DYNAMIC- STATIC
Hybrid P-Graph Representation	Hybrid P-Graph Representation
(Virtual Nodes, Virtual Edges)	(Virtual Nodes, Physical Edges)

Figure: 4.0.2 A 4 - Quadrant representation of a circuit graph structures

2.2 Interconnect and Information Theory

A wire is essentially a simple communication channel that transmits information. A communication system has a number of constituent parts, but for our purposes we can use a simplified model. In a communication channel data are transformed into symbols from a fixed finite alphabet, here our data set is $\{0,1\}$. The sequence of 0's and 1's representing the data set are transmitted into the channel, here a wire, and received at the other end of the channel.

The concept of <u>entropy</u> plays a pivotal role in information theory, it is a mathematical formulation of the uncertainty in a data set that is being communicated [AA97]. It also can be thought of as the minimum number of bits required to efficiently encode the data set. To motivate the formula for Entropy, we introduce two functions hand its expectation H, h is the measure of the uncertainty about the outcome of a single experiment which is modeled by a random variable. Any viable candidate for the function should satisfy certain conditions. We want the function h to depend only on the probability of the outcome of an experiment or event. That is, $h:[0,1] \rightarrow R_+$ is a function of the probability $p \in [0,1]$ and takes values that are non-negative real numbers. If we combine two independent events occurring with probabilities p_1 and p_2 , then the uncertainty associated with the occurrence of both events is $h(p_1p_2)$. If the uncertainty of the occurrence of the first event is removed, then the result is the uncertainty of the occurrence of the second event. In symbols this is stated by $h(p_1p_2)-h(p_1)=h(p_2)$ or

$$h(p_1,p_2) - h(p_1) = h(p_2)$$
 2.2.1

It is natural to require h to be a monotonically decreasing function on the interval [0,1], since the more likely the occurrence of an event, the less uncertainty is associated with it. The following function for h has this characteristic

$$h(p) = -C \log p \qquad 2.2.2$$

There is no need to specify the base of the logarithm and in view of the positive multiplicative constant C, we can fix any base and modify the constant C accordingly.

The logarithmic form of the function h is also compatible with the manner in which data are encoded in a digital environment. For example, if we use a binary system, an integer in the range $[0,2^{n}-1]$ requires $n=log_{2}2^{n}$ bits to encode, which means by specifying ninstances of 0 or 1 we can completely remove the uncertainty about which number in the range $[0,2^{n}-1]$ is being encoded.

The entropy H is the expectation of the quantity h over all possible events. For example, assume that the random variable X takes discrete values 1, 2, 3,... with respective probabilities p_1 , p_2 , p_3 ,.... Then the function h is given by $h(p_i) = -C \log p_i$ and the expectation of h is

$$H(X) = -C\sum_{i} p_i \log p_i$$
 2.2.3

We can look at H axiomatically, much in the same way as we arrived at the logarithmic expression for h, and see what natural requirements we may impose on H. First assume that our random variable X only takes finite values 1, 2,..., N and each value occurs with probability 1/N. Then we write

$$H(X) = H(\frac{1}{N}, ..., \frac{1}{N}) = \rho(N)$$
 2.2.4

In this situation H is a function of N and the expected uncertainty in the result of an experiment with N equally likely outcomes is, or should be, a monotonically increasing function of N. Now suppose we combine two independent experiments or random variables X and Y whose outcomes are specified by the integers $\{1, 2, ..., N\}$ and $\{1, 2, ..., M\}$ respectively. Furthermore we assume all outcomes of X are equally likely, and similarly all outcomes of Y are equally likely. Then the average uncertainty in carrying out both experiments X and Y is H(XY) and if we remove from the average uncertainty associated with the experiment X we should be left with the average uncertainty of Y, i.e.,

$$H(XY) = H(\frac{1}{MN}, ..., \frac{1}{MN}) = H(X) + H(Y) = \rho(N) + \rho(N)$$
(2.2.5)

We can remove the constraint that the outcomes of X are equally likely and let X assume value j with probability p_j . We can group the outcomes of the experiment into two groups corresponding to the partition $\{1, 2, ..., N-1, N\} = A \cap B$ where $A = \{1, 2, ..., L\}$ and $B = \{L+1, L+2, ..., N\}$. Then X takes values in A (with respect to B) with probability $P_A = p_1 + ... p_L$ (resp. $P_B = p_{L+1} + ... + p_N$). X is a compound experiment where we first choose A or B with probabilities P_A and P_B . If A is chosen then the outcome will be $j \in \{1, 2, ..., L\}$ with probability $\frac{p_j}{P_A}$, and similarly for B. The average uncertainty

associated with the outcome of being in A is $H(\frac{P_1}{P_A}, ..., \frac{P_L}{P_A})$. In carrying out the experiment we first choose one of the groups A and B. The associated average uncertainty with this choice is $H(P_A, P_B)$. If this quantity is removed from $H(p_1, ..., p_N)$ then we are left with the average (relative to the choice of group A or group B) of the uncertainties of

the outcome of being in A or in B. Since the average uncertainty, given that the outcome is in group A, is $H(\frac{p_1}{P_A},...,\frac{p_L}{P_A})$ and similarly for the outcome of being in B, we obtain

$$H(p_1,...,p_N) = H(P_A,P_B) + P_A H(\frac{p_1}{P_A},...,\frac{p_L}{P_A}) + P_B H(\frac{p_{L+1}}{P_B},...,\frac{p_N}{P_B})$$
(2.2.6)

The properties described by (2.2.4), (2.2.5) and (2.2.6) together with the requirement that H(p, 1-p) be a non-negative continuous function of $p \in [0,1]$, determine H up to a positive multiplicative constant C. That is, H is necessarily of the form

$$H(X) = -C\sum_{i} p_i \log p_i \tag{2.2.7}$$

The fact that an expression of this form satisfies the stated requirements is straightforward, and the proof is omitted [AA97,RS90, TJW91].

Having defined the notion of entropy, we can now define some related concepts. Let X and Y be not necessarily independent random variables, and taking values $\{1, 2, ..., N_X\}$ and $\{1, 2, ..., N_Y\}$ respectively. Then H(X, Y) is the entropy associated with carrying out both experiments X and Y. To define conditional entropy we first note

$$H(X | Y = k) = -C \sum_{j=1}^{N_X} P(X = j | Y = k) \log P(X = j | Y = k)$$

and then define the conditional entropy of X given Y as

$$H(X | Y) = -C \sum_{k=1}^{N_{Y}} \sum_{j=1}^{N_{X}} P(Y = k) P(X = j | Y = k) \log P(X = j | Y = k)$$
(2.2.8)

which is equal to

$$-C\sum_{i,k} P(X = j | Y = k) \log P(X = j | Y = k)$$
(2.2.9)

With this definition it is easy to show the following proposition:

Let X and Y be discrete random variables, then

$$H(X | Y) = H(X,Y) - H(Y)$$

The information about a random variable is a quantity which should be in some sense the inverse to the uncertainty about it. For instance, if we have two possibly correlated random variables X and Y, then the *average information conveyed by* Y *about* X is reasonably defined as

$$I(X | Y) = H(X) - H(X | Y)$$

Notice that this definition says that if we subtract from the average uncertainty about X, the average uncertainty about X given Y then we have the information about X which is conveyed by Y. We see that the quantity I(X|Y) is symmetric in X and Y and we have

$$I(X|Y) = I(Y|X)$$
 (2.2.10)

In view of symmetry it is reasonable to call I(X|Y)=I(Y|X) the mutual information of X and Y. We can now apply these to entropy. Wire frequency data gives us the average utilization or bit rate for each wire for one execution of the simulation, which can be used to compute the likelihood of a single bit being transmitted on the wire, p_i . This value is used as an estimate of the first order information transmitted, with each wire viewed as a communication channel, and it is roughly equivalent to the first order "unconditional" or H₁ entropy for the connections into and out of each block,

$$\sum_{\forall i} p_i \log p_i$$

The <u>order</u> of the entropy refers to how much history of the wire utilization is included in the computation. Buffering several time steps of wire utilization implies more information and less uncertainty. We can define higher order entropy as follows, where the x_i are values for x at different, discrete, times steps:

$$H_n(s) = \sum p(x_{n-1}x_{n-2}...x_0)\log p(x_{n-1} | x_{n-2}...x_0)$$
 and

$$H_0(s) > H_1(s) > H_2(s) > \dots$$

Information in a computer is certainly higher order, but capturing and utilizing higher order encodings can be very expensive. We have not necessarily assumed that there is no higher order information, but that we will not try to leverage that information, therefore throughout the rest of this dissertation we will use H_1 exclusively. The savings we get in coding efficiency will be dependent on how much information we are willing to capture, but this also increases the implementation complexity. In many applications, there seems to be a point of diminishing returns for n=3.

2.3 Information Theoretic Rent's Rule

E.F. Rent of IBM published two internal memoranda in 1960 that contained the log-log plots of the "number of pins" versus "number of circuits" in a logic design.[DVSJ00]. These data tended to form a straight line in a log-log plot and yield the relationship

$$N_p = K_p N_G^b$$

Where,

 N_p is the number of external signal connections ("pins") to a logic block

 N_g is the number of logic gates in the block

b is Rent's constant, and

 K_p is a proportionality constant

The values of K_p and b for the IBM computers were reported to be 2.5 and 0.6, respectively

This rule gives an *a priori* estimate of the wire lengths and the connectivity requirements of the design. We can calculate the Rent's Rule parameters by first translating the circuit into graphs which are then partitioned. Interconnect requirements are determined by equating the graph vertices inside each partition as the "number of circuits" and the edges cutting the partitions as the "number of pins" for each partition. After several iterations over various sizes of partitions Rent's parameters can be obtained for a design.

When we consider Rent's Rule as it is currently defined, it is a *static* measure based on the connection locality of the physical structures used to implement the system's function. This version is calculated using the static graphs of the circuits where each node takes as its value the area of the primitive cell from the cell library. The value of each connection between partitions is equal to the number of individual connections to each net.

One can also define a *dynamic* version of Rent's Rule that measures the number of bits communicated. We proceed as we did before, by creating a graph of the design for a certain partitioning of the circuit. In this case the graph vertices are weighted by the number of transitions at the output and the weight of each edge is the number of transitions (during the test period) on the wire the edge represents. At each clock cycle, a wire can assume 3 states {0, 1, Z} which are logic low, logic high, or high impedance. We run the system under standard operating conditions and take the state of all wires at each clock cycle. The number of state transitions in each wire with respect to the number of clock transitions gives us the bit rate or wire utilization. The weight of each node is still the physical area of the logic inside the partition represented by the node. The resulting graph then is used to compute the Rent statistics for different partition sizes as we did before. The dynamic Rent's Rule depends on both the number and length of physical interconnect and its dynamic utilization, which gives us a measure of the utilization of the wires during the circuits operation.

The bit rate of each wire obtained for the dynamic Rent's rule is synchronous with the clock. It is possible that the information the wire contains is in the timing information provided by the wire and not in the transition itself. We have not incorporated such cases into our methodology. In such cases the bit rate obtained would not necessarily correspond to the actual information being communicated on the wire.

CHAPTER 3

Description of the Experimental System

3.1 Experimental System

We developed a system that captures data from real designs and simulations in order to compute the static and dynamic Rent's Rules parameters. In addition to the standard static (physical implementation) Rent's Rule, this system, based on dynamic data from a logic simulation calculates the entropy of each wire and uses these data to obtain the connectivity requirements by defining a new set of Rent's parameters dependent on the information rate of each wire. This system, in turn, creates the necessary graph structures for computing the information theoretic Rent's Rule. We capture both the static and dynamic measures for the p-graph of a design. The representation of the system and the data flow are shown in figure 3.1.

3.2 Rent's Rule and A Priori Interconnect Estimation

Designing a computer system requires several iterations of the physical design cycle. After partitioning the circuit and creating a floor plan, the circuit components are placed. The wires between the components are then routed based on the placement. A bad placement cannot be solved by routing, therefore, the information obtained during routing generally leads to a new placement step to improve the placement results. Another routing is then performed and so on. To improve the placement and routing results, it is important to use the best possible estimates of area and wire length. Three general kinds of estimates are used: *a priori*, *on-line*, and *a posteriori* [ZDM98, V82]. The first, a priori, estimates circuit

priori, on-line, and a posteriori [ZDM98, V82]. The first, a priori, estimates circuit parameters before any of the layout steps (floor planning, placement, or routing) are performed. On-line estimates are obtained during the layout process and are based on the information that results, though sometimes incomplete, from layout itself. A posteriori estimates are obtained after a complete layout step and are, obviously, the most accurate.

In this research work we have used a priori estimates to compute Rent's rule. A priori estimates need basic information about the circuit to be designed, about the circuit/device architecture in which it is to be designed, and about the layout process that performs the implementation of the circuit. We have taken the actual design in Verilog which gives us the circuit functionality and then synthesized the design by targeting it to the architecture and a layout process, yielding detailed information about the net list (interconnections), the sizes of the elements, and the I/O. This information about the interconnect complexity is then used to estimate the Rent's rule [Pg22].

Rent's Rule has been found to apply well to several real designs [DS98, V82, LR71]. Moreover, it has been shown that random graphs do not obey Rent's Rule [DS99]. So, Rent's Rule is rather specific for "real circuits" and not for "any arbitrary circuit graph." Based on experimental analysis, there are reasons to believe that Rent's Rule says something intrinsic about the way we design logic. For example, the design process is hierarchical and the interconnection complexity must be manageable at each level of the hierarchy. Therefore, it seems reasonable to assume that the conceptual interconnection complexity remains equal at each stage, introducing a self-similarity within the circuit structure.

In most cases, Rent's Rule is a reasonably good fit for a wide range of module sizes. The validity of this rule is a result of the self- similarity that exists in real circuits. A high Rent exponent implies many global interconnections, and hence a high interconnection complexity. For large module sizes, which correspond to the highest level of hierarchy, the number of terminals quite often appears to be lower than expected from Rent's Rule. This deviation of the data from the main power law of Rent's rule is known as region II in Rent's rule. It can be intrinsic to the functionality of the circuit, but it is mostly due to the technological constraints.

Also, some circuits show a deviation from Rent's rule for low values of terminal and gate counts, which is referred to as region III. It occurs when there is a mismatch between the average number of ports and the Rent's coefficient. Rent's rule describes the first-order statistics of the terminals-count distribution for different partitions of the circuit. A simple stochastic model can be derived to describe the higher order statistics. [DVSJ00]. Circuits that follow Rent's rule are considered homogeneous. At higher levels of the hierarchy mismatches occur due to heterogeneity. This heterogeneity and the impact on the Rent's characteristics can be modeled [PV2000]. For the purpose of our research we have assumed that our modules can be modeled with the first order Rent's rule terminal-count distribution.

These exceptions (Region I and Region III) to Rent's Rule can generally be addressed in the context of systems design. For example, at the boundaries of the circuit (the inputs and the outputs), the complexity may be distorted for a couple of reasons:

1. Designers are forced to modify the number of circuit inputs and outputs because of pin limitations. Several techniques can change the pin count (e.g., serializing

30

the output lines), lowering the interconnection complexity but increasing the timing complexity.

2. The encoding of input/output information. Information theory tells us that we can often re-encode information to reduce the required bandwidth³. If we have to input a single integer value between 0 and 255 to a circuit, we will not use 256 lines with a one-hot encoding. Instead, we will use a binary encoding which requires only 8 lines to represent the integer. For both input and output the number of pins is lowered resulting in a deviation from the static version of Rent's Rule, generally known as Rent's region II [LR71].

Also, local variations in interconnection complexity do occur in real circuits [VSV95]. A processor, for example, it has cache memory on chip and an ALU. These two blocks have a totally different interconnection complexity since the cache is a rather regular structure while the ALU is not. In each of the parts, Rent's Rule will still hold, but the Rent exponents may be different.

However, even in such cases, an "overall Rent exponent" exists as an average. In [ZDM98], a heterogeneous Rent's Rule is presented as an extension to include such variations. Are results based on this heterogeneous Rent's Rule still valid? What if hierarchical design is not used, would Rent's Rule still apply? If self-similarity also results from the way we describe algorithms, it could well be that it is preserved and Rent's Rule may still hold. If true, then Rent's Rule may result not only from the design process itself but also from the algorithm description. For the data collected here, we assume that for intermediate sized blocks we can state that inside large blocks (e.g., after

³ Note, this does not change the information content, but just encodes information more efficiently.

partitioning and floor planning), an average Rent's Rule is a result of the self-similarity and is a meaningful measure for the results presented below.

I.

3.3 Assumptions

There are a number of assumptions that were made concerning the collection of the data. During the measurement of static data the primitives in the design are taken from the logic units of the EDIF file. The system assumes the weight of each of these logic units is equivalent to three fundamental primitives or gates.

For dynamic data measurement the circuit is operated under standard conditions for a fixed amount of time. Data are collected that are used as estimates of the wire utilization, from these data we can calculate the first order entropy.

3.4 Static Measurements

To capture the static parameters, we begin with a Verilog design that is synthesized into EDIF (Electronic Data Interchange Format version 2.00) and mapped to AMI 1.2 μ technology using the Mentor ADK tools. This EDIF file is then converted to a c-graph, where the nodes are weighted using cell sizes from the target AMI library.

We then map the c-graph onto a p-graph with fixed size blocks. The graph is partitioned to determine the relationship between the nodes (entities in the partition) and the wires going into and out of each partition. We then iterate over a range of block sizes.

To partition the design for different block sizes we used HMETIS, a hyper-graph partitioning package [4], that divides the graph into k roughly equal parts such that the weights of vertices (area required by logic circuits) in each partition are roughly equal and that the number of hyper edges cutting each partition is minimized, where a hyper edge is an edge connects more than one vertices. The algorithms used in HMETIS do multilevel hyper-graph partitioning and are described in [GVR95, GRVS97]. This method combines the vertices and the edges to a coarse graph (coarsening phase), partitions it and then restores the graph to a partitioning of the original graph.

3.5 Dynamic Measurements

In order to obtain dynamic information for the circuit, we first consider the cgraph, storing the edge identifiers that connect the vertices together. In the next step, the synthesized EDIF netlist was converted into Verilog. This Verilog version is then simulated using a test bench in ModelSim that captures the signal values per clock for each connection identifier. These data are then used to collect data frequency for each connection. The dump file consists of data sampled at each of the wires and gates at each clock edge synchronously

This wire frequency data gives us average utilization or bit rate for each wire for that simulation run, which can in turn be used as an estimate of pulse probability, p_i , where *i* is the index of the wire. The p_i , in turn, are used to compute the first order entropy of each wire.

- 1. With this calculation each wire in the design is viewed as a communication channel between the nodes connected to that wire.
- 2. This calculation is then roughly equivalent to the first order "unconditional," or H_1 entropy for the connections into and out of each block

$$\sum_{\forall i} p_i \log p_i$$

The procedures used to estimate the Static and Dynamic Rent's rule is summarized as follows.

Static Rent's Rule Estimations:

1. The physical circuit is used to create a graph, the node values represent the size of the logic represented by the node, and the connections represent the metal interconnect these are then weighted by how many logic elements share the wire.

2. For a range of block sizes, the graph is then partitioned into roughly equal weight blocks.

3. The average block size and inter-block connectivity is collected for each partitioning, this forms one data point for the calculation of the Rent's rule parameters.

Dynamic Rent's Rule Estimations:

1. The design is simulated and values for all connections for the simulation time are collected, the simulator provides transition frequency from which the probabilities of 1s and 0s on the wire are measured, the probabilities allow the Entropy for the wire to be calculated

2. A graph is created where the node values are the physical sizes of the nodes, the connection weights are the entropy for each wire.

3. At this point partitioning of the graph and measuring the block sizes and connection counts is done exactly as with the static Rent's rule parameters. The interblock connectivity is the entropy on the inter-block connections.





Chapter 4

Results and Conclusions

L

This chapter summarizes the results presented in this dissertation and discusses the implications of these results. We also make suggestions for possible future work. In this dissertation we have defined an information theoretic version of Rent's Rule and developed a system for computing both the traditional static Rent's Rule and our information theoretic version from actual Verilog based designs. We now present and discuss these results.

Key factors that influence Rent's constants include system architecture, organization and implementation. The design philosophy and methodology also affect these constants. For example, to ensure the highest possible speed of data transfer, multiplexed I/O pins and serial transmission of data over available pins are avoided when possible, which results in higher pin counts. On the other hand, to achieve low-cost packaging in commercial microprocessors and memories, bi-directional and multiplexed I/O pins and serial data transformation are employed. And there are other artifacts, for example, two chips with exactly the same number of gates may have different pin counts due to package cost or power dissipation considerations. Or when the average interconnection calculations derived for one design methodology (for example gate arrays) are applied to a chip designed by another methodology (for example custom designed microprocessors).

36

System or Chip type	Rent's Exponent	Rent's multiplier
Static memory	0.12	6
Microprocessor	0.45	0.82
Gate array	0.50	1.9
High speed computer	0.63	1.4
chip and module level		
High speed computer	0.25	82
Board and system level		

Table 4.1 A table comparing Rent's parameters for various systems obtained fromvarious Rent's curves of standard designs [DLL99]

To minimize these effects we have used two designs that were targeted to the same technology, used the same basic implementation methodology and have similar architectures. One design is a multi-master, bi-directional serial bus interface that provides a simple and efficient method of data transfer between devices, including collision detection and arbitration to prevent data corruption when two or more masters attempt to access a bus at the same time. The other design is an implementation of the CORDIC (coordinate rotation digital computer) algorithm, which is a method for computing elementary trigonometric and exponential functions using simple hardware such as shifts, additions, subtractions and compares.

4.1 Results From Calculating The Rent's Parameters (Static)

The system that calculates the Rent's parameters starts with a Verilog HDL description of the design. The partitioning algorithm uses the silicon area of the primitives to form the weights for the logic units. And for interconnect, the number of connections to each wire is used to specify the wire's "weight."

The system used to calculate the static Rent's Rule parameters produced the following results, $K_p = 2.28$ and C = 0.51, where, K_p is the proportionality constant and b is Rent's constant.

4.2 Results From Calculating The Rent's Parameters (Dynamic)

The system designed to calculate the dynamic information theoretic Rent's parameters is dependent on the test vectors used for the simulation, and the operation of the design under the simulated conditions. The test benches for the systems under test were run under standard operating conditions, and were designed to test most of the simulation possibilities. For example, the test bench for the multi master bidirectional bus instantiates the master and slave modules, resets the system, sets the initial values of the registers, and checks the read and write aspects. There are inbuilt checks for bits to check memory access, and it aborts the test run on encountering errors. This simulation was run for 40,000ns.

After the test runs, the log file is checked for the state changes on wires. This program ignores both the unknown state 'X' and the high impedance state 'Z'. The system for the dynamic Rent's Rule parameters gave the following results, $K_p = 2.21$ and C = -0.0021.

4.3 Implications

The research reported in this dissertation shows that the interconnect bandwidth is vastly underutilized when comparing the entropy of the interconnections and their actual bit rates to the physical interconnection architectures. Figures 4.1 and 4.4 which show the plots of static and dynamic curves clearly exhibit a vast discrepancy between the Rent's Parameters obtained, substantiating the conclusion. Previous work in interconnection architectures has focused on the maximum bandwidth of interconnected subsystems, no attempt was made to obtain detailed information on the temporal utilization of a wire. However, the significant difference between the static and dynamic Rent's Rules shows that the information flow is extremely small and localized. For the two designs we studied, under typical execution patterns, the bit rate of the wires indicated that they were vastly underutilized. The obvious question concerns whether we can utilize these data to help us build more efficient computing structures. We next discuss a few options to increasing wire utilization.

4.4 Virtual Wires

It is possible to multiplex a number of connections over a single physical wire. There are a number ways to do this, but the most straightforward is Time Division Multiplexing (TDM), where an address is sent that indicates a change in the signal on the addressed "wire." TDM can be implemented in a variety of ways from simple synchronous multiplexing to asynchronous packets. Multiplexed connections can be thought of as *virtual wires*. In this situation, the c-graph is different from the p-graph, where the p-graph refers to the physical wire and the c-graph to the virtual wires multiplexed over the physical. Virtual wires can have several advantages, including the possible reduction of interconnect area and power. And in some cases performance can also be improved by allowing the use of larger shared drivers and wires. Depending on the length and utilization of a wire, the use of virtual wires for some connections would increase wire utilization, though the cost-performance trade-offs are not always obvious. The physical area of the wire and its drivers are traded-off with that of the multiplexing hardware, and the switching costs. A thorough examination of the price-performance of these trade-offs for several real designs is required, but this is beyond the scope of the work presented here.

On the utility of the wire is about 10% or lesser, so theoretically about 10 physical wires can have one virtual wire. But, in the actual implementation the complexity of sending data, its encoding and later decoding is complex and would involve the use of extra hardware.

Another problem with virtual wires is that they may add delay to wires, which would be a problem for some wires, though not for all. Designers are beginning to look at asynchronous circuits again, in such an environment the small time delays added by wire multiplexing could be less of a problem.

4.5 Costs-Performance Evaluation and Future Work

To get a sense of the trade-offs involved in using virtual wires, we have evaluated multiplexing a wire using TDM (Time Division Multiplexing) data with a 4-1 multiplexer (MUX) for the data. We assumed the AMI 1.2 μ technology with minimum sized transistors. The area for the multiplexer would be 24.75 μ^2 . Introducing multiplexers such as this for short-distance point-point or high bandwidth connections would probably

not be cost-effective, since the area saved by having fewer wires would not offset the area for the multiplexers on each end of the wire.

Using virtual wires also involves the extra hardware of select signals (transistors or a tri-state gate) for the logic units. The number, size of select signals, the buffers required will all depend on the size of the unit being selected and could increase the total area for each logic unit by about 3-5%. The Table 4.3 presents information about the local, semi-global and global interconnects. It does not appear to be cost effective for local interconnect.

But for global interconnect especially for connections greater than >10mm, we can justify trading off increased local hardware to decreased long range connectivity. If global interconnect are as underutilized as our data indicate, this tradeoff of interconnect vs. multiplex costs could be very effective. Here we are trading off inexpensive local computation for expensive non local communication.

Type of interconnect	Typical Length
Global wires (2 levels)	>16mm
"Low-fat" global wires (2 levels)	<16mm
Semi-global wires (2 levels)	~2.5mm
Local wires (3 levels)	~0.35mm

Table 4.2 routing for a 50nm microprocessor [Dns99]Other advantages of this trade off are:

<u>Energy/power</u>: As the number of repeaters increases for each generation, (from 14 to 138 for the 50nm technology [Den98]), a 50% increase in power for 3*Win of top level metal is anticipated, where Win is the width of the top layer. These

penalties are caused by scaled wires and repeaters and can be reduced by decreasing the number of global wires.

Į.

- 2. <u>Area</u>: Although the area for local computation increases, the total area is not much different from the original design. Depending on the number of wires which can be multiplexed in a given design, this area increase would be a mere 2-4% of the total area.
- 3. <u>Performance</u>: The inductance due to metal connections, crosstalk, and issues such as connectivity limitations can be addressed using this alternative. Also, with fewer wires they can be farther apart and larger, thus reducing the resistance and inter-wire capacitance.

In this chapter we have explored the possibility of utilizing the available bandwidth of the interconnections more effectively by using multiplexed virtual wires. The use of virtual wires has promise, but much more study of the various trade-offs is required.

4.6 Summary

The primary contribution of this research is the definition of an Information Theoretic version of Rent's Rule. A complex system using state-of-the-art EDA tools was built that allowed us to capture from simulations of real designs the various data needed for computing both the traditional and information theoretic rules. The final contribution is the result that in the real designs we studied, the wires are significantly underutilized. This led to speculation that the use of multiplexed virtual wires could possibly save significant area and improve performance. In addition to its use for existing technology, these results have application to nano- and molecular-scale devices where connectivity limitations are going to be much more severe than they are today. These technologies will most likely require a greater ratio of local computation to non-local communication, making the use of virtual wires even more appealing.

ł



Dynamic and Static Rent's Parameters

Figure 4.0.1 Static and Dynamic Rent's Parameters

43



Figure 4.0.2 Static Rent's Parameters on a log- log Scale



Figure 4.0.3 Dynamic Rent's Parameters on a log-log plot



Figure 4.4 Static and Dynamic Rent's Parameters on a log- log Scale

References

1 .

ik s n

[MDZ01] J.D.Meindl, J.A.Davis, P.Zarkesh-Ha, C.S.Patel, K.P.Martin, and P.A.Kohl, "Interconnect opportunities for gigascale integration," ,IBM journal of Research and Development :Scaling CMOS to the limits, Vol 46, Numbers 2/3, 2002. pg 245-264

[Hor03] Mark Horowitz, [2000]"Life After Silicon: An oxymoron?", Slow wires, hot chips and leaky transistors: New challenges in New Millenium ISCA 2000[Online] Available :www.cs.wisc.edu/~arch/www/ISCA-2000-panel/ Mark.Horowitz.isca2000.ppt [Viewed : March 16, 2004]

[DVSJ00] J. Dambre, P. Verplaetse, D. Stroobandt and J. Van Campenhout, "On Rent's rule for rectangular regions," ELIS Technical Report, GHENT University 2000. In Proc. of Intl. Workshop on System-Level Interconnect Prediction, March 2001. [online] Available: http://citeseer.ist.psu.edu/dambre01rents.html [Viewed : March 16, 2004]

[PD00] Phillip Christie, and Dirk Stroobandt, "The Interpretation and Application of Rent's Rule," IEEE transactions on VLSI systems, December 2000, pg 639-648

[DS98] D. Stroobandt.[1998] "Analytical methods for a priori wire length estimates in computer systems", November 1998. Ph.D. thesis (translated from Dutch), University of Ghent Faculty of Applied Sciences [Online] Available: http://www.elis.rug.ac.be/~dstr/dstr.html [Viewed : March 16, 2004]

[DS99] D. Stroobandt. "Rent's Rule Coincidence or the Result of the Design Process?" A talk in the System level Interconnect Prediction Conference 1999. [Online] Available: http://www.elis.ugent.be/~dstr/SLIP.html [Viewed : Dec 10, 2003]

[MM02] Mathew M Zeigler and Micea R Stan, "A case for CMOS/Nano Co-Design,", proceeding archives, ICCAD 2002 [Online]

Available :http://www.sigda.org/Archives/ProceedingArchives/Iccad/Iccad2002/05c_1.p df [Viewed : March 16, 2004]

[KB99] Kwabena A.Boahen, "Point-to-Point connectivity between Neuromorphic chips using address events," IEEE Transactions on circuits and systems, Vol XX, No V, pg 100-117

[GRVS97]George Karypis, Rajat Aggarwal, Vipin Kumar and Shashi Shekar, "Multilevel hypergraph partitioning:application in VLSI domain", Design Automation Conference, 1997, pages 526-529.

[GV95] G.Karypis and V.Kumar, "A fast and highly quality multilevel scheme for partitioning irregular graphs," SIAM journal on scientific computing, 20(1998), pg. 359-392.

[GVR95] G. Karypis, V. Kumar, R. Aggarwal, and S. Shekhar, *hMeTiS A Hypergraph Partitioning Package* Version 1.0.1. University of Minnesota, Department of Comp. Sci. and Eng., Army HPC Research Center, Minneapolis, 1998.[Online] Available : http://www.cs.umn.edu/ karypis [Viewed : March 16, 2004]

1.

a la come de la come de

[CA95] Charles J. Alpert and Andrew B. Kahng," Recent directions in netlist partitioning", Integration, the VLSI Journal, vol. 19(1-2), pp. 1--81, 1995.

[JM95] J. D. Meindl, "Low Power Microelectronics: Retrospect and Prospect," Proc. IEEE 83, April 1995. pg 619–635

[BAA96] M. Bohr, S. S. Ahmed, S. U. Ahmed, M. Bost, T. Ghani, J. Greason, R. Hainsey, C. Jan, P. Packan, S. Sivakumar, S. Thompson, J. Tsai, and S. Yang, "A High Performance 0.25 Micron Logic Technology Optimized for 1.8V Operation," IEDM Tech. Digest, pp. 847–850, December 1996

[ITRS] International Technology Roadmap for Semiconductors (ITRS), 1999 Edition, Semiconductor Industry Association, [Online] Available: http://public.itrs.net/Files/2002Update/2002Update.pdf [Viewed: March 16, 2004]

[ZDM98] P. Zarkesh-Ha, J. A. Davis, W. Loh, and J. D. Meindl, "On a pin versus gate relationship for heterogeneous systems: Heterogeneous rent's rule," In Proc. of IEEE Custom Integrated Circuit Conf., pages 93-96. IEEE, May 1998

[V82] W. V. Vilkelis, "Lead reduction among combinatorial logic circuits," IBM Journal of Research and Development, no. 26: pages 342-348, 1982

[DLL99] David.L.Landis "Rent's rule overview: Rent's Rule and Machine Organization" A tutorial on Rent's Rule,1999.[Online] Available: www.cedcc.psu.edu/ee497i/rents_rule.PDF [Viewed : March 16, 2004]

[LR71] B. S. Landman and R. L. Russo, "On a Pin Versus Block Relationship for Partitions of Logic Graphs," IEEE Trans. Comp., C-20(12):1469-1479, 1971

[VSV95] H. Van Marck, D. Stroobandt, and J. Van Campenhout, "Towards an extension of Rent's Rule for describing local variations in interconnection complexity," In S. Bai, J. Fan, and X. Li, editors, Proc. 4th Intl. Conf. for Young Computer Scientists, pages 136-141. Peking University Press, 1995.

[TJW91] Thomas M. Cover and Joy A Thomas, Wiley, 1991, Elements of Information Theory. Chapters 1-2.pg 4-112

48

[RS90] Robert M. Gray, Springer-Verlag, 1990. Entropy and information theory. Chapters 1-5 pg 3-234

1.0

alk a w

[AA97] Amir Asadi ,[1997] The Center for Mathematical Sciences (CMS) at the University of Wisconsin Madison 1997.A tutorial on information theory[Online] Available : www.cms.wisc.edu/~cvg/course/491/modules/ INFORM/Information/node2 [Viewed : March 16, 2004]

[PV2000] P.Verplaestse "A stochastic model for the interconnection topology of digital circuits", IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 9, no. 6, pp 938-942, December 2001