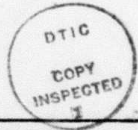


unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A188 702

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM														
1. REPORT NUMBER 87-11-10	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER														
4. TITLE (and Subtitle) Hercules: A Power Analyzer for MOS VLSI Circuits		5. TYPE OF REPORT & PERIOD COVERED Technical														
7. AUTHOR(s) Akhilesh Tyagi		6. PERFORMING ORG. REPORT NUMBER														
9. PERFORMING ORGANIZATION NAME AND ADDRESS NW Laboratory for Integrated Systems Dept. of Computer Science, FR-35 University of Washington, Seattle, WA 98195		8. CONTRACT OR GRANT NUMBER(s) MDA 903-85-K-0072 ARPA-4563, #2 code 5D30														
11. CONTROLLING OFFICE NAME AND ADDRESS DARPA - ISTO 1400 Wilson Boulevard Arlington, VA 22209		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS														
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) ONR University of Washington 315 University District Building 1107NE 45th St., JD-16, Seattle, WA 98195		12. REPORT DATE November, 1987														
		13. NUMBER OF PAGES 4														
16. DISTRIBUTION STATEMENT (of this Report) Distribution of this report is unlimited.		15. SECURITY CLASS. (of this report) unclassified														
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE														
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		<table border="1"> <tr><td>Accession For</td><td></td></tr> <tr><td>NTIS GRA&I</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>DTIC TAB</td><td><input type="checkbox"/></td></tr> <tr><td>Unannounced</td><td><input type="checkbox"/></td></tr> <tr><td>Justification</td><td></td></tr> <tr><td>By</td><td></td></tr> <tr><td>Distribution/</td><td></td></tr> </table>	Accession For		NTIS GRA&I	<input checked="" type="checkbox"/>	DTIC TAB	<input type="checkbox"/>	Unannounced	<input type="checkbox"/>	Justification		By		Distribution/	
Accession For																
NTIS GRA&I	<input checked="" type="checkbox"/>															
DTIC TAB	<input type="checkbox"/>															
Unannounced	<input type="checkbox"/>															
Justification																
By																
Distribution/																
18. SUPPLEMENTARY NOTES DISTRIBUTION STATEMENT A Approved for public release; Distribution Unlimited		<p style="text-align: center;">DTIC ELECTED</p> <p style="text-align: center;">DEC 16 1987</p> <p style="text-align: center;">H</p>														
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) VLSI, Hercules, CAD, nMOS, CMOS, electromigration		<table border="1"> <tr><th colspan="2">Availability Codes</th></tr> <tr><td>Dist</td><td>Avail and/or Special</td></tr> <tr><td>A-1</td><td></td></tr> </table>	Availability Codes		Dist	Avail and/or Special	A-1									
Availability Codes																
Dist	Avail and/or Special															
A-1																
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) We present new accurate slope based models for efficient computation of both the switching and direct currents in nMOS and CMOS. These models, along with the depth-first search of stages, are incorporated into a power analysis CAD tool Hercules. In addition to the current levels in a circuit, Hercules reports the worst case voltage drops from the power pin to the device drain/source validating the power bus design. The metal segments in the power/ground bus having potential electromigration problem are flagged. All the cascaded drivers whose size ratio deviates significantly from the ideal ratio based on the power consumption are also detected.																



DTIC ELECTED
DEC 16 1987
H

Hercules: A Power Analyzer for MOS VLSI Circuits

Akhilesh Tyagi

Department of Computer Science
University of Washington
Seattle, WA 98195

Technical Report 87-11-10

†Presented at the IEEE International Conference on Computer-Aided Design, ICCAD-87, November, 1987.

• Supported in part by DARPA under Contract MDA 903-85-K-0072.

87 12 8 081

Hercules: A Power Analyzer for MOS VLSI Circuits

AKHILESH TYAGI

University of Washington, Seattle

Abstract

We present new accurate slope based models for efficient computation of both the switching and direct currents in nMOS and CMOS. These models, along with the depth-first search of stages, are incorporated into a power analysis CAD tool *Hercules*. In addition to the current levels in a circuit, *Hercules* reports the worst case voltage drops from the power pin to the device drain/source validating the power bus design. The metal segments in the power/ground bus having potential *electromigration* problem are flagged. All the cascaded drivers whose size ratio deviates significantly from the ideal ratio based on the power consumption are also detected.

1 Introduction

Within the past decade, the CAD community has developed an impressive array of graphics layout systems, geometric design rule checkers, circuit extractors, simulation tools and timing analyzers to manage the mass complexity [HJK*85] of the current day VLSI digital designs. Unfortunately we do not yet have a CAD tool to help us detect the problems associated with the excessive power consumption. We have reached a point where we have a technology, gallium arsenide, that is primarily limited by its power requirements. A gallium arsenide gate at peak performance level, can dissipate between 1mW to 5mW of power. Even the domino GaAs circuits reported [YCL87] have the power requirements upwards of 0.55mW/gate. Examples of nMOS systems, whose cost is determined primarily by the power supplies and the cooling peripherals, abound in the literature. There are cases when a CMOS circuit will not function correctly because the effect of excessive voltage drops in the power bus was not taken into account. These are precisely the reasons why we felt a definite need for a CAD tool that can validate a design with respect to the power consumption.

We know of two programs that estimate power consumption of nMOS circuits only. The Berkeley tools included a power estimator called *powest* for nMOS [Cme82]. It simply counts the number of pull-up load devices (e.g. depletion load). For each type of pull-up device, e.g. depletion mode n-channel, a fixed number (user modifiable) is used to represent the fraction of time it would usually be on. The product of number of devices and this fraction is reported as the power consumption. This information has a very limited use. More recently, Jeff Wilson [Wil85] developed *puranal*, which is a more sophisticated tool. It retains the topology of the power bus as a *metal tree*. *puranal*

uses the metal tree representation to report the pin to device terminal voltage drops. However, *puranal* can deal only with a single power pin and one layer metal. Moreover it breaks the metal bus structure loops arbitrarily, which can give rise to very pessimistic numbers while dealing with a commonly encountered form of a loop in an array, a *comb*. Unfortunately, the load current estimation algorithms are only a small improvement over *powest*.

The two main problems a power analysis tool needs to address are the current flow estimation and the power bus evaluation. We propose the slope based models for calculating the switching current levels. Ousterhout [Ous84] has used slope based models for computing the effective resistances of MOS devices in the delay estimator *crystal*. However, in the CMOS case, estimating the load switching current is not the end of the story. Due to slow rising input transitions, there could be a low impedance direct path between *Vdd* and *Gnd* for a considerable duration. Veendrick [Vee84] observes that this direct current component in CMOS inverters can be as high as the load current. Thus any system estimating the current flow in a CMOS circuit ought to account for both the load current and the direct current. We model the total charge flow due to the short circuit direct current component in the slope framework. This gives us a very accurate model of the total switching current. We observe at the outset that our techniques and models are of a very general nature. They can be easily specialized to any enhancement or depletion based technology like nMOS, CMOS or GaAs. We have used these models in *Hercules*, a prototype program written in C running on a VAX-11/780.

As we mentioned earlier, the other problem is that of estimating the voltage drops in the power bus. In that direction, we extend the metal tree idea of Wilson [Wil85] to accommodate circuits with multiple power pins, multilayer metal bus and loops. It is not very often that a circuit designer routes power with loops in the power distribution metal bus structure. Nonetheless, we observed that in most of the cases when a loop is introduced, it is a *comb* loop (described later). Most of the array constructs tend to use it. Our aim was to analyze these loops efficiently. We do not yet have an efficient method of dealing with a general kind of loop.

The innards of *Hercules* consist of current estimation and metal tree evaluation algorithms, which are conceptually independent problems. For this reason, we describe them in different sections. We explain the current estimation algorithms in Section 2 followed by the power tree evaluation algorithms in

1 Current Estimation Algorithm

We wanted to be able to model the current flow as an instance of *max flow* problem with charge being the commodity pushed through the network. The switch model provides the ideal starting point for this idea. Each switch can be thought of as a node in the commodity network with the switch capacity (transistor width) and switch status (*on* or *off*) determining the network node capacity. Applying this idea to the whole circuit can still be expensive. For an exact analysis, we need to determine the state of all the switches in the network to work out the node capacities. This involves a complete simulation of the circuit for all 2^n input vectors (for an n input circuit), to determine the worst case. Thus, we need to take a value-independent approach to control the problem complexity, like the timing verifiers TV [Jou83] and Crystal [Ous85] do. The *stage* decomposition provides just the right abstraction for an approximate analysis. A *stage* is a chain of transistors leading from a strong voltage source (like *Vdd*, *Ground*, *input* or a *highly capacitive bus*) to an output or a gate. Typically a stage consists of a logic gate along with all the pass transistors following it. Refer to Figure 1 for an example of a stage. Note that a chain of transistor channels forms an electrical path from a voltage source to the output of a stage. Thus, the flow of charge through a stage is conserved. One transistor in

The stage activated by signal in1 going from 0 → 1 is shown by the thick line. It starts with the n-channel connecting to in1, followed by two pass transistors A and B, terminating in the gate C.

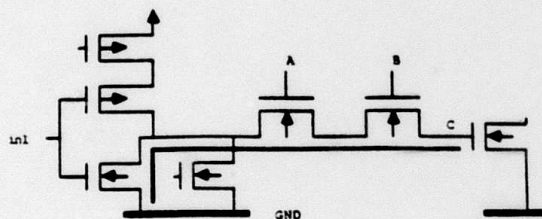


Figure 1: An Example of a Stage

a stage is distinguished as the *trigger*. It is the transistor directly driven by the signal, *e.g.*, for the stage depicted in Figure 1 the *n*-type transistor driven by *in1* is the trigger. All the other transistors in a stage are assumed to be fully *on*, unless otherwise specified by the user. This gives rise to a worst case estimate.

As we had mentioned earlier, we need to be able to estimate the short-circuit direct current to analyze a CMOS circuit accurately. Thus the total current flow has two com-

ponents. Both these components, load and direct, need to be calculated. We compute both the load current and the direct current at the stage level. We also distinguish between the average and peak current levels. The average current is total current averaged over a clock period. The average figure should be used to determine if a circuit is prone to such statistical phenomenon as metal migration. The peak current is the maximum level of current during a clock period. The peak current figure is used in estimating the worst case voltage drops in the power bus.

Before *Hercules* can be instructed to do power analysis, it has to read the circuit to be analyzed in CIF form. It builds up a metal tree data structure (described in Section 3) out of CIF description. A user can pose a question such as, "What are the repercussions of clock ϕ_1 going from 0 → 1? Which metal wire segments are susceptible to electromigration?". The stages are traced out in a depth first manner. *Hercules* finds all the nodes that can be affected by this clock transition. These nodes are the basis of the initial set of stages. Each of these stages is handled in turn. For each stage, the average and peak load and direct current calculations are performed which associates a current flow value with each node in the stage. The output node of each of these stages can activate another set of stages. This set of *children* stages is visited next.

1.1 The Load Current Estimation

In a stage, every transistor other than the trigger is assumed to be fully *on*. Thus their current flow could be as high as their saturation current. The trigger transistor, on the other hand, is going through an input transition. The rise time of the input signal, the gain β of the device, and the load determine the total amount of current flow. The key to an efficient implementation of this model is that all these factors can be combined into a single number called *rise time ratio*, which is the input signal rise time divided by the native rise time. Native rise time is the rise time of the output node when a step input is applied. The information about the load current can be extracted from SPICE output on a device with different rise time ratio values. This can be stored in a table indexed by the rise time ratio. *Hercules* uses this table to estimate the current flow through a stage. Circuit designers do not use the whole continuum of the device sizes available to them. In practice, people tend to use only a fixed set of sizes in a given circuit. Thus it is very feasible to build such a table, and fortunately it is also compact. For the rise time ratios exceeding the table bounds, extrapolation is used. Thus the current modeling for the trigger device costs one table lookup and some arithmetic. This approach was first used by Ousterhout [Ous84] to model device resistances.

At this point, we know the current flow capacities of every transistor in the stage. The peak load current through a stage is the minimum of the current flow capacities of

all the transistors in the stage. The average charge flow through a stage is the product of total load capacitance and voltage swing. The average load current through a stage is estimated by dividing the average charge by the delay through that stage. The side paths from a stage are assumed to be *off*. In practice, in a structure like a pass transistor array (barrel shifter), not many such paths originating at the same node seem to be active simultaneously.

1.2 Direct Current Models

In CMOS static logic gates, there is a direct (short-circuit) current component when the input signal switches. Clearly the RC model would not be sufficient to model this component, because it assumes step function input signals. We noticed that the total charge that flows from *Vdd* to *Gnd* due to direct current is a function only of the *rise time ratio*. Contrast it with the load current model, where it is the load current, rather than charge, that is a function of the rise time ratio. Once again, the information about direct charge flow can be extracted from many output decks of SPICE runs on a device with a range of rise time ratio values. Two tables indexed by the rise time ratio can be built, one each for the average and peak direct charge levels. *Hercules* can then use these tables to model the average and peak direct current flow through a stage with an inverting structure. The stage building algorithm tags each stage that contains a chain of *p*-type and *n*-type devices between

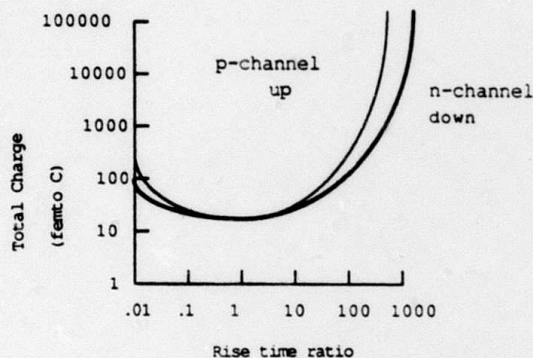


Figure 2: A Plot of Total Direct Charge Flow Vs. Rise Time Ratio

Vdd and *Gnd* with a common input. To recapitulate, this technique works for the following reason. Most of the circuits are designed with few combinations of transistor and load sizes. SPICE runs on small pieces of circuits give almost all the information about the operating environment of these small circuits within a very large circuit.

Figure 2 shows the dependence of total charge flow due to direct current on the rise time ratio for *n*-channel and *p*-channel devices. Note that the charge flow is directly proportional to the power consumption due to direct current. For the rise time ratio of 1, the total charge flow is

minimized. Veendrick also makes the same observation in [Vee84] and uses this fact to design buffers optimal with respect to power consumption. The peak and average direct currents are maximum when there is no load. Both of them tend to decrease with increasing load. The reason for it is that a part of the direct current goes into driving the load. With a higher load, the amount of excess charge available to flow from *Vdd* to *ground* is less. With rise time ratio equal to one, both the input and output signal speeds are balanced. If the input is driven at a slower rate than this, then the time during which the inverter is in short-circuit mode ($V_{inu} - V_{thn} \leq V_{ir} \leq V_{inu} + V_{thp}$) increases. This gives rise to higher direct charge flow as in Figure 2. On the other hand, if the input signal is faster than this, then the load cannot absorb all of the direct current. Thus, there is more of the excess charge available to flow between *Vdd* and *ground*. In any case, Veendrick remarks that for a rise time ratio of one, the short-circuit power dissipation would be less than 20% of the total power dissipation. Now we have a quantitative measure of what it means to have a good driving ratio between the driver size and the load. We can easily identify all the inverters (stages) with a rise time ratio within δ of 1, for a user specified δ . The user can choose a δ based on how finely tuned a system (s)he wants and *Hercules* will flag all the stages outside this bound. Note that *Hercules* computes the rise time ratio for each stage, anyway, to model the current flow.

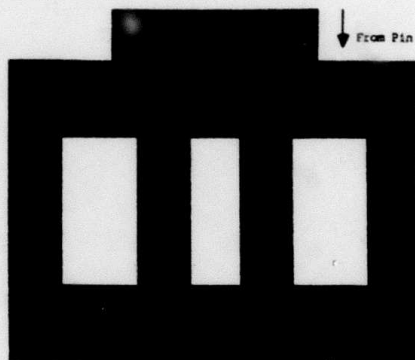


Figure 3: An Example of a Comb Structure

2 Metal Bus Data Structure

Distributing *power* and *ground* with the least resistance is one of the major concerns in a design. Many a time, a circuit fails to perform because the drain terminals receive a voltage level significantly lower than *Vdd* and the source terminals are way above *ground*, although the external power supply is functioning properly. What can be worse is that different subsystems might be receiving different operating voltages.

How should the loops in *power/gnd* structure be handled? If the loops are broken at an arbitrary edge, there could be a large error in the power bus resistance calculation. We observed that it is very seldom that a designer routes power with looping structures. However the pattern

shown in Figure 3 is used quite often to feed power to an array of cells. The bottom line introducing the loops reinforces the distribution. *Hercules* attempts to do a good job with comb loop structures. We have an exact and efficient (linear time) algorithm for analyzing a power bus with comb loops. For the details of the Metal Tree Data Structure (MTDS), the reader is referred to the author's thesis [Tya87].

3 Performance

Both the short-circuit current and load current models were validated with presently operational version of *Hercules*. *Hercules* was run on a control PLA and a register file for a 32-bit microprocessor, QuarterHorse [HJK*85], designed at the University of Washington. The control PLA has 19 inputs, 68 outputs and 93 implicants. The register file has 32, 32-bit registers designed in the cross-coupled static style. *Hercules* was also tried on some 3 stage NAND and NOR networks. The results are encouraging. For all the cases, the average load current and the average direct current reported are within 25% of the SPICE reported numbers. However the peak load current was overestimated by as much as 100%. Recall that the peak load current is the minimum of the peak currents of all the transistors in a stage. The peak direct current is calculated by dividing the total charge flow by the time input signal takes to go from $V_{inv} - V_{thn}$ to $V_{inv} + V_{thp}$. This number was also found to be off by as much as 80% on the higher side. We believe that we can do better than this with the peak load current calculation. We still do not know how to better predict the peak direct current.

4 Acknowledgements

The idea that max-flow could be useful in current analysis was suggested to me by my advisor, Larry Snyder. I would like to thank him for many invaluable discussions. The need to consider the direct current component in CMOS was pointed out to me by Bill Beckett. This work has benefitted a lot from discussions with Bill Beckett, Larry McMurchie and Bill Yost. Mike Foster of Columbia University suggested using a linear time band matrix algorithm within a comb loop. It is a pleasure to acknowledge the stimulating discussion with him. I also wish to acknowledge the influence of John Ousterhout's and Jeffrey Wilson's work on the way *Hercules* evolved.

References

- [Cme82] R. Cmelik. *Berkeley VLSI CAD Tools Manual*. Computer Science Dept., University of California, Berkeley, 1982.
- [HJK*85] S. Ho, B. Jinks, T. Knight, J. Schaad, L. Snyder, A. Tyagi, and C. Yang. The Quarter Horse:

A Case Study in Rapid Prototyping of a 32-bit Microprocessor Chip. In *IEEE Proceedings of the International Conference on Computer Design: VLSI in Computer*, IEEE Computer Society, 1985.

- [Jou83] N.P. Jouppi. Timing Analysis for nMOS VLSI. In *Proceedings of 20th Design Automation Conference*, ACM-IEEE, 1983.
- [Ous84] J.K. Ousterhout. Switch-Level Delay Models for Digital MOS VLSI. In *Proceedings of 21st Design Automation Conference*, ACM-IEEE, 1984.
- [Ous85] J.K. Ousterhout. A Switch-Level Timing Verifier for Digital MOS VLSI. *IEEE Transactions on Computer Aided Design*, July 1985.
- [Tya87] A. Tyagi. *The Role of Energy in VLSI Computations*. PhD thesis, Department of Computer Science, University of Washington, Seattle, 1987.
- [Vee84] H. J. M. Veendrick. Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits. *IEEE Journal of Solid State Circuits*, August 1984.
- [Wil85] J. Wilson. *Analysis of Power Requirements inside of nMOS Integrated Circuits*. M.S. Thesis, Computer Science Dept., Oregon Graduate Center, Beaverton, 1985.
- [YCL87] L. Yang, R. Chakharapani, and S. Long. A High Speed Dynamic Domino Circuit Implemented with GaAs MESFETs. *Journal of Solid State Circuits*, submitted, 1987.

This work was supported in part by DARPA under Contract MDA903-85-K-0072.