Power Optimization in Logic Isomers

Ramesh Panwar[§†], David Rennels[§†], and Leon Alkalaj[§]

[§]Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

[†]Department of Computer Science
University of California
Los Angeles, CA 90024

panwar, rennels, leon@telerobotics.jpl.nasa.gov

Abstract

Logic isomers are labeled, 2-isomorphic graphs that implement the same logic function. Logic isomers may have significantly different power requirements even though they have the same number of transistors in the implementation. The power requirements of the isomers depend on the transition activity of the input signals. The power requirements of isomorphic graph isomers of n-input NAND and NOR gates are shown. Choosing the less power-consuming isomer instead of the others can yield significant power savings. Experimental results on a ripple-carry adder are presented to show that the implementation using the least power-consuming isomers requires approximately 10% less power than the implementation using the most power-consuming isomers. Simulations of other random logic designs also confirm that designs using less power-consuming isomers can reduce the logic power demand by approximately 10% as compared to designs using more power-consuming isomers.

1 Introduction

The power consumed in CMOS VLSI primarily results from charging and discharging the node capacitances in the VLSI circuit. The implementation of a logic function can be repsented by a labeled graph. There are several different implementations possible even if we constrain the implementations to use the same number of transistors. Logic isomers are 2-isomorphic graphs that implement the same function. The isomers of a logic function have the same number of transistors but may have different CV^2f power requirements depending on the transition activity of the inputs. This paper addresses the power requirements of isomers of n-input NAND and NOR gates which can be represented by isomorphic graphs. The power requirements of general 2-isomorphic graph isomers will be the topic of a future paper. We show that a 2-input NAND gate has two isomers which may have different power requirements because the inputs to the gate are not equivalent from a consideration of the CV^2f power requirements even though they may be logically equivalent. We also show expressions for the power requirements of isomers of n-input NAND and NOR gates. We then present simulation results of a ripple-carry adder to show how the different power requirements of the isomers may lead to significant power savings by a logic designer who chooses to implement the design with the least power-consuming isomers based on transition activity simulations. The simulations were performed using a prototype TRansition ACTivity SIMulator called TRACSIM which we have developed for the purpose of studying transition activity in VLSI designs. Our simulation results on the ripple-carry adder and other random logic designs show that VLSI designers may be able to reduce the CV^2f power demand by approximately 10% by choosing the least power-consuming isomers in the design.

*Partial support for this work was provided by the Office of Naval Research under grant N00014-91-J-1009 at UCLA and the Advanced Flight Computing Program, NASA Code-C OACT RTOP 586-03 at JPL.

2 CV^2f Power

2.1 Node Activity

The CV^2f power [1] dissipated by a node depends on the switching behavior of the node. If the node switches an average of δ times every cycle, then the power dissipated by the node is δCV^2f. One needs to take into account the dynamic transitions rather than the logical transitions made by the node. To illustrate the difference between dynamic transitions and logical transitions, consider the circuit shown in figure 1. Let

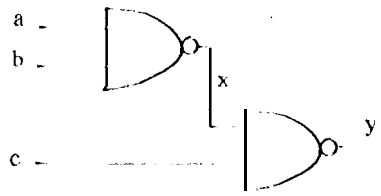

Figure 1: Example circuit for dynamic transitions

us assume that the inputs (a, b, and c) come from registers or flip-flops that have the same output delay so that they all change at some time instant t_0 in the clock cycle under consideration. Let us assume that a, b, and c were all 0 before the start of the current clock cycle and that they all change to 1 in the current clock cycle. Prior to the start of the clock cycle, the value of signal y is 1. Assuming the two NAND gates have the same delay (Δ), signal y will change to 0 at time $t_0 + \Delta$ when the effect of the change of input c propagates to the output y. Signal y will then change back to 1 at time $t_0 + 2\Delta$ when the effect of the changes at the a and b inputs propagates to the output y. Signal y thus undergoes two dynamic transitions (manifested by a "glitch") even though it does not undergo any logical transition during the cycle under consideration. In other words, a node may logically switch only once during a cycle. However, a node may dynamically switch multiple times before it settles in a machine cycle depending on the arrival of the inputs.

Definition 1 *The transition activity (δ_x) of a node (x) is defined to be the average number of dynamic transitions made by the node in a single cycle.*

2.2 Logic Isomers

A CMOS logic circuit may be represented by a labeled graph. The graph representing a CMOS circuit is constructed such that:

- The vertices in the graph represent the source/drain connections of the transistors in the circuit
- The edges in the graph are the transistors that connect particular source-drain vertices

Figure 2 shows a circuit which implements the 4 input function $f(a,b,c,d) = a'b'c' + a'b'd'$ along with its labeled graph. Consider a 2-input NAND function of two inputs x and y. Figure 3 shows one implementation of the NAND function along with its graph representation. Figure 4 shows another implementation of the NAND function along with the graph representation. The two graphs are isomorphic but the two implementations may have different power requirements depending on the transition activity of the inputs. The two different isomorphic graphs implementing the NAND function are referred to as logic isomers of the NAND function. There may be non-isomorphic graph implementations of a function that have the same number of transistors (edges). An example of a function which has non-isomorphic graph implementations is the 4-input function shown in figure 2. Another implementation of the function is shown in figure 5 along with its graph representation. The two implementations of the 4 input function are 2-isomorphic [2]. This paper will address power requirements of only n-input NAND or NOR function realizations which have isomorphic graphs; the power requirements of general 2-isomorphic graph functions will be the subject of a later paper. However, for the sake of completeness, we provide the definition of a 2-isomorphic graph:

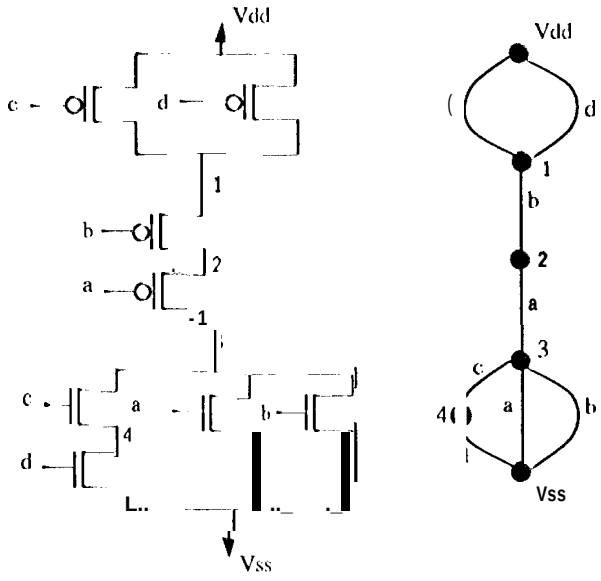

Figure 2: 4-input function

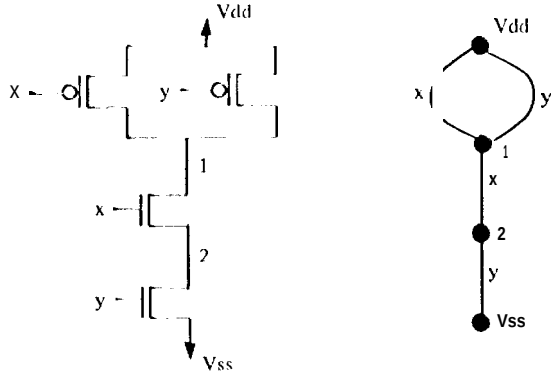

Figure 3: 2-input NAND Isomer 1

Definition 2 *Two graphs G_1 and G_2 are said to be 2-isomorphic if they have circuit correspondence which implies that there is a one-to-one correspondence between the edges of G_1 and G_2 and a one-to-one correspondence between the circuits of G_1 and G_2, such that a circuit in G_1 formed by certain edges of G_1 has a corresponding circuit in G_2 formed by the corresponding edges of G_2, and vice versa.*

Note that isomorphism is a special case of 2-isomorphism. We are now in a position to provide a formal definition of logic isomers.

Definition 3 *Logic isomers are labeled, 2-isomorphic graph realizations of a logic function*

Definition 4 *The isomeric cardinality of a logic graph is the total number of logic isomers of the graph.*

Thus, an n-input NAND gate has an isomeric cardinality of $n!$. The circuit shown in figure 2 has an isomeric cardinality of 12.

2.3 NAND Isomers

Consider the 2-input NAND gate for the purpose of illustrating the differences in power dissipation in logic isomers. Let us consider the behavior of the isomer in figure 3 for input switching activity. Both nodes 1

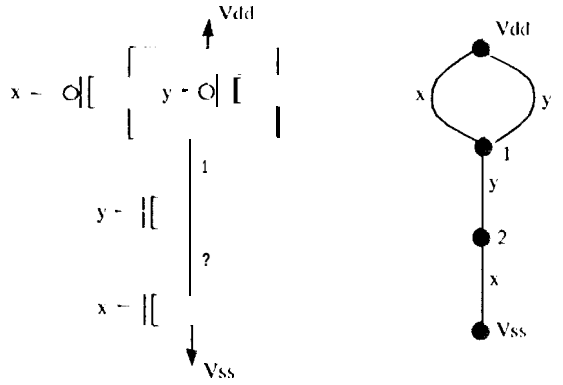

Figure 4: 2-input NAND Isomer 2

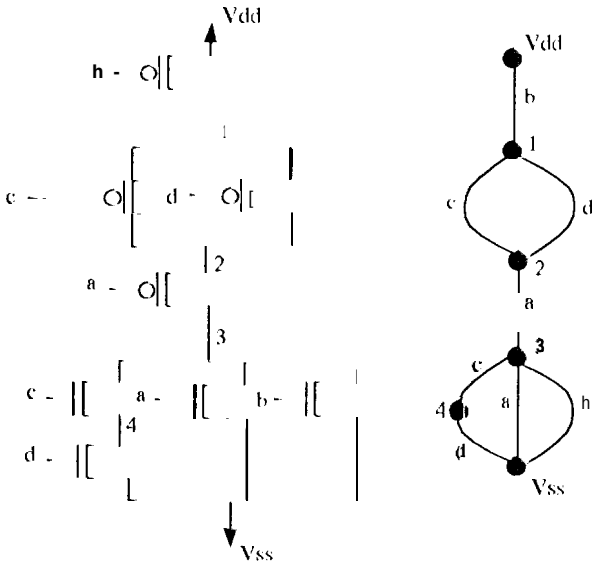

Figure 5: 4 input function

and 2 discharge if signal x is 1 while y switches from 0 to 1. Also, both nodes 1 and 2 will charge if signal x is 1 while y switches from 1 to 0. Now consider the isomer shown in figure 4. Only node 1 discharges if signal x is 1 while y switches from 0 to 1. Similarly, only node 1 charges if signal x is 1 while y switches from 1 to 0. If the capacitances of nodes 1 and 2 are denoted by C_1 and C_2, then the amount of energy required by the aforementioned operations in the two isomers are $(C_1 + C_2)V^2$ and C_1V^2 respectively. The power dissipated by the aforementioned operations is given by $\delta_y(C_1 + C_2)V^2f$ and $\delta_y C_1V^2f$ in the two isomers since the operations occur with a frequency of $\delta_y f$. The power dissipated in the two isomers due to the switching activity of signal x is given by $\delta_x C_1V^2f$ and $\delta_x(C_1 + C_2)V^2f$ respectively. Thus the total power required by the two isomers is given by:

$$P_1 = \delta_x C_1V^2f + \delta_y(C_1 + C_2)V^2f$$

$$P_2 = \delta_x (C_1 + C_2)V^2f + \delta_y C_1V^2f$$

where P_1 is the power dissipated in isomer 1 and P_2 is the power dissipated in isomer 2. The two isomers may thus have different power requirements depending on the transition activity δ_x and δ_y of signals x and y. If $\delta_x > \delta_y$, then the first isomer is preferable to the second to conserve power.

The values of the node capacitances C_1 and C_2 are as follows:

$$C_1 = 3C_d + C_L$$

$$C_2 = 2C_d$$

where C_d is the diffusion capacitance of a transistor and C_L is the load capacitance driven by the gate. The value of C_L will depend on the gate fanout and the interconnect capacitance (which depends on the net routing). The ratio of the power required by one isomer to the power required by the other is given by:

$$r_P = \frac{\delta_x(5C_d + C_L) + \delta_y(3C_d + C_L)}{\delta_x(3C_d + C_L) + \delta_y(5C_d + C_L)}$$

If we choose r_C to represent the ratio C_L/C_d, then the ratio of the power required by the isomers is given by:

$$r_P = \frac{5\delta_x + 3\delta_y + r_C(\delta_x + \delta_y)}{3\delta_x + 5\delta_y + r_C(\delta_x + \delta_y)}$$

To understand the typical value of the parameter r_C, we look at the Honeywell RICMOS process [3] which is used for ASIC design at JPL. The Honeywell RICMOS process has a typical routing capacitance of $1.4C_g$ for each load driven by the gate (where C_g is the gate capacitance of a transistor). If we assume that the average fanout is 4, then the typical value of C_L is $9.6C_g$. Thus the typical value of r_C is $9.6C_g/C_d$. Typically, $C_d = 4C_g$, which implies that a value of approximately 2.5 is typical for the parameter r_C.

2.4 Power Ratio

Definition 5 *The power ratio of a circuit is the ratio of the power required by the implementation using the most power-consuming isomer to the power required by implementation using the least power-consuming isomer.*

For a two input NAND gate, the power ratio is given by:

$$r_P = \frac{5\delta_1 + 3\delta_2 + r_C(\delta_1 + \delta_2)}{3\delta_1 + 5\delta_2 + r_C(\delta_1 + \delta_2)}$$

where we have assumed that $\delta_1 > \delta_2$. Figure 6 shows the typical power ratios of a two input NAND gate for various values of the parameter r_C. The asymptotic value of the power ratio (the value when one signal has a transition activity that is infinitely times that of the other signal) is given by $(5 + r_C)/(3 + r_C)$ which implies that the upper-bound on the power ratio for a 2-input NAND gate is 5/3. The power ratios are plotted as functions of the ratio of the transition activity of the two inputs ($r_\delta = \delta_1/\delta_2$).

The power ratio of an n-input NAND gate can be shown to be:

$$r_P = \frac{(3n + 1 + r_C)\sum_{i=1}^{n}\delta_i + 2\sum_{i=1}^{n} i\delta_i}{(n + 1 + r_C)\sum_{i=1}^{n}\delta_i + 2\sum_{i=1}^{n} i\delta_i}$$

where the activities are such that $\delta_1 > \delta_2 > \ldots > \delta_n$. It may be shown that an n-input NOR gate has a similar expression for the power ratio.

The power ratio of a 3 input NAND gate (or a 3-input NOR gate) is thus:

$$r_P = \frac{8\delta_1 + 6\delta_2 + 4\delta_3 + r_C(\delta_1 + \delta_2 + \delta_3)}{4\delta_1 + 6\delta_2 + 8\delta_3 + r_C(\delta_1 + \delta_2 + \delta_3)}$$

where $\delta_1 > \delta_2 > \delta_3$.

The power ratio of a 4 input NAND gate (or a 4 input NOR gate) can be shown to be:

$$r_P = \frac{11\delta_1 + 9\delta_2 + 7\delta_3 + 5\delta_4 + r_C(\delta_1 + \delta_2 + \delta_3 + \delta_4)}{5\delta_1 + 7\delta_2 + 9\delta_3 + 11\delta_4 + r_C(\delta_1 + \delta_2 + \delta_3 + \delta_4)}$$

To choose the best isomer of an n-input NAND or NOR gate, we now propose two rules which can be used by the logic designer (or a logic synthesis tool) to optimize the power dissipation in a logic design.

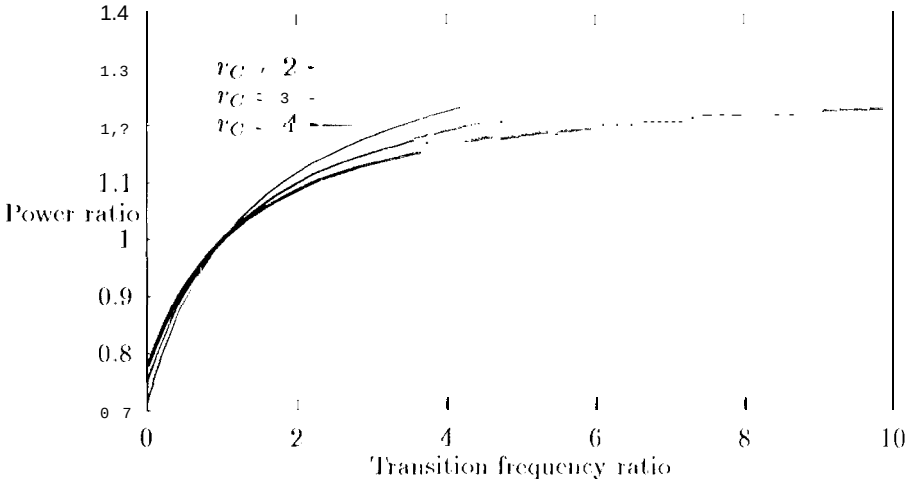

Figure 6: 2-input NAND power ratio

Rule 1 *If the logic graph has signals in series in the n-logic block, then the series signals should be ordered with respect to the signal transition activity; the lowest activity signal should be closest to the ground pin.*

Rule 2 *If the logic graph has signals in series in the p-logic block, then the series signals should be ordered with respect to the signal transition activity; the lowest activity signal should be closest to the V_{cc} pin.*

3 Experimental Results

3.1 TRACSIM

TRansition ACtivity SIMulator (TRACSIM) is a tool that we developed to study the transition activity in VLSI circuits. The input to the simulator is a netlist describing the circuit along with the transition activities of the circuit inputs. We use a simplified timing model of circuit behavior. We assume that all the gates have the same delay and that two signals changing at the inputs to a gate can cause at most one transition only on the output node. The latter assumption is quite valid since SPICE simulations confirmed that the 0(11)) 11(of a gate undergoes at most one transition even if two or more inputs change as long as the inputs change within a certain time window. The output from TRACSIM is a list of transition activities at the nodes of the circuit along with the power ratio of the circuit graph.

3.2 Ripple-carry Adder

TRACSIM was used to simulate a ripple-carry adder for various number of inputs (4 bit, 8 bit and 16 bit inputs) with different transition activities on the input signals. We simulated the ripple-carry adder with the same transition probability for each input to the adder (except the carry-in input which was set to 0) for 10000 cycles. Typically, the inputs to the adder are driven by register outputs and hence the transition activity of the inputs is less than 1. The full stage of the ripple adder consists of two circuits: one to compute the carry and the other to compute the sum. The circuit for computing the carry consists of three 2-input NAND gates and a 3 input NAND gate as shown in figure 7. The circuit for computing the sum consists of four 3-input NAND gates and a 4 input NAND gate as shown in figure 8. of the signals internal to the ripple adder, the carry signal has a fanout of 6 and the outputs of the inverters have a fanout of 2 each.

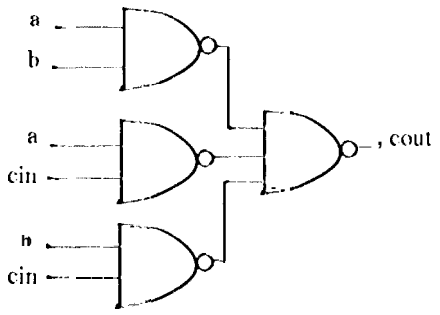

Figure 7: Carry Circuit

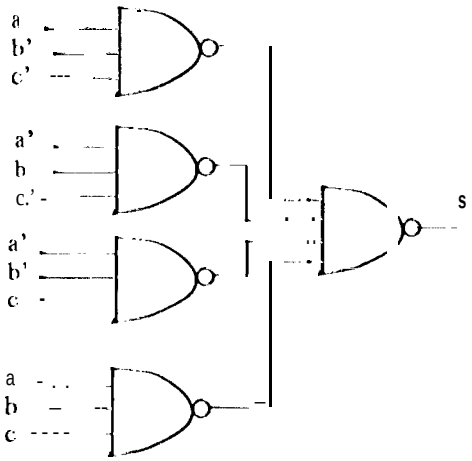

Figure 8: Sum circuit

Assuming that the routing capacitance for each load is $1.5C_g$ and that the sum outputs of the adder drive unit loads, the average r_C for the ripple adder is approximately 1. Table 1 summarizes the power ratio of the ripple-carry adder for a 4-bit adder (RA4), an 8-bit adder (RA8), and a 16-bit adder (RA16) for various values of the input transition activity (δ_i). The table shows that the power ratio of the adder varies from 1.08 to 1.12 which implies that the implementation using the least power-consuming isomers offers a power saving of approximately 10% over the implementation using the most power-consuming isomers. Figure 9

δ_i	RA4	RA8	RA16
0.1	1.12	1.12	1.11
0.3	1.12	1.12	1.11
0.5	1.12	1.11	1.11
0.7	1.12	1.10	1.09
0.9	1.12	1.09	1.08

Table 1: Ripple adder power ratio

shows the transition activity of the carry signals of the ripple-carry adder while figure 10 shows the transition activity of the sum outputs of the adder.

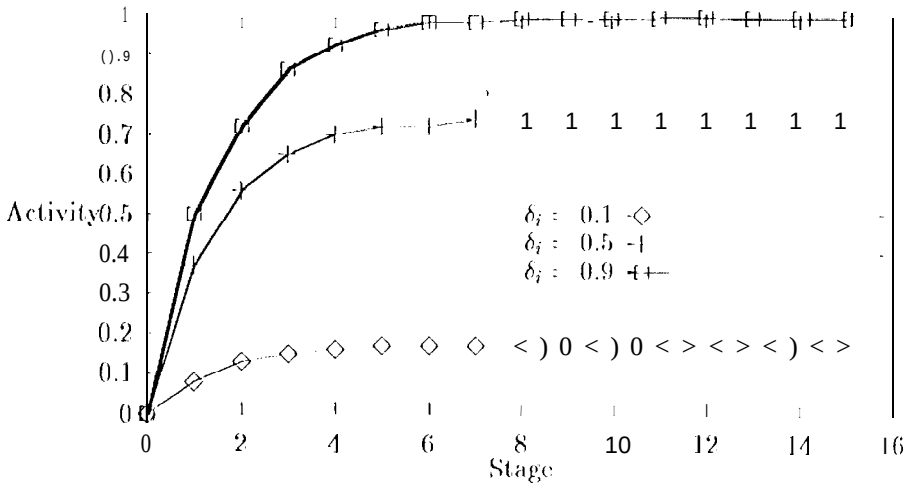

Figure 9: Ripple adder carry transition activity

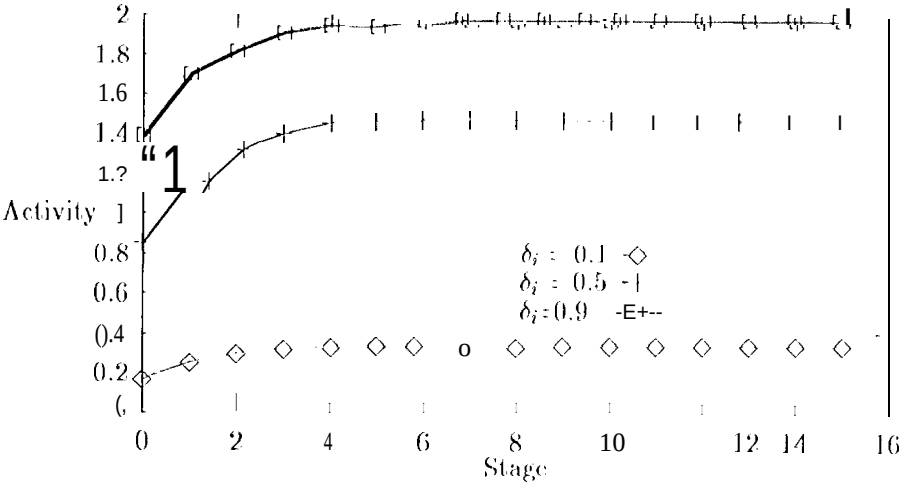

Figure 10: Ripple adder sum transition activity

3.3 Random Logic

We simulated various random logic designs with various input transition activites for $r_C = 2.5$ and found that the power ratio of 1.1 was typical for most logic designs. This implies that a power saving of 10% is possible by just a judicious choice of isomers by the logic designer.

4 Future Work

A couple of interesting problems remain for future research. The first problem that will be looked into is the generalization of the results on power ratio and the two rules of thumb to handle 2-isomorphic graphs. The second problem that will be looked into is the dependence of the power ratio on the logic depth of the circuit. We have not simulated a sufficient number of random logic designs at the time of this publication to make any conclusions regarding this problem.

5 Conclusion

We have presented a novel way of optimizing power in CMOS VLSI. We have shown that the inputs to a gate may not be equivalent from a consideration of the CV^2f power requirements even though they may be logically equivalent. Simulation results for a ripple-carry adder were presented which showed a power saving of approximately 10% in a low-power isomer design as compared to a high-power isomer design. This was also confirmed by simulating other random logic designs. The only side-effect that results from using this technique is that logically equivalent gate inputs may no longer be considered swappable by routing tools. Other than this side-effect, the power optimization offered by the technique is free.

References

[1] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. Reading, MA: Addison-Wesley, 988.

[2] N. Deo, *Graph Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1974.

[3] *The Honeywell RICMOS ASIC Library Data Book*. 99.