Chapter 2

DESIGN OF ENERGY EFFICIENT DIGITAL CIRCUITS

Bart R. Zeydel and Vojin G. Oklobdzija
ACSEL Laboratory, University of California, Davis

Abstract: Recent technology advances have resulted in power being the major concern for digital design. In this chapter we address how transistor sizing affects the energy and delay of digital circuits. The state of the art in circuit design methodology (Logical Effort) is examined and we identify its limitations for design in the energy-delay space. We examine how to explore the entire energy-delay space for a circuit and present an approach for the design and analysis in the energy-delay space which allows for energy reduction without performance penalty. Finally, we present techniques for the design of energy-efficient digital circuits.

Key words: digital circuits; energy-delay optimization; energy-delay space; performance optimization; power optimization; transistor sizing

1. Introduction

Advances in CMOS technology have led to dramatic improvements in performance while maintaining constant power density. However, as device dimensions continue to decrease traditional constant field scaling can no longer be applied [1–3]. The problem with this trend is that performance and power no longer scale proportionally across technology nodes leading to increasing power density. Further adding to this problem has been the drive to produce chips operating at higher and higher clock frequencies, which has caused circuit designers to focus solely on optimizing circuits and implementations for delay regardless of energy.

In this chapter we present models to examine the energy and delay characteristics of digital circuits and relate these characteristics to the physical

Vojin G. Oklobdzija and Ram K. Krishnamurthy (eds.),
High-Performance Energy-Efficient Microprocessor Design, 31–56.

dimensions of transistors. Using these models we will analyze Logical Effort (LE) [4, 5], the state of the art design methodology for digital circuits. The location of the LE solution in the energy-delay space is then examined to determine its applicability to energy-efficient design. The analysis demonstrates that LE does not guarantee an energy-efficient circuit. To address this we examine the entire energy-delay space for a circuit that can be obtained through transistor sizing. From this we present a simplified approach for the high-level exploration of the energy-delay characteristics of a circuit. Based on this analysis we present guidelines for the design of energy-efficient digital circuits.

2. RC Modeling of Gate Delay

Delay modeling techniques for evaluating large circuits have historically involved the simplification of current based delay modeling. The most common simplification assumes a step input, allowing for the current to be approximated over the time of interest [6–10].

2.1. Logic Gate Characteristics

In this section the physical characteristics of a CMOS logic gate are related to its delay characteristics. The layout of a CMOS inverter is shown in Figure 1. The physical parameters are W_n, W_p, L_n, and L_p which represent the widths and channel lengths of the nMOS and pMOS transistors respectively. Understanding the dependence of gate capacitance, parasitic capacitance and effective channel resistance on these physical parameters is essential to the use of RC modeling for the optimization of CMOS logic gates.

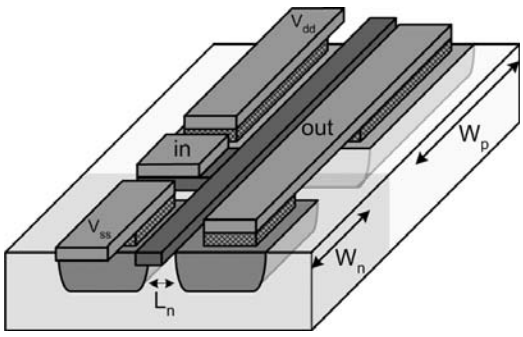

Figure 1. CMOS Inverter.

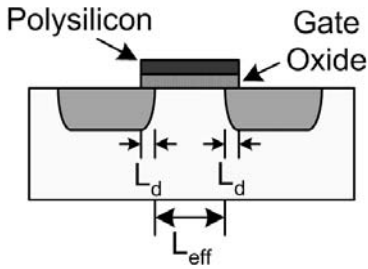

Figure 2. MOSFET Gate Capacitance.

2.1.1. Gate capacitance

Gate capacitance, C_{gate}, is a function of the effective channel length, L_{eff}, and the width of the transistor, W. The effective channel length can be calculated from the drawn transistor length as $L_{\text{eff}} = L_{\text{drawn}} - 2L_d$, as seen in Figure 2, where L_d refers to the lateral diffusion length of the source or drain into the channel. To simplify notation L_{eff} will be referred to as L.

The gate capacitance of each transistor can be calculated from the width and length of the transistor and the per area capacitance of the gate, C_{ox}.

$$C_{\text{gate}} = W \cdot L \cdot C_{\text{ox}}$$

The gate capacitance is directly proportional to the width of the transistor. Thus, as the width changes by a factor α the gate capacitance also changes by the same factor α.

$$C_{\text{gate}} = \alpha \cdot W \cdot L \cdot C_{\text{ox}}$$

The capacitance of an input to a gate, C_{in}, is the sum of the gate capacitances attached to the input. For example, the input capacitance of an inverter is:

$$C_{\text{in}} = (W_n \cdot L_n + W_p \cdot L_p) \cdot C_{\text{ox}}$$

Scaling the width of each transistor in the inverter by a factor α causes C_{in} to also scale by α.

2.1.2. Parasitic capacitance

The parasitic capacitance of a transistor has two components. The junction capacitance, C_{ja}, expressed in F per area in μm^2, and the periphery capacitance, C_{jp}, expressed in F per μm of the periphery length. These components are shown in Figure 3.

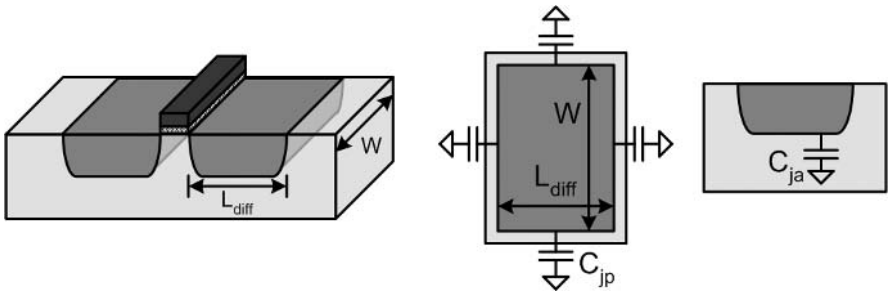

Figure 3. MOSFET Parasitic Capacitance.

The parasitic capacitance of each transistor can be computed directly from layout as:

$$C_p = C_{\text{ja}} \cdot W \cdot L_{\text{diff}} + C_{\text{jp}} \cdot (2 \cdot W + 2 \cdot L_{\text{diff}})$$

Parasitic capacitance is only roughly proportional to changes in gate width by α, due to its constant term $2C_{\text{jp}}L_{\text{diff}}$. To simplify analysis this term is often ignored allowing for the parasitic capacitance to be proportional to α.

2.1.3. Resistance

The channel resistance, R_{channel}, in a MOSFET is dependent on its region of operation, transistor width, and channel length. In saturation R_{channel} can be expressed as follows, where μ is the mobility of the channel and λ is the Early effect:

$$R_{\text{channel(sat)}} = \frac{\partial V_{\text{ds}}}{\partial I_{d(\text{sat})}} = \frac{2 \cdot L}{W \cdot \mu \cdot C_{\text{ox}} \cdot (V_{\text{gs}} - V_t)^2 \cdot \lambda}$$

In linear or triode, R_{channel} can be expressed as:

$$R_{\text{channel(lin)}} = \frac{\partial V_{\text{ds}}}{\partial I_{d(\text{lin})}} = \frac{L}{W \cdot \mu \cdot C_{\text{ox}} \cdot (V_{\text{gs}} - V_t - V_{\text{ds}})}$$

The resistance of the channel is inversely proportional to the width of the transistor in both saturation and linear regions of operation. Thus, by changing the width of the transistor by a factor α, the resistance of the transistor changes by $1/\alpha$.

2.2. RC Delay Model

The propagation delay of a CMOS logic gate can be represented using a RC-model [6]. The model can be derived assuming a step input (Figure 4),

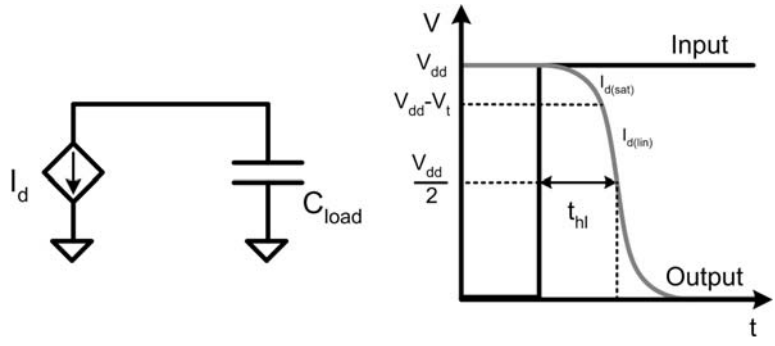

Figure 4. Step input response of a CMOS logic gate.

and related to gate capacitance, parasitic capacitance and channel resistance. The load, C_{load}, consists of the output load, C_{out}, and the parasitic load at the output of the gate, C_p. The derivation will only be shown for the high-to-low propagation delay, t_{hl}, however a similar derivation can be performed for the low-to high propagation delay, t_{lh}.

The propagation delay, t_{hl}, can be calculated from:

$$-I_d = C_{\text{load}} \cdot \frac{\partial V_{\text{out}}}{\partial t}$$

where t_{hl} is given by:

$$t_{\text{hl}} = -\int_{V_{\text{dd}}}^{V_{\text{dd}}/2} \frac{C_{\text{load}}}{I_d} \partial V_{\text{out}}$$

For a step input, the transistor will be in saturation for V_{out} from V_{dd} to $V_{\text{dd}} - V_t$. In the saturation region, the drain current is given by:

$$I_{d(\text{sat})} = \mu_n \cdot C_{\text{ox}} \cdot \frac{W}{L} \frac{(V_{\text{dd}} - V_t)^2}{2}$$

The transistor will be in the linear region for V_{out} from $V_{\text{dd}} - V_t$ to $V_{\text{dd}}/2$. In the linear region, drain current is given by:

$$I_{d(\text{lin})} = \mu_n \cdot C_{\text{ox}} \cdot \frac{W}{L} \left((V_{\text{dd}} - V_t) \cdot V_{\text{out}} - \frac{V_{\text{out}}^2}{2} \right)$$

Substituting into the integration for t_{hl}:

$$t_{\text{hl}} = -\int_{V_{\text{dd}}}^{V_{\text{dd}}-V_t} \frac{C_{\text{load}}}{I_{d(\text{sat})}} \partial V_{\text{out}} - \int_{V_{\text{dd}}-V_t}^{V_{\text{dd}}/2} \frac{C_{\text{load}}}{I_{d(\text{lin})}} \partial V_{\text{out}} = t_{\text{hl(sat)}} + t_{\text{hl(lin)}}$$

Integrating gives:

$$t_{\text{hl(sat)}} = -\frac{C_{\text{load}}}{\mu_n \cdot C_{\text{ox}} \cdot \frac{W}{L} \frac{(V_{\text{dd}}-V_t)^2}{2}} \int_{V_{\text{dd}}}^{V_{\text{dd}}-V_t} \partial V_{\text{out}} = \frac{2 \cdot V_t \cdot C_{\text{load}}}{\mu_n \cdot C_{\text{ox}} \cdot \frac{W}{L}(V_{\text{dd}} - V_t)^2}$$

$$t_{\text{hl(lin)}} = -\frac{C_{\text{load}}}{\mu_n \cdot C_{\text{ox}} \cdot \frac{W}{L}} \int_{V_{\text{dd}}-V_t}^{V_{\text{dd}}/2} \left(\frac{1}{(V_{\text{dd}} - V_t) \cdot V_{\text{out}} - \frac{V_{\text{out}}^2}{2}} \right) \cdot \partial V_{\text{out}}$$

$$= \frac{C_{\text{load}}}{\mu_n \cdot C_{\text{ox}} \cdot \frac{W}{L}(V_{\text{dd}} - V_t)} \cdot \ln\left(3 - 4\frac{V_t}{V_{\text{dd}}}\right)$$

Substituting $t_{\text{hl(sat)}}$ and $t_{\text{hl(lin)}}$ into t_{hl}:

$$t_{\text{hl}} = \frac{C_{\text{load}}}{\mu_n \cdot C_{\text{ox}} \cdot \frac{W}{L}(V_{\text{dd}} - V_t)} \cdot \left(\frac{2 \cdot V_t}{V_{\text{dd}} - V_t} + \ln\left(3 - 4\frac{V_t}{V_{\text{dd}}}\right) \right)$$

The channel resistance is physically dependent on W, L, μ_n, and C_{ox}. These terms can be grouped to describe the effective resistance of the channel, R_{channel}:

$$R_{\text{channel}} = \frac{L}{\mu_n \cdot C_{\text{ox}} \cdot W \cdot (V_{\text{dd}} - V_t)}$$

The remaining terms can be grouped into a constant determined from V_{dd} and V_t:

$$\kappa = \left(\frac{2 \cdot V_t}{V_{\text{dd}} - V_t} + \ln\left(3 - 4\frac{V_t}{V_{\text{dd}}}\right) \right)$$

The resulting delay of a gate can be expressed as:

$$t_{\text{hl}} = \kappa \cdot R_{\text{channel}} \cdot C_{\text{load}} = \kappa \cdot R_{\text{channel}} \cdot (C_{\text{out}} + C_p)$$

In this form, delay is seen to be linear with respect to C_{load}. A graphical representation of this model is shown in Figure 5. R_{up} and R_{down} denote the equivalent pull-up and pull-down resistance of a gate.

We would like to observe the delay dependence as transistor widths are scaled by a factor α. The original resistances and capacitances will be referred to as the template. The resistance of a gate changes inversely with α, as:

$$R_{\text{channel}} = R_{\text{template}}/\alpha$$

The input capacitance and parasitic capacitance of the gate both change directly with α:

$$C_{\text{in}} = C_{\text{template}} \cdot \alpha$$
$$C_p \approx C_{p(\text{template})} \cdot \alpha$$

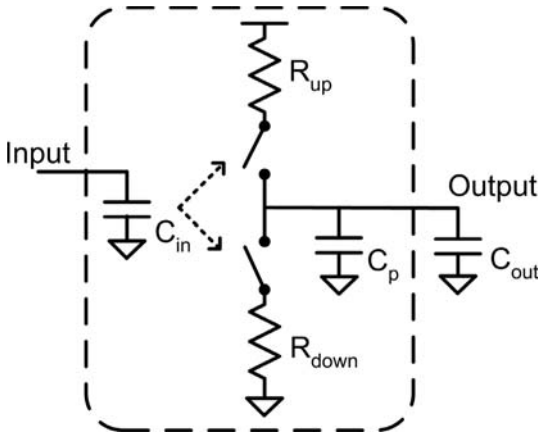

Figure 5. RC Model for a CMOS gate.

Plugging the scaled values for resistance and capacitance into the RC delay model yields:

$$t_d = \kappa \cdot \left(\frac{R_{\text{template}}}{\alpha}\right)(C_{\text{out}} + \alpha \cdot C_{p(\text{template})})$$

It is observed that the parasitic delay of a gate does not change with the size of the gate.

$$t_d = \kappa \cdot \left(\frac{R_{\text{template}}}{\alpha}\right) \cdot C_{\text{out}} + \kappa \cdot R_{\text{template}} \cdot C_{p(\text{template})}$$

However, the delay associated with a constant load changes inversely with the sizing factor α. Through substitution, delay can be expressed in terms of C_{in} and C_{out} of the gate instead of using α.

$$t_d = \kappa \cdot \left(R_{\text{template}} \cdot C_{\text{template}} \cdot \left(\frac{C_{\text{out}}}{C_{\text{in}}}\right) + R_{\text{template}} \cdot C_{p(\text{template})}\right)$$

2.3. Logical Effort Delay Model

In 1991 R. F. Sproull and I.E. Sutherland suggested that a technology independent delay could be obtained by normalizing the RC-delay model of a gate [4, 5]. They suggested that the delay of a gate be normalized to the per fanout delay of an inverter.

$$t_d = \kappa \cdot R_{\text{inv}} \cdot C_{\text{inv}} \left(\frac{R_{\text{template}} \cdot C_{\text{template}}}{R_{\text{inv}} \cdot C_{\text{inv}}} \cdot \left(\frac{C_{\text{out}}}{C_{\text{in}}}\right) + \frac{R_{\text{template}} \cdot C_{\text{parasitic}}}{R_{\text{inv}} \cdot C_{\text{inv}}}\right)$$

The technology dependent constant is referred to as τ.

$$\tau = \kappa \cdot R_{\text{inv}} \cdot C_{\text{inv}}$$

The logical effort (g), or relative drive capability, of each gate is given by:

$$g = \frac{R_{\text{template}} \cdot C_{\text{template}}}{R_{\text{inv}} \cdot C_{\text{inv}}}$$

The parasitic delay (p) of each gate is given by:

$$p = \frac{R_{\text{template}} \cdot C_{\text{parasitic}}}{R_{\text{inv}} \cdot C_{\text{inv}}}$$

The relationship of output load to input capacitance is referred to as the electrical effort (h) of the gate.

$$h = \frac{C_{\text{out}}}{C_{\text{in}}}$$

Using these terms, delay can be expressed as:

$$t_d = (gh + p) \cdot \tau$$

The logical effort of a gate can be determined by equalizing the resistance of the gate to the inverter and computing the ratio of input capacitances. The input capacitance is proportional to the sum of the gate widths attached to an input of the circuit. For example, input-a of the 2-input NOR gate in Figure 6

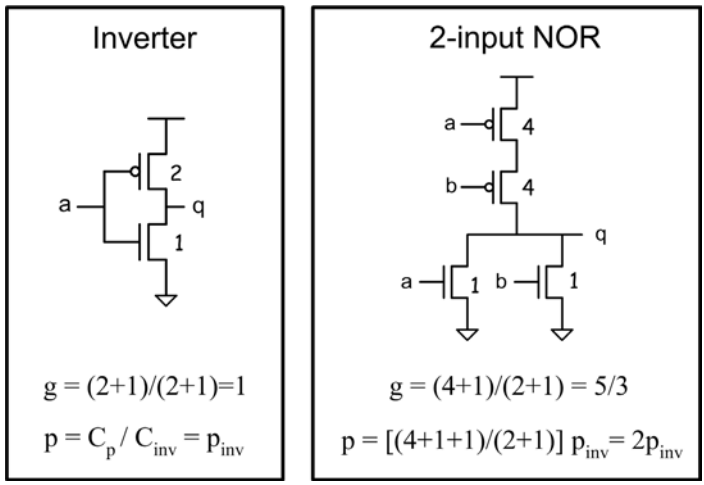

Figure 6. Logical Effort of an Inverter and a 2-input NOR gate.

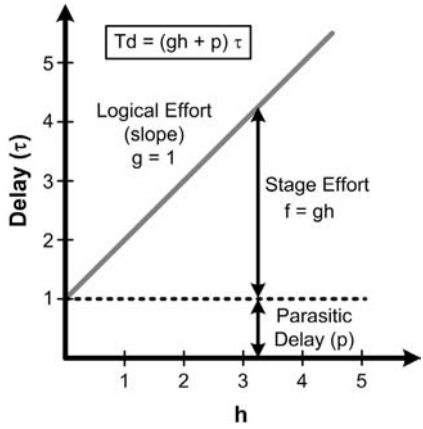

Figure 7. Logical Effort Delay Components.

has a total width of 5 which when normalized to the input capacitance of the inverter, yields a logical effort g of 5/3.

The parasitic delay can be determined from the ratio of transistor widths attached to the output node. For example, in the 2-input NOR gate the total transistor width attached to the output node is 6 which when normalized to the input capacitance of the template inverter, give a parasitic delay of $2p_{\text{inv}}$. To simplify analysis, it is often assumed that $C_{p(\text{inv})}$ equals C_{inv} which makes p_{inv} equal to 1.

A graphical representation of the LE terms is shown in Figure 7. The product of gh is referred to as the stage effort, f, and represents the delay associated with the output load of a gate. By plotting delay versus H the logical effort values can be obtained from simulation. The parasitic delay can be found from the delay intercept when h is 0, while the logical effort can be found from the slope of the delay versus h. To obtain delay in terms of τ, each delay target is normalized to the per fanout delay of the inverter.

3. Designing Circuits for Speed

Designing circuits for speed has been the focus of digital circuit designers since the inception of CMOS technology. To achieve better speed, designers initially focused on reducing the number of logic stages on the critical path. Designers soon realized that the fan-in and fan-out of circuits needed to be accounted for when analyzing circuits for speed [11]. As CMOS technology progressed, designers were also given the ability to modify transistor sizes to improve the performance of circuits. To address the issue of transistor sizing CAD tools, such as TILOS [12], were used to optimize the performance of

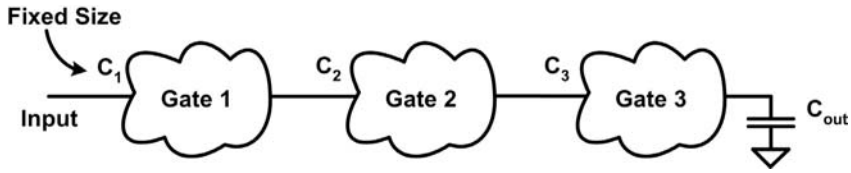

Figure 8. Chain of Gates with a Fixed Output Load and Fixed Input Size.

circuits. However, these tools offered designers little or no insight into why one design was faster than another or how gates should be sized for optimal delay. Logical Effort filled this void by providing designers with the ability to compare delay optimized digital circuits in an intuitive manner.

3.1. Delay Optimization of a Single Path Circuit

Logical Effort provides a method for optimizing the delay of a chain of gates driving a load. The constraints on the optimization are a fixed output load and a fixed input size. The derivation for delay optimal sizing of a chain of gates will be shown for the example in Figure 8.

The delay of the path can be expressed as:

$$T_{\text{path}} = \left[\left(g_1\frac{C_2}{C_1} + p_1\right) + \left(g_2\frac{C_3}{C_2} + p_2\right) + \left(g_3\frac{C_{\text{out}}}{C_3} + p_3\right)\right] \cdot \tau$$

The input capacitances of gates 1, 2 and 3 are referred to as C_1, C_2, and C_3 respectively. The minimum delay of the path with a fixed output load, C_{out}, and fixed input size, C_1, can be found by taking the derivative of the path delay with respect to C_2 and C_3.

$$\frac{\partial T_{\text{path}}}{\partial C_2} = \frac{g_1}{C_1} - g_2\frac{C_3}{C_2^2} = 0 \qquad \frac{\partial T_{\text{path}}}{\partial C_3} = \frac{g_2}{C_2} - g_3\frac{C_{\text{out}}}{C_3^2} = 0$$

Rearranging the expression yields:

$$g_1\frac{C_2}{C_1} = g_2\frac{C_3}{C_2} \qquad g_2\frac{C_3}{C_2} = g_3\frac{C_{\text{out}}}{C_3}$$

Expressed in terms of stage effort, $f_1 = f_2$ and $f_2 = f_3$. Thus the minimum delay of the path is achieved when the stage efforts of each gate. The optimal stage effort, f_{opt}, can be found from:

$$f_{\text{opt}} = \left(g_1\frac{C_2}{C_1} \cdot g_2\frac{C_3}{C_2} \cdot g_3\frac{C_{\text{out}}}{C_3}\right)^{1/3} = \left(\frac{C_{\text{out}}}{C_1} \cdot \prod_{i=1}^{3} g_i\right)^{1/3}$$

Generalized to an N-stage chain of gates:

$$f_{\text{opt}} = \left(\frac{C_{\text{out}}}{C_1} \cdot \prod_{i=1}^{N} g_i \right)^{1/N}$$

The following definitions are introduced to simplify discussion. The electrical effort or gain of a path, H, is defined as the ratio of output to input capacitance of the path.

$$H = \frac{C_{\text{out}}}{C_{\text{in}}}$$

The Logical Effort of the path, G, is defined as the product of the logical effort of the gates along the path.

$$G = \prod g_i$$

Using these simplifications, the optimal stage effort for a path is:

$$f_{\text{opt}} = (GH)^{1/N}$$

The optimal delay for a chain of gates is given by:

$$T_{\text{path}} = \left(N \cdot (GH)^{1/N} + \sum_{i=1}^{N} p_i \right) \cdot \tau$$

3.1.1. Example of delay optimized sizing

This example demonstrates how the sizes of the gates in Figure 9 are optimized such that the delay of the path with a fixed input size and fixed output load is minimal. The input capacitances of each gate on the path are referred to as C_{in}, C_2, C_3, and C_4, respectively.

The optimal sizing is obtained from f_{opt}:

$$f_{\text{opt}} = (GH)^{1/4} = \left(\left(\frac{5}{3} \cdot \frac{4}{3} \cdot \frac{5}{3} \cdot 1 \right) \cdot 21.87 \right)^{1/4} = 3$$

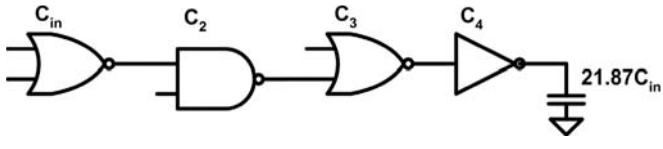

Figure 9. Example Chain of Gates.

The resulting optimal delay of the path is $T_d = (12 + 7p_{\text{inv}})\tau$. Using f_{opt}, the input capacitance of each gate can be computed as:

$$C_i = \frac{g_i \cdot C_{i+1}}{f_{\text{opt}}}$$

3.2. Delay Optimization of Circuits with Branching

Although the solution to the previous problem is useful for a simple chain of gates, it does not account for circuits with multiple paths. LE introduces branching (b) to allow for the analysis of multi-path circuits. Branching relates the off-path capacitance, $C_{\text{off-path}}$, to the on-path capacitance, $C_{\text{on-path}}$.

$$b = \frac{C_{\text{on-path}} + C_{\text{off-path}}}{C_{\text{on-path}}}$$

This often leads to confusion as the definition for electrical effort, h, includes the branching factor:

$$h = \frac{C_{\text{on-path}} + C_{\text{off-path}}}{C_{\text{in}}} = \left(\frac{C_{\text{on-path}} + C_{\text{off-path}}}{C_{\text{on-path}}}\right)\left(\frac{C_{\text{on-path}}}{C_{\text{in}}}\right)$$
$$= b \cdot \frac{C_{\text{on-path}}}{C_{\text{in}}}$$

When applying to a path it can be seen that

$$\prod_{i=1}^{N} h_i = H \cdot \prod_{i=1}^{N} b_i = HB \quad \text{where, } B = \prod_{i=1}^{N} b_i$$

Resulting in the following expression for f_{opt}:

$$f_{\text{opt}} = (GBH)^{1/N}$$

3.2.1. Multi-Path circuit optimization example

To achieve minimum delay in the multi-path circuit shown in Figure 10, the delay through Path A and Path B should be equal [13, 14].

The delay for Path A and B can be expressed as:

$$T_{\text{Path-A}} = [(g_1h_1 + p_1) + (g_2h_2 + p_2) + (g_3h_3 + p_3)] \cdot \tau$$
$$T_{\text{Path-B}} = [(g_1h_1 + p_1) + (g_4h_4 + p_4) + (g_5h_5 + p_5)] \cdot \tau$$

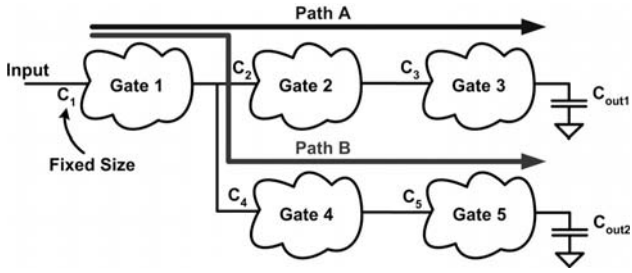

Figure 10. Example Multi-Path Circuit.

The branching at the output of Gate 1 for Path A and B can be determined as follows:

$$b_{\text{Path-A}} = \frac{C_2 + C_4}{C_2} \qquad b_{\text{Path-B}} = \frac{C_4 + C_2}{C_4}$$

Solving for C_2 and C_4:

$$C_2 = \frac{g_2 g_3 \cdot C_{\text{out1}}}{f_2 f_3} \qquad C_4 = \frac{g_4 g_5 \cdot C_{\text{out2}}}{f_4 f_5}$$

Substituting C_2 and C_4 into $b_{\text{Path-A}}$ and $b_{\text{Path-B}}$:

$$b_{\text{Path-A}} = \frac{\frac{g_2 g_3 \cdot C_{\text{out1}}}{f_2 f_3} + \frac{g_4 g_5 \cdot C_{\text{out2}}}{f_4 f_5}}{\frac{g_2 g_3 \cdot C_{\text{out1}}}{f_2 f_3}} \qquad b_{\text{Path-B}} = \frac{\frac{g_4 g_5 \cdot C_{\text{out2}}}{f_4 f_5} + \frac{g_2 g_3 \cdot C_{\text{out1}}}{f_2 f_3}}{\frac{g_4 g_5 \cdot C_{\text{out2}}}{f_4 f_5}}$$

Previously it was demonstrated that the optimal delay of a path without branching occurs when each stage has the same stage effort. Simplifying the delay to only include stage effort (by ignoring the parasitic delay difference between the Path A and B) the delay of each branch is equal when $f_2 = f_3 = f_4 = f_5$. Allowing for $b_{\text{path-A}}$ and $b_{\text{path-B}}$ to be expressed as:

$$b_{\text{Path-A}} = \frac{g_2 g_3 \cdot C_{\text{out1}} + g_4 g_5 \cdot C_{\text{out2}}}{g_2 g_3 \cdot C_{\text{out1}}} \qquad b_{\text{Path-B}} = \frac{g_4 g_5 \cdot C_{\text{out2}} + g_2 g_3 \cdot C_{\text{out1}}}{g_4 g_5 \cdot C_{\text{out2}}}$$

A special case for branching occurs when $g_2 g_3 = g_4 g_5$ and $C_{\text{out1}} = C_{\text{out2}}$. In this case $b_{\text{Path-A}} = b_{\text{Path-B}} = 2$.

While branching allows for off-path gate load to be included in LE, constant off-path loads of minimum sized gates and interconnect are not accounted for as they introduce nonlinearity into the branching computation. Further complicating branching are paths with different number of stages. Accurate accounting for these factors when optimizing for delay requires the use of numerical optimization.

Table 1. Delay Comparison of two circuits X and Y

Parasitic Delay (P)	Logic Complexity (GB)	Logic Stages (S)	Best Design for all H
$P_X = P_Y$	$G_X B_X = G_Y B_Y$	$S_X = S_Y$	Equal delay
$P_X = P_Y$	$G_X B_X < G_Y B_Y$	$S_X = S_Y$	X is faster
$P_X < P_Y$	$G_X B_X = G_Y B_Y$	$S_X = S_Y$	X is faster
$P_X < P_Y$	$G_X B_X < G_Y B_Y$	$S_X = S_Y$	X is faster
$P_X < P_Y$	$G_X B_X > G_Y B_Y$	$S_X = S_Y$	Depends on H
-	-	$S_X ! = S_Y$	Depends on H

3.3. Designing High-Performance Circuits

The delay optimal solution for a path has two components. A constant parasitic delay and a variable delay dependent on the gain of the path, H. As H decreases, the delay of the path approaches the parasitic delay.

$$T_{\text{path}} = \left(N \sqrt[N]{GBH} + \sum_{i=1}^{N} p_i \right)$$

The Logical Effort, G, of a path is constant, regardless of H. While branching, B, is approximately constant depending on the impact of nonlinearities such as wire and minimum sized gates with respect to H. These parameters define the inherent complexity of a circuit. We refer to the product of GB as the logic complexity of a circuit. By analyzing the logic complexity of a circuit in conjunction with its parasitic delay it is possible to compare circuits over a range of H to gain insight into designing high-performance circuits (Table 1).

From the table, it is seen that two circuits X and Y, which have the same number of stages and implement the same function, will always have the same delay if they have the same parasitic delay and logic complexity. Circuit X will always be the same speed or faster than Y if its parasitic delay is less than or equal to that of Y and its logic complexity is less than or equal to that of Y. However, if the circuit has less parasitic delay yet more logic complexity than the other circuit, the faster design will depend on the value of H. For implementations which use a different number of stages the best design depends on H.

4. Design in the Energy-Delay Space

CMOS technology scaling no longer has the favorable characteristics of constant power-density. As a result it is no longer possible to design solely for delay. Instead, both the energy and delay of a circuit must be accounted

for. In this section we present a basic energy model which can be combined with RC-delay modeling to provide an energy estimate for LE delay optimized points. From these points the energy-delay space of digital circuits can be explored to identify the efficient region of operation and to identify energy-efficient characteristics of circuits.

4.1. Energy Model

An energy model which yields reasonable results that can be computed directly from gate size and output load is desirable (due to its compatibility with the transistor sizing described in section 3). For hand estimation, the dynamic energy of a circuit can be computed directly from the output load of the circuit, as:

$$E = C_{\text{load}} \cdot V_{\text{dd}}^2 = (C_p + C_{\text{out}}) \cdot V_{\text{dd}}^2$$

This model neglects the energy associated with short-circuit current and leakage. The model can be improved through simulation to include the energy associated with short-circuit current and leakage. The energy of a 2-input NAND gate obtained from simulation is shown in Figure 11. A linear dependence of energy on input size and output load is observed [15].

An offset can occur at zero size due to internal wire capacitance estimation, which can be accounted for by $E_{\text{internal-wire}}$. The dynamic energy associated

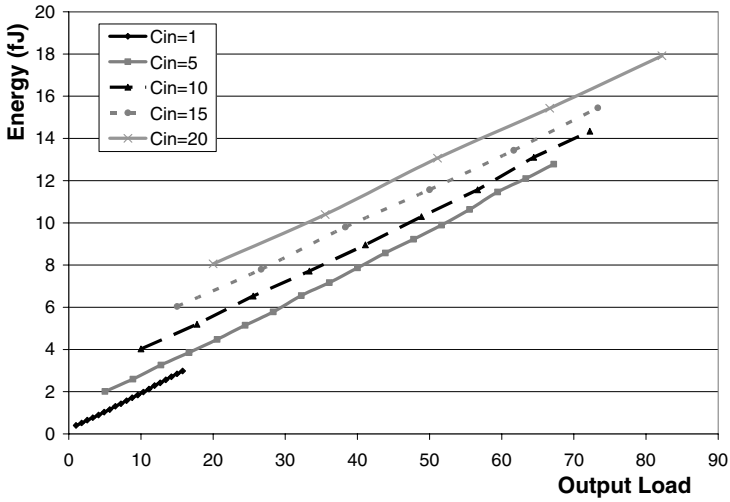

Figure 11. Energy Dependence on Input Size and Output Load for a 2-input NAND gate.

with the output of a gate can be expressed as:

$$E = E_p \cdot \text{gate size} + E_g \cdot C_L + E_{\text{internal-wire}}$$

E_p represents the energy per size and E_g represents the energy per output load. These terms can be obtained from simulation and directly account for the energy associated with output load and parasitic capacitance while providing a best fit for short-circuit and leakage current. The static energy of a gate per unit time, E_{leakage}, can be estimated by hand or obtained from simulation, from which the total static energy of the gate to be computed as $E = E_{\text{leakage}} \cdot$ gate size $\cdot$ period. The switching activity of each gate is incorporated when estimating the energy of an entire circuit.

4.2. Minimal Energy Circuit Sizing for a Fixed Output Load and Fixed Input Size

To optimize a circuit with a fixed output load and a fixed input size it is first necessary to understand where the Logical Effort design point lies in the energy-delay space. The energy-delay space obtained through changing the sizes of the second and third inverter in a chain of three inverters with a fixed output load and fixed input size is shown in Figure 12. As can be seen, the solution space is vast even for such a simple circuit. In this solution space the

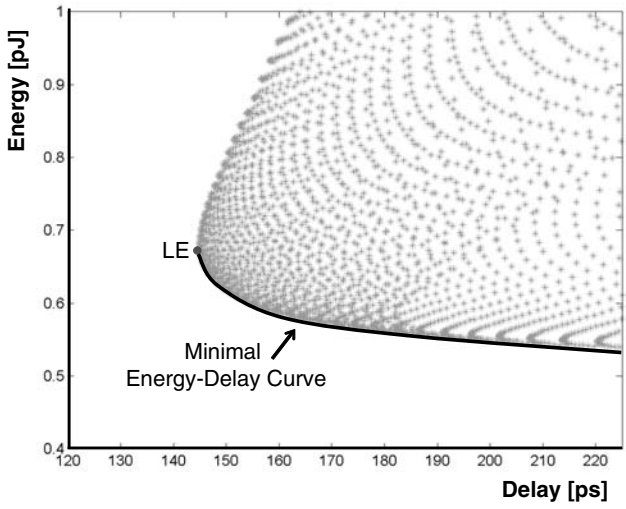

Figure 12. Energy-Delay Solution Space for a Chain of 3 inverters with a Fixed Output Load and Fixed Input Size.

delay optimized sizing of LE sets the performance limit for the circuit. Efficient design points in this solution space are those that achieve minimal energy for each delay. These points are obtained by relaxing the delay target from the LE point and resizing the circuit to reduce energy. The combined result of these optimizations yields the minimal Energy-Delay curve of a circuit for a fixed output load and a fixed input size.

It has been suggested that a tangent to this curve can be used to select an efficient design point [16–20]. For high-performance, some commonly used tangents are Energy. Delay^2(ED^2), Energy-Delay Product (EDP), and other ED^X metrics. The difficulty with designing for these metrics is that they can not be directly computed and can not be used to achieve a desired delay target or energy target. A minimal energy-delay curve for a fixed output load and a fixed input size obtained through transistor sizing along with various design metrics is shown in Figure 13.

The transistor sizings corresponding to each metric in Figure 13 are shown in Figure 14.

Energy decreases dramatically from the LE point at only a slight increase in delay. The rapid decent is due to the rippling affect of reducing the size of a gate that occurs later in the path. This weighting of gates along a path can be seen in the computation of the input capacitance of the *k-th* gate:

$$C_k = C_{\text{load}} \cdot \prod_{i=k}^{N} \frac{g_i}{f_i}$$

By changing the size of the *N-th* gate of the path by a factor α (equivalent to changing f_N by $1/\alpha$), the size of each preceding gate along the path also

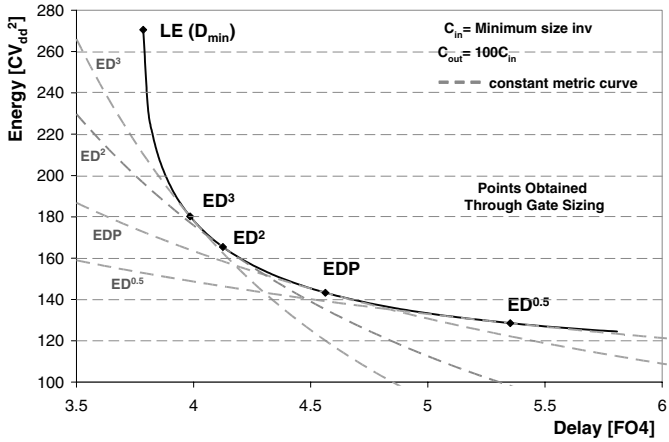

Figure 13. Minimal Energy-Delay Curve for a Chain of 6 inverters with Fixed output load and fixed input size.

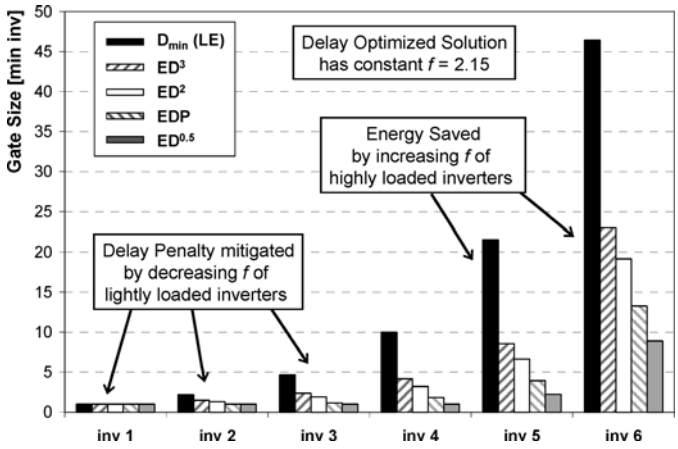

Figure 14. Corresponding Gate Sizing for Design Metrics on the Energy-Delay Curve.

changes by $1/\alpha$. It is this weighting of gates that causes the sizing to differ dramatically from the LE solution of equal f. By allowing the total delay of the path to be relaxed, the excess delay can be redistributed amongst the gates which contribute the most energy to the path (by changing f_i of these gates) to reduce the total energy [21, 22].

4.3. Circuit Sizing for Minimal Energy with a Fixed Output Load and Variable Input Size

In practice, circuit designers usually do not have the flexibility of degrading the performance of a circuit as it is often tied to the performance of the entire system. As a result, metrics which relate delay variation to energy variation are inapplicable at the circuit level. Instead the circuit should be designed for minimal energy at the desired performance target. As shown previously for a fixed delay, fixed input size and fixed output load there exists only one solution with minimal energy. However, if the input size is allowed to change, multiple energy solutions can be obtained at a fixed delay [21, 22]. The solution space obtained by varying the input size of a static 64-bit Kogge-Stone Adder [23] is shown in Figure 15.

The upper bound of the solution space consists of the delay optimized points obtained for each input size. The lower bound of the energy-delay space is constructed from the minimal energy points for each delay. Increasing the input size causes H to be reduced, allowing for performance to improve at the cost of increased energy. The minimum efficient input size for each delay is associated with the delay optimized point, while the maximum efficient input

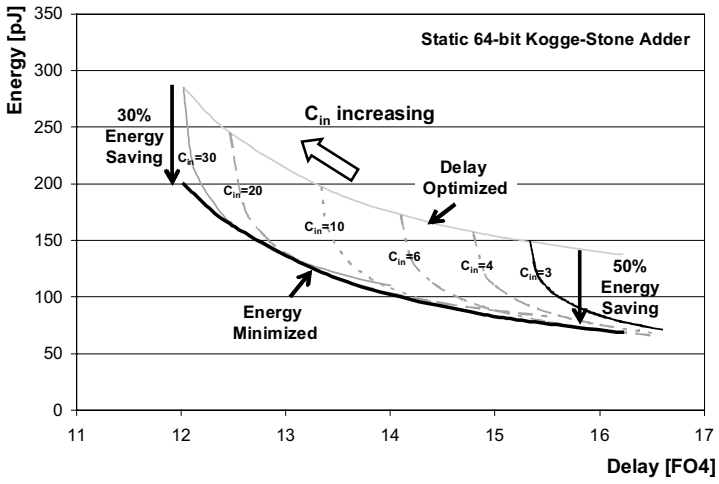

Figure 15. Energy-Delay Space for a Static 64-bit KS Adder with Fixed Output Load and Variable Input Size.

size for each delay is associated with the energy minimized point. By analyzing the complete energy-delay space of a circuit for a fixed output load, a potential 30–50% energy savings is observed in the adder example, depending on delay target.

4.3.1. Example: Energy minimization of an inverter chain

A chain of 6-inverters will be used to demonstrate how gate sizes change to achieve the same delay for different input sizes. The minimal energy sizings are shown in Figure 16 for several input sizes, with C_{out} equal to $100C_{\text{min-inv}}$ and a delay target of 18.9τ. As the input size is increased from minimal, a smaller delay can be achieved due to reduced H. The excess delay is redistributed amongst the gates to reduce total energy by reducing the sizes of the gates that impact energy the most and by increasing the sizes of the gates that have a smaller impact on energy to achieve the same delay. An increase in input size by 20% allows for a 22.3% reduction in energy. Further increases in input size yield savings at a diminishing rate.

4.3.2. Energy minimization of multi-path circuits

When optimizing circuits, the optimal solution occurs when the delay of each path from input to output is equalized [13, 14]. When analyzing the energy

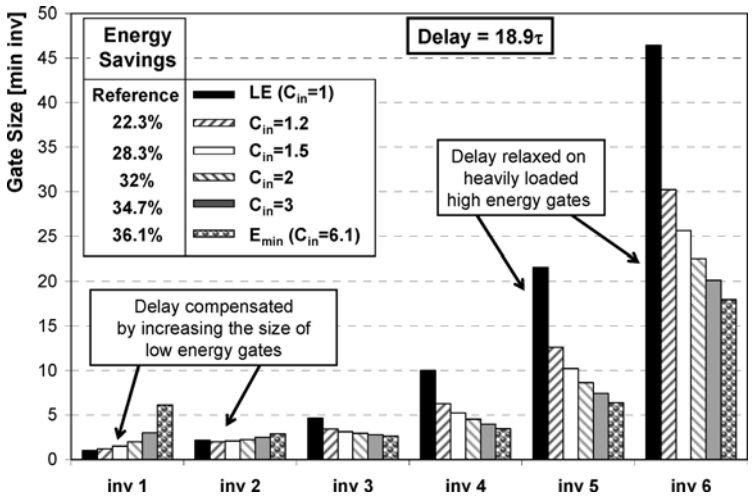

Figure 16. Gate Sizing of an Inverter Chain for Energy Reduction at a fixed Delay.

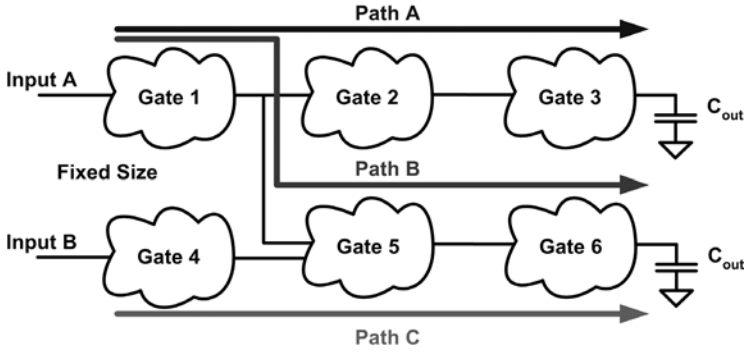

Figure 17. Multi-Path Circuit.

of a circuit, it is necessary to know the sizes of each gate in the circuit (not just those on the critical path). This further complicates the optimization process, as seen in the example of Figure 17. Paths A and B must now be optimized to include the constraint of having equal delay to that of Path C.

The exact solution to this problem requires a numerical approach such as convex optimization [24], from which we can obtain little to no intuition. In [21] we presented a simplified approach to analyze the energy-delay characteristics of an entire circuit. In this approach each gate is assigned to a logic stage, with every gate in the logic stage sized to have the same delay. Gates are assigned to stages starting from the input and moving towards the output. If a path has fewer stages than another path, the last gate of the path is sized to include the

combined delay of the additional stages of the longest path. Using this approach the delay of each path in the circuit is always equal, allowing for optimization to be performed at the stage level. The optimization has only a few variables (equal to the number of stages) and can be performed using widely available optimizers such as Matlab and Microsoft® Excel's Solver.

5. Designing Energy-Efficient Digital Circuits

Designing energy-efficient digital circuits requires different guidelines than those developed from Logical Effort. In LE a chain of inverters optimized for delay is used to demonstrate the relative insensitivity of design implementation to number of stages (Figure 18). While delay is relatively insensitive around the optimal number of stages, energy is very sensitive to the number of stages. Inverter chains which contain more stages than delay optimal are always sub-optimal in terms of energy. This result is contrary to LE, and requires that the number of stages be carefully analyzed to obtain an energy-efficient design.

In Figure 19 the minimal energy-delay curves of several inverter chains are shown, each with the same output load and input size. It is observed that the five and six stage designs are never energy-efficient, while the two, three and four stage designs have regions of energy-efficiency depending on the desired operating target.

Contrary to delay based optimization, the location of gates within a chain impacts the energy characteristics of a circuit. For example, the two chains

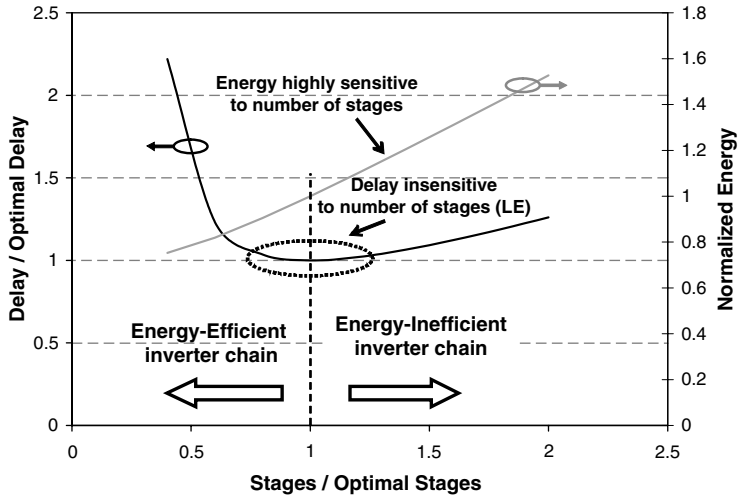

Figure 18. Optimal Number of Stages for an Inverter Chain.

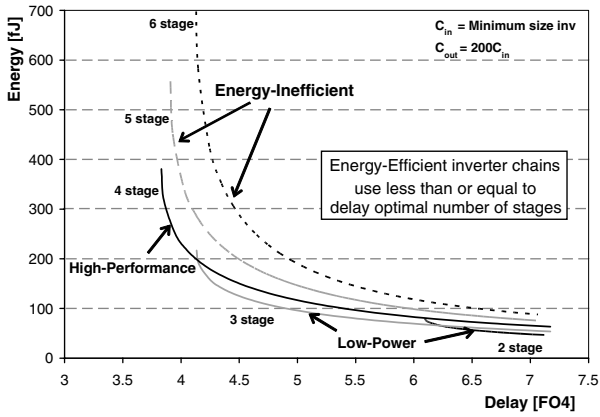

Figure 19. Optimal Number of Inverters for Fixed Output Load and Fixed Input Size with Varying Delay Target.

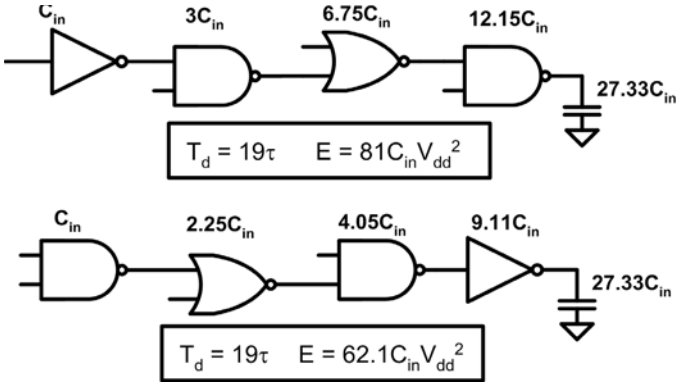

Figure 20. Energy Impact of Gate Placement in a Chain of Gates.

of gates in Figure 20 consist of the same gates, output load and input size, which results in the two paths having the same delay. However, the relative energy of each chain differs, from $81C_{\text{in}}V_{\text{dd}}^2$ to $62.1C_{\text{in}}V_{\text{dd}}^2$. Simpler gates, i.e. those with smaller g and p such as the inverter in the example, require less energy to drive a load than more complex gates. This is because for the same delay they present a smaller input capacitance to the previous gate and have less parasitic capacitance compared to more complex gates. Thus, simple gates should be placed in the most energy sensitive logic stage of a circuit whenever possible.

The arrangement of circuits also has implications on the optimal number of stages. For example, if we examine the impact of adding inverters to the

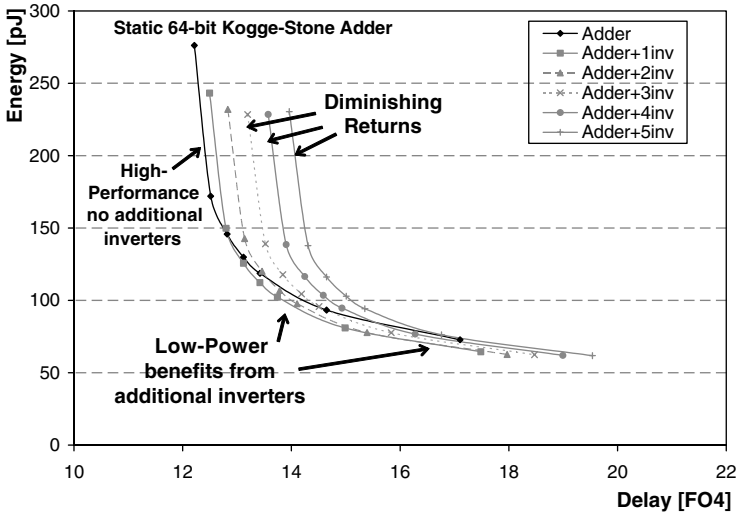

Figure 21. Impact of Buffers Insertion at the Output of a 64-bit KS Adder.

output of a 64-bit static KS adder as in Figure 21. Energy savings are obtained if up to 3 inverters are added at the output, although each occurs at a degraded performance target. Despite having more stages than delay optimal, the energy still decreases. This is because by adding simpler gates to the output, the size of the complex gates in the adder can decrease dramatically (similar to the example in Figure 20). Therefore, despite paying a slight delay penalty due to an extra logic stage, the energy of the design is decreased.

6. Conclusion

The design of digital circuits in current and future technologies requires an understanding of the energy-delay space. Design principles developed for optimizing delay, such as Logical Effort, no longer guarantee efficient designs when energy is considered. We have demonstrated that an energy model can be used in conjunction with standard RC-models to evaluate the energy-delay characteristics of a circuit. The analysis leads to the realization that ED^x metrics can not be used when designing a circuit for a fixed delay or energy. Instead circuits should be optimized for minimal energy at a fixed delay for a variety of system constraints. Using this approach a potential 30–50% energy savings can be achieved for circuits with no performance penalty compared to delay optimized results.

Acknowledgements

The authors would like to thank Hoang Dao and Milena Vratonjic for their comments and suggestions.

References

[1] Taur, Y. "CMOS design near the limit of scaling," *IBM Journal of Research and Development*, **2002**, *46*(2/3).
[2] International Technology Roadmap for Semiconductors, public.itrs.net.
[3] Meyerson, B. "How does one define "Technology" Now That Classical Scaling is Dead?", Keynote presentation, 42nd annual Design Automation Conference, Anaheim, CA, June **2005**.
[4] Sutherland, I.E.; Sproull, R. F. "Logical Effort: Designing for Speed on the Back of an Envelope," Advanced Research in VLSI, Proceedings of the 1991 University of California, Santa Cruz, Conference, Sequin, C.H. ed., MIT Press, **1991**, 1–16.
[5] Sutherland, I.E.; Sproull, R.F.; Harris, D. *Logical Effort Designing Fast CMOS Circuits*, Morgan Kaufmann Pub., **1999**.
[6] Horowitz, M. "Timing Models for MOS Circuits," PhD Thesis, Stanford University, December **1983**.
[7] Rubenstein, J.; Penfield, P.; Horowitz, M. A. "Signal Delay in RC Networks," *IEEE Transactions on Computer Aided Design*, **1983**, *Cad-2*(3), 202–211.
[8] Hodges, D.; Jackson, H. *Analysis and Design of Digital Integrated Circuits*, McGraw Hill, **1988**.
[9] Sakurai, T.; Newton, A. R. "Alpha-Power Law MOSFET Model and Its Application to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid-State Circuits*, **1990**, *25*(2), 584–594.
[10] Weste, N.; Eshraghian, K. *Principles of CMOS VLSI Design A Systems Perspective*, Addison Wesley, **1992**.
[11] Oklobdzija, V. G.; Barnes, E. R. "On Implementing Addition in VLSI Technology," *IEEE Journal of Parallel and Distributed Computing*, **1988**, *5*, 716–728.
[12] Fishburn, P.; Dunlop, A.E. "TILOS: A Posynomial Programming Approach to Transistor Sizing," *International Conference on Computer Aided Design*, November **1985**, 326–328.
[13] Sundararajan, V.; Sapatnekar, S. S.; Parhi, K. K. "Fast and Exact Transistor Sizing Based on Iterative Relaxation," *IEEE Transactions on Computer Aided Design of Circuits and Systems*, **2002**, *21*(5), 568–581.
[14] Sapatnekar, S. *Timing*, Kluwer Academic Publishers, Boston, MA, **2004**.
[15] Oklobdzija, V. G.; Zeydel, B. R.; Dao, H. Q.; Mathew, S.; Krishnamurthy, R. "Comparison of High-Performance VLSI Adders in Energy-Delay Space", *IEEE Transaction on VLSI Systems*, **2005**, *13*(6), 754–758.
[16] Zyuban, V.; Strenski, P. "Unified Methodology for Resolving Power-Performance Tradeoffs at the Micro-achitectural and Circuit Levels", *IEEE Symposium on Low Power Electronics and Design*, **2002**.
[17] Zyuban V.; Strenski, P. "Balancing Hardware Intensity in Microprocessor Pipelines," *IBM Journal of Research and Development*, **2003**, *47*(5/6).
[18] Stojanovic, V.; Markovic, D.; Nikolic, B.; Horowitz, M.A.; Brodersen, R.W. "Energy-Delay Tradeoffs in Combinational Logic using Gate Sizing and Supply Voltage

Optimization," Proceedings of the 28th European Solid-State Circuits Conference, ESSCIRC'2002, Florence, Italy, September 24–26, **2002**, 211–214.

[19] Markovic, D.; Stojanovic, V.; Nikolic, B.; Horowitz, M.A.; Brodersen, R.W. "Methods for True Energy-Performance Optimization," *IEEE J. Solid-State Circuits*, **2004**, *39*(8), 1282–1293.

[20] Hofstee, H. P. "Power-constrained microprocessor design," in Proc. Int. Conf. Computer Design, **2002**, 14–16

[21] Dao, H.Q.; Zeydel, B.R.; Oklobdzija, V.G. "Energy Minimization Method for Optimal Energy-Delay Extraction", European Solid-State Circuits Conference, Estoril, Portugal, September 16–18, **2003**.

[22] Dao, H. Q.; Zeydel, B. R.; Oklobdzija, V. G. "Energy Optimization of Pipelined Digital Systems Using Circuit Sizing and Supply Scaling," *IEEE Transaction on VLSI Systems*, **2006**, *14*(2), 122–134.

[23] Kogge, P.M.; Stone, H.S. "A parallel algorithm for the efficient solution of a general class of recurrence equations", *IEEE Trans. Computers*, August **1973**, *C-22*(8), 786–793.

[24] Boyd, S.; Vandenberghe, L. *Convex Optimization*, Cambridge University Press, **2004**.