Finding the Energy Efficient Curve: Gate Sizing for Minimum Power under Delay Constraints

Yoni Aizik and Avinoam Kolodny
yoni.aizik@intel.com, kolodny@ee.technion.ac.il
Technion, Israel Institute of Technology, Haifa, Israel

Abstract

A design scenario examined in this paper assumes that a circuit has been designed initially for high speed, and it is redesigned for low power by downsizing of the gates. Such a design flow is interesting because design methods had been traditionally focused on performance, hence deeply rooted engineering practices tend to overemphasize circuit speed at the cost of excessive power dissipation. In recent years, as power consumption has become a dominant issue, new optimizations of circuits are required for saving energy. This is done by trading off some speed in exchange for reduced power. For each feasible speed, an optimization problem is solved in this paper, finding new sizes for the gates such that the circuit satisfies the speed goal while dissipating minimal power. Since both dynamic and leakage energy depend linearly on the gates' sizes, downsizing of the gates decreases both dynamic and leakage energy dissipation. Energy/delay gain (EDG) is defined as a metric to quantify the most efficient tradeoff. The EDG of the circuit is evaluated for a range of reduced circuit speeds, and the power-optimal gate sizes are compared with the initial sizes. The power reduction process is applied to several typical circuits in 32nm technology, and power reduction of up to 25% for delay increase of 5% (EDG=5) is demonstrated. Most of the energy savings occur at the final stages of the circuits, while the largest relative downsizing occurs in middle stages. Typical tapering factors for power efficient circuits are larger than for speed-optimal circuits. Signal activity and signal probability affect the optimal gate sizes in the combined optimization of speed and power.

Key words:
Power Performance Tradeoff, Sizing, Energy Delay Gain, EDG, Hardware

1. Introduction

Optimizing a digital circuit for both energy and performance involves a tradeoff, because any implementation of a given algorithm consumes more energy if it is executed faster. The tradeoff between power and speed is influenced by the circuit structure, the logic function, the manufacturing process, and other factors. Traditional design practices tend to overemphasize speed and waste power. In recent years power has become a dominant consideration, causing designers to downsize logic gates in order to reduce power, in exchange for increased delay. However, resizing of gates to save power is often performed in a non-optimal way, such that for the same energy dissipation, a sizing that results in better performance could be achieved.

In this paper, we explore the energy-performance design space, evaluating the optimal tradeoff between performance and energy by tuning gate sizes in a given circuit. We describe a mathematical method that minimizes the total energy in a combinational CMOS circuit, for a given delay constraint. It is based on an extension of the Logical Effort [8] model to express the dynamic and leakage energy of a path as well as the delay. Starting from the minimum achievable delay, we apply the method for a range of longer delays, in order to find the optimal energy-delay relation for the given circuit. We show that downsizing all gates in a fast circuit by the same factor does not yield an energy-efficient design, and we characterize the differences between gate sizing for high speed and sizing for low power.

In trading off delay for energy, we are interested only in a subset of all the possible downsized circuits - those implementations that are energy efficient. A design implementation is considered to be energy efficient when it has the highest performance among all possible configurations dissipating the same power ([13, 1]). When the optimal implementations are plotted in the energy-delay plane, they form a curve called the *energy efficient curve*. In Figure 1, each point represents a different hardware implementation. The implementations which belong to the energy efficient family reside on the energy efficient curve.

Zyuban and Strenski ([1, 2]) introduce the *hardware intensity* metric. Hardware intensity (η) is defined to be the ratio of the relative increase in energy to the corresponding relative gain in performance achievable **locally**

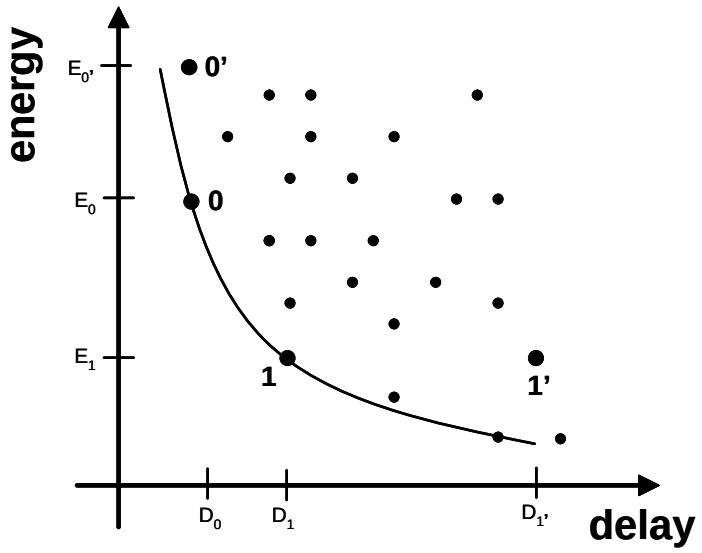


Figure 1: **Energy Efficient Curve**. Although implementations 0 and 0' of the given circuit have the same delay (D_0), implementation 0 consumes less energy. Similarly, implementations 1 and 1' consume the same energy, but implementation 1 has a shorter delay (D_1), hence is preferable. Points 0 and 1 are on the energy efficient curve. All implementations have the same circuit topology, with different device sizes.

through gate resizing and logic manipulation at a fixed power-supply voltage for a power efficient design. Simply put, it is the ratio of % energy per % speed performance tradeoff for an energy-efficient design. Since speed performance is inversely proportional to delay,

$$\eta = -\frac{\frac{1}{E}}{\frac{1}{D}} \frac{\partial E}{\partial D} \tag{1}$$

where D is delay, E is the dissipated energy, and η represents the hardware intensity. The hardware intensity is a measure of the **differential** energy-performance tradeoff (the energy gained if the delay is relaxed by a small ΔD around a given delay and energy point on the energy efficient curve), and is actually the sensitivity of the energy to the delay.

As shown in [1] , each point on the energy efficient curve corresponds to a different value of the hardware intensity η. The hardware intensity decreases along the energy efficient curve towards larger delay values. According to [1],

η is equivalent to the tradeoff parameter n in the commonly used optimization objective function combining energy and delay

$$F_{opt} = E \cdot D^n, n \geqslant 0 \tag{2}$$

In [14], Brodersen et. al. formalize the tradeoff between energy and delay via sensitivities to tuning parameters. The sensitivity of energy to delay due to tuning the size W_i of gate i is defined as:

$$\theta(W_i) = -\frac{\frac{1}{E}}{\frac{1}{D}} \cdot \frac{\partial E/\partial W_i}{\partial D/\partial W_i} \tag{3}$$

where $\theta(W_i)$ is the sensitivity, D is the delay, E is the energy, $\partial E/\partial W_i$ is the derivative of energy with respect to size of device i, and $\partial D/\partial W_i$ is the derivative of delay with respect to size of device i. To achieve the most energy-efficient design, the energy reduction potentials of all the tuning variables must be the same. Therefore, for an energy efficient design, (3) is equivalent to (1) for all points on the energy efficient curve.

The focus of this paper is on the conversion to low power of circuits that were optimized only for speed during their initial design process. Optimal downsizing is applied to each gate for each relaxed delay target, such that the whole energy efficient curve is generated for the circuit. Note that the gate sizes are allowed to vary in a continuous manner between a minimum and a maximum size. While the resultant gate sizes would be mapped into a finite cell library in a practical design, the continuous result for some basic circuits provide guidelines and observations about CMOS circuit design for low power.

The rest of this paper is organized as follows: The design scenario is described in Section 2. Usage of logical effort to analyze the delay and energy is described in Section 3. The optimization problem is formalized in Section 4. Typical circuit types are analyzed in Section 5. Section 6 concludes the paper.

2. Power Reduction Design Scenario

Typically, an initial circuit is given, where speed was the only design goal. In order to save energy, the delay constraint is relaxed, and the gates sizes are reduced. For example, consider Figure 1, with the initial circuit implementation 0, which is energy efficient. While relaxing the delay constraint

(moving from D_0 to D_1), the design gets downsized, which results in circuit implementation 1.

To calculate the energy gain achievable by relaxing the delay by X percent, we define a metric we call "Energy Delay Gain" (EDG). The EDG is defined as the ratio of relative decrease in energy to the corresponding relative increase in delay, w.r.t. the initial design point (D_0, E_0). D_0 is the initial delay (not necessarily the minimum achievable delay), and E_0 is the corresponding initial energy. Note that the EDG defines the total energy-performance tradeoff, as opposed to the differential tradeoff - the hardware intensity. Mathematically, EDG at a given delay D with corresponding energy E is defined as

$$EDG = \frac{(E_0 - E)/E_0}{(D - D_0)/D_0}. \tag{4}$$

For example, assuming the initial design point in Figure 1 is implementation 0, then the EDG of point 1 is

$$\frac{(E_0 - E_1)/E_0}{(D_1 - D_0)/D_0}.$$

Figure 2 illustrates the difference between hardware intensity and EDG. It shows the energy efficient curve of a given circuit, where D_0 is the initial delay, and E_0 is the corresponding initial energy. The hardware intensity is the ratio between the slope of the tangent to the energy efficient curve at point (D, E), to the slope of the line connecting the origin to point (D, E). The EDG is the ratio between the slope of the line connecting points (D_0, E_0) and (D, E), to the slope of the line connecting the origin to point (D_0, E_0). Note that when point (D, E) is close to (D_0, E_0), the two definitions converge.

Re-sizing of the gates to tradeoff performance with active energy is the most practical approach available to the circuit engineer. Continuous gate sizes has been used for optimizing delay under area constraints and vice versa ([25]). Other degrees of freedom include logic restructuring, tuning of threshold voltages or supply voltage, and power gating. Changing the threshold voltage affects mainly the leakage energy, and not the dynamic energy dissipation ([3, 23]). So does power gating ([6, 7]). Logic restructuring of the circuit could be an effective method to trade off energy and performance, by reducing the load on high activity nets, and by introducing new nodes that have a lower switching activity ([17]). However, changing the circuit topology

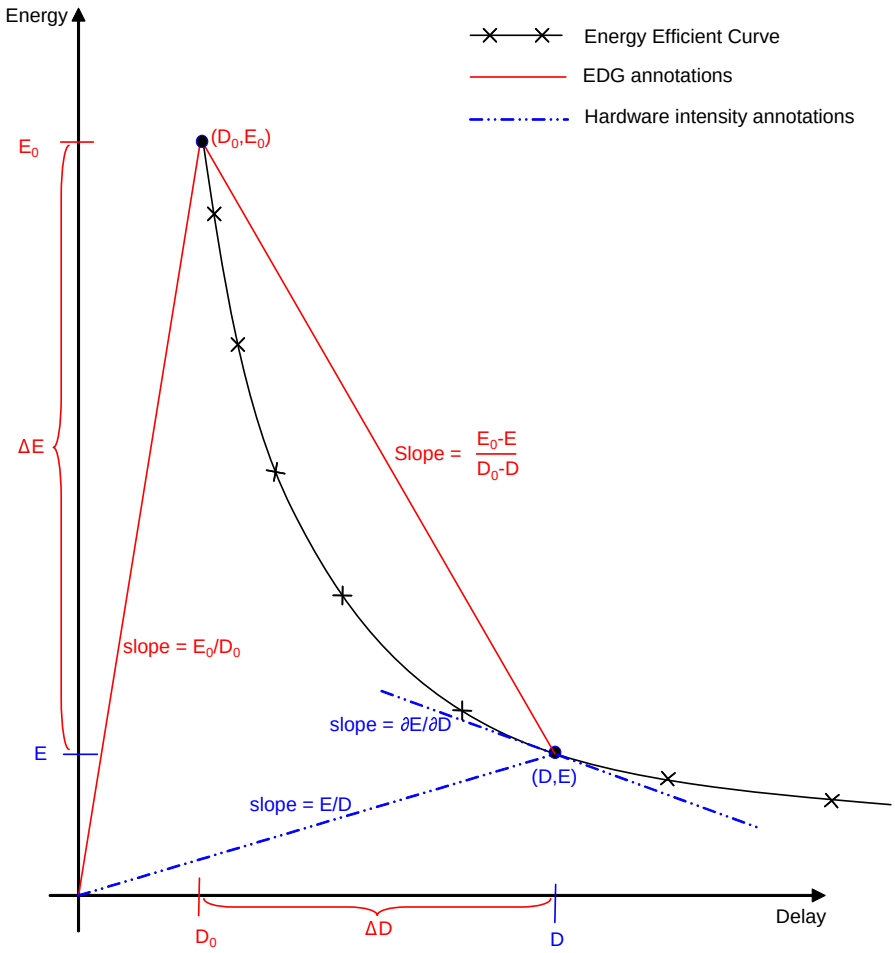


Figure 2: **EDG and Hardware Intensity**. Note that when $(D, E) \to (D_0, E_0)$, Hardware intensity and EDG converge.

may increase the time required for the design process to converge. Changing the supply voltage is an effective technique as well ([14, 17, 1, 12, 3, 4, 5]). However, in most cases, changing the supply voltage for a sub-circuit requires major changes in the package and in the system, and therefore is not practical. For instance, latest state of the art CPUs include only 1-2 power planes ([18, 19]).

In the following sections, we set up an optimization framework that maximizes the energy saving for any assumed delay constraint in a given com-

binational CMOS circuit. It determines the appropriate sizing factor for each gate in the circuit. For primary inputs and outputs of the circuit we assume fixed capacitances. Given activity factor and signal probability are assumed at each node of the circuit. The result of this optimization process is equivalent to finding the energy-efficient curve for the given circuit.

3. Analytical Model

The optimization problem we solve is defined as follows: given a path in a circuit with initial delay (minimum or arbitrary) D_0 and the corresponding energy consumption E_0, find gate sizing that maximizes the EDG for an assumed delay constraint. We use the logical effort method ([8]) in order to calculate the delay of a path, and adapt it to calculate the dynamic and leakage energy dissipation of the circuit.

For a given path (Figure 3), we assume constant input and output loads, and an initial sizing that is given as input capacitance for each gate. For each gate we apply a sizing factor k. The input capacitance of the resized i^{th} gate is expressed as the initial input capacitance C_{0_i} multiplied by k_i. The energy-delay design space is explored by tuning the k's.

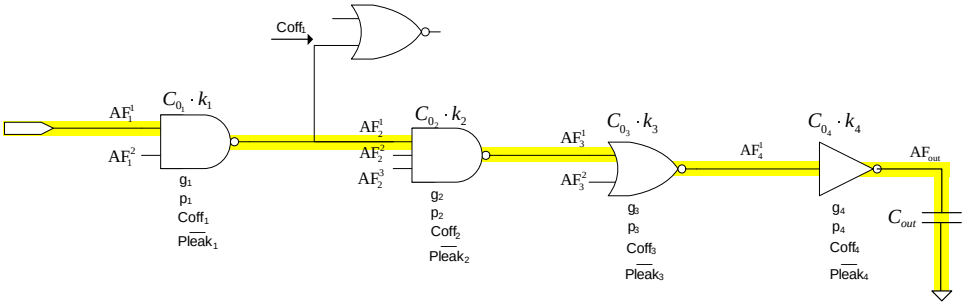


Figure 3: **Example path**. Each gate is assigned with logical effort notation, initial input capacitance (C_{0_i}) and sizing factor (k_i)

The following properties are defined:

M_i - Number of inputs to gate i
AF_i^j - Activity factor (switching probability) of input j in gate i

AF_o^i - Output activity factor of gate i
g_i - Logical effort of gate i
p_i - Parasitic delay of gate i
C_{0_i} - Initial capacitance of gate i that achieves initial path delay (corresponds to (D_0, E_0))
C_{off_i} - Off-path constant capacitance driven by gate i
P_{leak_i} - the average leakage power for gate i, for a unit input capacitance
k_i - Sizing factor for gate i. The k's are used in the gate downsizing process. For each gate i, $k_i \cdot C_{0_i}$ is the gate size. Although specified, k_1 is assumed to be 1 (constant driver).

3.1. Energy of a Logic Path

3.1.1. Switching Energy

The switching energy of a static CMOS gate i with M_i inputs and a single output is

$$\text{Switching Energy} = \underbrace{\sum_{j=1}^{M_i} AF_i^j \cdot C_j \cdot V_j^2}_{\text{input energy}} + \underbrace{AF_{out_i} \cdot C_{out_i} \cdot V_{out_i}^2}_{\text{output energy}} \tag{5}$$

Assuming the voltage amplitude for each net in the design is the same (Vcc), we can define a parameter called dynamic capacitance (C_{dyn}), which is the switching energy normalized by Vcc. The dynamic capacitance of a gate i (C_{dyn_i}), is -

$$C_{dyn_i} = \frac{\text{Switching Energy}}{Vcc^2} = \sum_{j=1}^{M_i} AF_i^j \cdot C_j + AF_{out_i} \cdot C_{out_i} \tag{6}$$

Without loss of generality, we assume that the first input of each gate resides on the investigated path. We assume that the inputs of the gates we deal with are symmetrical (input capacitance on each input pin is equal) and the gates are non-compound (i.e., gates implementing functions like $\overline{a \cdot b + c}$ are out of scope). Our method can be easily extended to support these types. Under these assumptions, all input capacitances of a given gate are identical.

Therefore, the input C_{dyn} of gate i ($C_{dyn_{in}}i$) is :

$$\begin{aligned} C_{dyn_{in}}i &= C_{0_i} \cdot k_i \sum_{j=1}^{M_i} AF_i^j \\ &= C_{0_i} \cdot k_i \cdot AF_i \end{aligned} \tag{7}$$

Where AF_i is defined to be $\sum_{j=1}^{M_i} AF_i^j$ - sum of activity factors for input pins of gate i. Note that unlike calculating the delay of a gate, when calculating the gate energy, all input and output nets of a gate have to be taken into consideration. The C_{dyn} of the nets not in the desired path should not be overlooked.

The output capacitance of a gate is defined to be its self loading, and is combined mainly of the drain diffusion capacitors connected to the output. The parasitic delay of gate i in logical effort method, denoted by p_i, is proportional to the diffusion capacitance. The logical effort of gate i, denoted by g_i, expresses the ratio of the input capacitance of gate i to that of an inverter capable of delivering the same current. It is easy to see that the output capacitance of gate i can be expressed as

$$C_{out_i} = \frac{C_{in_i}}{g_i} p_i \tag{8}$$

We can now re-write (6) using the notation defined above:

$$C_{dyn_i} = C_{0_i} k_i \cdot AF_i + \frac{C_{0_i} k_i}{g_i} p_i \cdot AF_o^i \tag{9}$$

Besides the gates in the path, we have to take into account the C_{dyn} of the side loads. Multiplying Coff_i by AF_i^1 results in the C_{dyn} of the off-path load driven by gate i. We use (9) to calculate C_{dyn} of a desired path -

$$\begin{aligned} C_{dyn} &= \underbrace{AF_1 \cdot Cin_1}_{\text{input } C_{dyn}} + \sum_{i=2}^{N} \underbrace{AF_i \cdot Cin_i + AF_o^{i-1} \cdot Cout_{i-1}}_{\text{stage } i \ C_{dyn}} \\ &+ \underbrace{AF_o^N (Cout_N + C_{load})}_{\text{output } C_{dyn}} + \sum_{i=1}^{N} \underbrace{\mathrm{Coff}_i \cdot AF_i^1}_{C_{dyn} \text{ of off path load i}} \end{aligned} \tag{10}$$

Substituting input C_{dyn} with (7) and $Cout_i$ with (8), and rearranging the formula, we get:

$$\begin{aligned} C_{dyn} &= \sum_{i=1}^{N} k_i(AF_i \cdot C_{0_i} + AF_o^i \cdot \frac{C_{0_i} \cdot p_i}{g_i}) + AF_{N+1} \cdot C_{load} \\ &+ \sum_{i=1}^{N} \text{Coff}_i \cdot AF_i^1 \end{aligned} \tag{11}$$

By defining

$$\begin{aligned} C_{dyn_i} &\triangleq AF_i \cdot C_{0_i} + AF_o^i \cdot \frac{C_{0_i} \cdot p_i}{g_i} \\ C_{dyn-off} &\triangleq \sum_{i=1}^{N} \text{Coff}_i \cdot AF_i^1 \end{aligned} \tag{12}$$

We get

$$C_{dyn} = \sum_{i=1}^{N} C_{dyn_i} \cdot k_i + AF_{N+1} \cdot C_{load} + C_{dyn-off} \tag{13}$$

The initial C_{dyn} is achieved by setting all $k_i's$ to 1 -

$$C_{dyn}^0 \triangleq C_{dyn} \mid_{k_i's=1} = \sum_{i=1}^{N} C_{dyn_i} + AF_{N+1} \cdot C_{load} + C_{dyn-off} \tag{14}$$

3.1.2. Leakage Energy

The leakage energy of a static CMOS gate i with M_i inputs and a single output can be expressed as

$$\text{Leakage Energy of Gate i} = T_{cycle} \cdot \bar{P_{leak_i}} \cdot C_{0_i} \tag{15}$$

Where T_{cycle} is the cycle time of the circuit, and $\bar{P_{leak_i}}$ is the average leakage power for gate i, for a unit input capacitance. $\bar{P_{leak_i}}$ is a function of the manufacturing technology, gate topology, and signal probability (SP - the probability for a signal to be in a logical TRUE state at a given cycle) for each input. See [22, 23, 24] for leakage power calculation methods. Under a given workload, $\bar{P_{leak_i}}$ should be pre-calculated for each gate i. Since $\bar{P_{leak_i}}$ is sensitive to the signal probability, it needs to be re-calculated whenever the workload is modified, to reflect changes in gates' signal probability

By dividing the leakage energy by V_{cc}^2, we can express the leakage in terms of capacitance -

$$\text{Leakage Capacitance of Gate i} \triangleq C_{leak_i} = \frac{1}{V_{cc}^2} T_{cycle} \cdot C_{0_i} \cdot \bar{P_{leak_i}} \tag{16}$$

And the total C_{leak} is equal to -

$$\begin{aligned} C_{leak} &= \frac{1}{V_{cc}^2} T_{cycle} (\sum_{i=1}^{N} \left(k_i \cdot C_{0_i} \cdot \bar{P_{leak_i}}\right) \\ &= \sum_{i=1}^{N} k_i \cdot C_{leak_i} \end{aligned} \tag{17}$$

The initial C_{leak} is achieved by setting all $k_i's$ to 1 -

$$C_{leak}^0 \triangleq C_{leak} \mid_{k_i's=1} = \sum_{i=1}^{N} C_{leak_i} \tag{18}$$

By combining (13, 14, 17, 18) we can express the total capacitance and the initial capacitance of a desired path -

$$C_{path} = \sum_{i=1}^{N} k_i \left(C_{dyn_i} + C_{leak_i}\right) + AF_{N+1} \cdot C_{load} + C_{dyn-off} \tag{19}$$

$$C_{path}^0 = \sum_{i=1}^{N} \left(C_{dyn_i} + C_{leak_i}\right) + AF_{N+1} \cdot C_{load} + C_{dyn-off} \tag{20}$$

The energy decrease rate (e_{dec}) due to downsizing of the gates by a factor of k is expressed as

$$e_{dec} = \frac{C_{path}^0 - C_{path}}{C_{path}^0} = \frac{\sum_{i=1}^{N} (C_{dyn_i} + C_{leak_i})(1 - k_i)}{\sum_{i=1}^{N} \left(C_{dyn_i} + C_{leak_i}\right) + AF_{N+1} \cdot C_{load} + C_{dyn-off}} \tag{21}$$

In order to estimate the upper bound of e_{dec}, we assume an initial design point with minimum delay for C_{path}^0, and set the sizes of the gates in the path to minimum allowed feature size (C_{min}), to reflect the minimum possible C_{path}. By defining

$$\begin{aligned} C_{dyn_i}^{\min} &\triangleq AF_i \cdot C_{min} + AF_o^i \cdot \frac{C_{min} \cdot p_i}{g_i} \\ C_{leak_i}^{\min} &\triangleq \frac{1}{V_{cc}^2} T_{cycle} \cdot C_{min} \cdot \bar{P_{leak_i}} \end{aligned} \tag{22}$$

We get

$$e_{dec} \leqslant e_{dec}^{\mathrm{MAX}} = \frac{\sum_{i=1}^{N}\left(C_{dyn_i}+C_{leak_i}\right)-\sum_{i=1}^{N}\left(C_{dyn_i}^{\min}+C_{leak_i}^{\min}\right)}{\sum_{i=1}^{N}(C_{dyn_i}+C_{leak_i})+AF_{N+1}\cdot C_{load}+C_{dyn-off}} \tag{23}$$

By using 23, the upper bound to the EDG at a given delay increase rate (d_{inc}) - $EDG_{(d_{inc})}^{\mathrm{MAX}}$ can also be calculated, simply by dividing e_{dec}^{MAX} by d_{inc} -

$$EDG_{(d_{inc})}^{\mathrm{MAX}} = e_{dec}^{\mathrm{MAX}}/d_{inc} \tag{24}$$

$EDG_{(d_{inc})}^{\mathrm{MAX}}$ can be used by the circuit designer to quickly evaluate the potential for saving power. However, the designer should note that the value of $EDG_{(d_{inc})}^{\mathrm{MAX}}$ is a non-reachable upper bound since the minimum sizing leads to a delay increase which is always greater than the one that the designer refers to. If the value of $EDG_{(d_{inc})}^{\mathrm{MAX}}$ is not sufficient, other energy reduction techniques should be considered.

3.2. Delay of a Logic Path

When using the logical effort notation, the path delay (D) is expressed as

$$D = \sum_{i=1}^{N} g_i h_i + P \tag{25}$$

The electrical effort of stage i (h_i) is calculated as the ratio between capacitance of gate $i+1$ and gate i, plus the ratio of side load capacitance of gate i and input capacitance of gate i. For the sake of simplicity, k_{N+1} and k_1 are defined to be 1. Using the notation defined earlier, the path delay D can be written as -

$$D = \sum_{i=1}^{N} g_i \left(\frac{C_{0_{i+1}} k_{i+1}}{C_{0_i} k_i} + \frac{\mathrm{Coff}_i}{C_{0_i} k_i}\right) + P \tag{26}$$

By defining

$$\begin{aligned} D_i^0 &\triangleq g_i \frac{C_{0_{i+1}}}{C_{0_i}} \\ D_i^1 &\triangleq g_i \frac{\mathrm{Coff}_i}{C_{0_i}} \end{aligned} \tag{27}$$

(26) becomes:

$$D = \sum_{i=1}^{N} \left(D_i^0 \frac{k_{i+1}}{k_i} + D_i^1 \frac{1}{k_i} \right) + P \tag{28}$$

The initial delay is achieved by setting all $k_i's$ to 1.

$$D_0 \triangleq D \mid_{k_i's=1} = \sum_{i=1}^{N} \left(D_i^0 + D_i^1 \right) + P \tag{29}$$

And therefore, the delay increase rate (d_{inc}) due to downsizing of the gates by a factor of k_i is

$$d_{inc} = \frac{D - D_0}{D_0} = \frac{\sum_{i=1}^{N} \left(D_i^0 \frac{k_{i+1}}{k_i} + D_i^1 \frac{1}{k_i} \right) + P - D_0}{D_0} \tag{30}$$

4. Optimizing Power and Performance

Given a delay value that is d_{inc} percent greater than the initial delay D_0, we seek the path sizing $(C_{0_2} \cdot k_2 \cdots C_{0_N} \cdot k_N)$ that maximizes the energy reduction rate e_{dec}.
From (21), maximizing e_{dec} is achieved by minimizing C_{dyn}. By ignoring the factors that do not depend on k_i and will not affect the optimization process in (19), we define an objective function f_0 -

$$f_0 = \sum_{i=1}^{N} k_i \left(C_{dyn_i} + C_{leak_i} \right) \tag{31}$$

Note that f_0 depends linearly on the dynamic and the leakage capacitances, which apply weights and determine the importance of each k_i. (31) can also be written as -

$$f_0 = \sum_{i=1}^{N} k_i C_{0_i} \left(\frac{1}{V_{cc}^2} T_{cycle} \bar{P}_{leak_i} + AF_i + AF_o^i \cdot \frac{p_i}{g_i} \right) \tag{32}$$

Note that when all gates in a path are of the same type, all activity factors are equal, and average leakage power for all gates in the path is equal, both C_{dyn_i}

and C_{leak_i} can be eliminated from (31) without affecting the optimization result. These conditions are satisfied on an inverter chain with input signal probability of 0.5, for instance. In this case, the leakage power of activity factor has no influence on the optimization result.
To get a canonical constraint goal, in which the constraint $\leqslant 1$, we re-arrange (30) to

$$\sum_{i=1}^{N}\left(D_i^0\frac{k_{i+1}}{k_i}+D_i^1\frac{1}{k_i}\right)=d_{inc}D_0+D_0-P \tag{33}$$

and define

$$\begin{aligned} D_i'^0 &\triangleq \frac{D_i^0}{d_{inc}D_0+D_0-P} \\ D_i'^1 &\triangleq \frac{D_i^1}{d_{inc}D_0+D_0-P} \end{aligned} \tag{34}$$

to get

$$\sum_{i=1}^{N}\left(D_i'^0\frac{k_{i+1}}{k_i}+D_i'^1\frac{1}{k_i}\right)=1 \tag{35}$$

We now can use (35) to get an optimization constraint -

$$f_1=\sum_{i=1}^{N}\left(D_i'^0\frac{k_{i+1}}{k_i}+D_i'^1\frac{1}{k_i}\right)\leqslant 1 \tag{36}$$

Combining (36) and (31) results in the following optimization problem -

Minimize $f_0(k_1\cdots k_N)$, subject to $f_1(k_1\cdots k_N)\leqslant 1$,where

$$\begin{aligned} f_0(k_1\cdots k_N) &= \sum_{i=1}^{N}k_i\left(C_{dyn_i}+Cleak_i\right) \\ f_1(k_1\cdots k_N) &= \sum_{i=1}^{N}\left(D_i'^0\frac{k_{i+1}}{k_i}+D_i'^1\frac{1}{k_i}\right) \end{aligned} \tag{37}$$

However, f_1 defined above is non-convex. We use geometrical program-

ming [9, 10, 11] to solve the optimization problem, by changing variables

$$\begin{aligned}
\widetilde{k_i} = log(k_i) &\Rightarrow k_i = e^{\widetilde{k_i}} \\
\widetilde{C_{dyn_i}} = log(C_{dyn_i}) &\Rightarrow C_{leak_i} = e^{\widetilde{C_{leak_i}}} \\
\widetilde{C_{leak_i}} = log(C_{leak_i}) &\Rightarrow C_{dyn_i} = e^{\widetilde{C_{dyn_i}}} \\
\widetilde{D_i'^0} = log\left(D_i'^0\right) &\Rightarrow D_i'^0 = e^{\widetilde{D_i'^0}} \\
\widetilde{D_i'^1} = log\left(D_i'^1\right) &\Rightarrow D_i'^1 = e^{\widetilde{D_i'^1}}
\end{aligned} \tag{38}$$

So the equivalent convex optimization problem (which can be solved using convex optimization tools) is -

Minimize $\widetilde{f}_0(k_1 \cdots k_N)$, subject to $\widetilde{f}_1(k_1 \cdots k_N) \leqslant 0$,where

$$\begin{aligned}
\widetilde{f}_0(k_1 \cdots k_N) &= log\left(\sum_{i=1}^{N} e^{\widetilde{k_i}+\widetilde{C_{dyn_i}}} + e^{\widetilde{k_i}+\widetilde{C_{leak_i}}}\right) \\
\widetilde{f}_1(k_1 \cdots k_N) &= log\left(\sum_{i=1}^{N} e^{\widetilde{k}_{i+1}-\widetilde{k_i}+\widetilde{D_i'^0}} + e^{\widetilde{D_i'^1}-\widetilde{k_i}}\right)
\end{aligned} \tag{39}$$

The convexity of (39) ensures that a solution to the optimization problem exists, and that the solution is the global optimum point. In order to obtain the EDG curve, the delay increase rate is swept from 0 to the desired value, and for each delay increase value, a different optimization problem is solved by geometrical programming.

This result can be extended to handle circuit delay, instead of a single path delay. All paths must be enumerated, and the optimized delay should reflect the critical path delay. The critical path delay is calculated as the maximum delay of all enumerated paths. However, the MAX operator cannot be handled directly in geometrical programming, since it produces a result which is not necessarily differentiable. Boyd et. al. ([10]) solve the general problem of using the MAX operator in geometrical programming ($MAX(f_1(x), f_2(x) \cdots f_N(x)) \leqslant 1$) by introducing a new variable t, and N

inequalities (N being the number of paths), to obtain

$$\begin{aligned} t &\leqslant 1 \\ f_1(x) &\leqslant t \\ f_2(x) &\leqslant t \\ \cdots & \\ f_N(x) &\leqslant t \end{aligned}$$

This transformation can be used in order to feed the critical path into the optimizer. To calculate the energy-delay tradeoff, the C_{dyn} of the entire circuit should be taken into account.

In the following sections, we employ this procedure to characterize the EDG and power reduction in typical logic circuits, and derive design guidelines.

5. Exploring Energy-Delay Tradeoff in Basic Circuits

We run numerical experiments that explore the EDG of some basic circuits. We use GGPLAB ([15]) as a geometrical programming optimizer, to solve the optimization problem (37, 39). GGLAB is a free open source library, and can be easily installed over Matlab. For each experiment, we provide an EDG curve which is obtained by optimizing the circuit for a wide range of increased delay values. Although the propagation delay and the active energy dissipation are technology independent, the leakage depends on the manufacturing technology and the circuit's cycle time. Throughout this section, the leakage is calculated according to the 32nm technology node of the ITRS 2007 projection [21], in which $C_{leak_{inv}}$ is calculated to be 0.5694, based on clock frequency of 2GHz and signal probability of 0.5.

5.1. Inverter Chain

Consider a chain consisting of N inverters, with output load of C_{out}. C_{0_1} is set arbitrarily to a constant value of 1 fF, and therefore the path electrical effort (H) is C_{out} (Figure 4). We set initial gate capacitances ($C_{0_2} \cdots C_{0_N}$) that ensure minimum delay, using the logical effort methodology. The minimum delay was obtained by setting the electrical effort to be the N^{th} root of the path electrical effort. The leakage calculation takes into account the signal probability of the inverters in the chain.

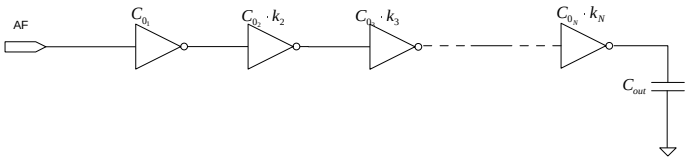


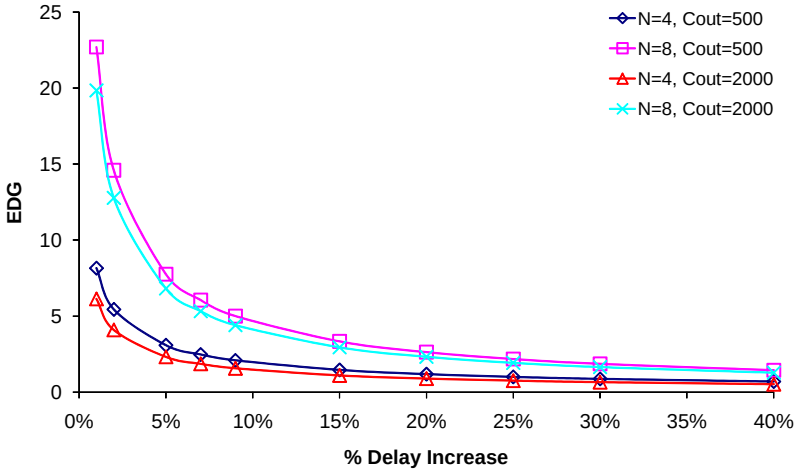
Figure 4: **Inverter Chain** - Consists of N stages, output load C_{out}, and initial capacitances ($C_{0_1} \cdots C_{0_N}$)

Figure 5(a) shows the EDG for different combinations of path electrical effort (H) and chain length (N) where the leakage energy is negligible. Figure 5(b) shows the same analysis, for negligible dynamic energy. In both cases, the largest potential for energy savings occurs near the point where the design is sized for minimum achievable delay. The potential for energy savings decreases as the delay is being relaxed further. This is consistent with the observation in [14].

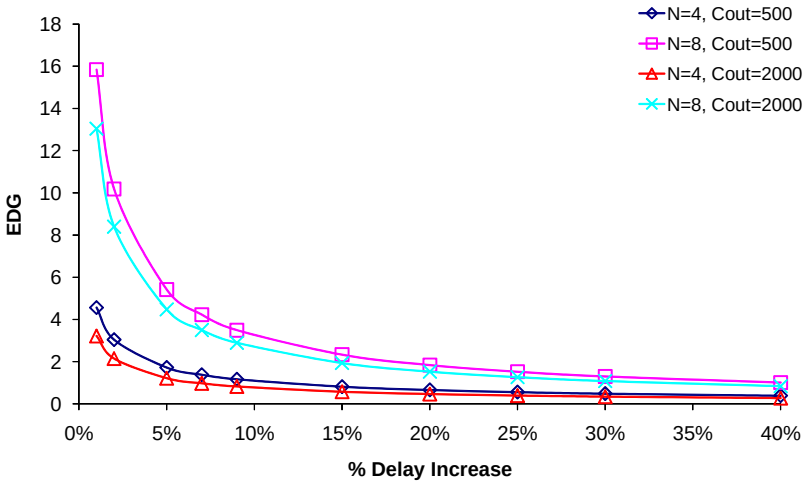
Figure 6 shows the optimal sizing of a fixed input and output load inverter chain with an arbitrary activity factor and signal probability of 0.5, for various delay increase values. For input signal probability of 0.5, all the gates in the inverter chain have the same signal probability. Therefore, the optimization process is indifferent to the average leakage power of each gate - P_{leak_i} in (16) is constant and can be eliminated from (37).

The optimization process leads to increasing the electrical effort of the last stages, and decreasing the electrical effort of the first stages, to meet the timing requirements (Figure 6(f)). The largest energy savings, for a given delay increase value, are achieved by downsizing the largest gates in the chain (6(e)). The relative downsizing, however, is maximal around the middle of the chain (6(c)), due to the fact that the first stage and the load are anchored with a fixed size. In order to understand the behavior of the middle stages, a 16-stage inverter stage simulation is plotted in Figure 6(d). As the delay increases, the gates towards the middle of the chain are downsized and form a plateau-like shape. Note that the optimal gate sizes might be limited by the minimum allowed size according to design rules.

Both Figures 6(a) and 6(b) (absolute sizing) and Figure 6(f) illustrate that as we move further from the minimal achievable delay (delay increase = 0, where all electrical efforts are identical), the difference between the



(a) Energy Delay Gain, Active Dominant Circuit



(b) Energy Delay Gain, Leakage Dominant Ciruit

Figure 5: **Inverter Chain** - various loads (C_{out}) and chain length (N)

electrical efforts of the stages increases. However, uniform downsizing (e.g. increase the delay by downsizing each gate by 5%) is sometimes used in the power reduction process by the circuit designer as an easy and straightforward method to trade off energy for performance. Figure 7 shows the energy

efficient curve (optimal sizing) vs. energy-delay curve generated by uniform downsizing of an 8-long inverter chain with out/in capacitance ratio of 200. The energy difference between the curves in the figure reaches up to 7%.

Most of the energy in the path is dissipated in the last stages of the chain, where the fanout factors are larger, in order to drive the large fixed output capacitance.

Figure 8 demonstrates the effect of chain length on delay and energy. The external load of the circuit is relatively large - 9pF, for which 8-long chain yields an optimal timing. The energy efficient curves for chains of 8, 6, and 4 inverters are plotted in the energy-delay plane. We can see that the number of stages is important when the optimal delay is required. Generally, as we move further from the smallest achievable delay, fewer inverters achieve better energy dissipation for the same delay. However, the difference in energy between the optimal number of inverters and a fixed number of inverters decreases as the delay is relaxed.

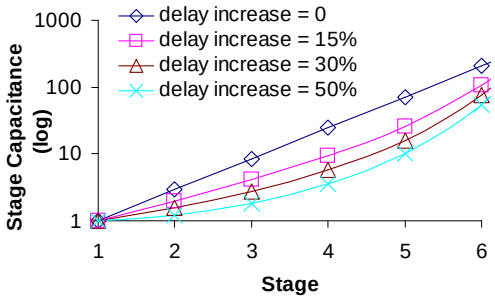
Figure 9 shows good correlation between $EDG_{10\%}^{\mathrm{MAX}}$ (see 24) and the actual energy delay gain. The energy saving opportunity increases when the output load is small, and when the number of stages in the path increases.

5.2. Activity and Signal Probability Effect on Sizing

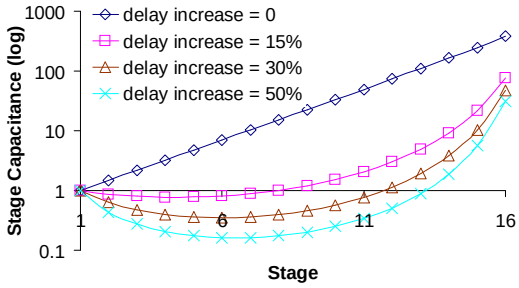
The more active a gate is, the more energy it consumes. In order to trade off delay and energy better, active gates in the timing critical path can be downsized more than inactive gates in the critical path. For instance, consider the circuit in Figure 10. The path from A to out is the timing critical. Input A has a fixed activity factor of 0.5, while the activity factor of input B is varied. In order to calculate the activity factor and signal probability of internal nodes, the method described in [20] for AF/SP propagation in combinational circuits is used - for a nand gate with uncorrelated inputs A and B and output O, the activity at its output is calculated as:

$$AF_O = AF_A \cdot SP_B + AF_B \cdot SP_A - \frac{1}{2} \cdot AF_A \cdot AF_B \tag{40}$$

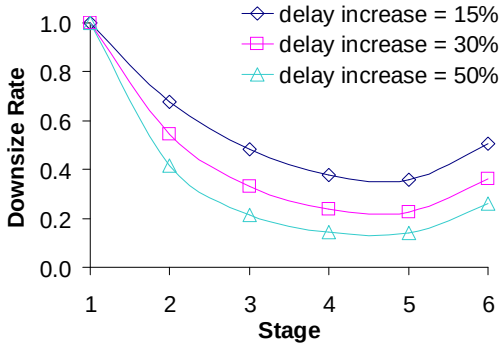
According to (40), the activity factor at the nand's gate output is $AF_{nand} = 0.25 + 0.5AF_B$ - the activity factor at the output of the nand is controlled by the activity factor of input B, and monotonically rises as AF_B increases.
When the delay constrains of the circuit are relaxed, As AF_B is increased, and with it AF_{nand}, we expect that the gates that are driven by the nand gate will get downsized at the expense of the gates driving the nand gate. Figure



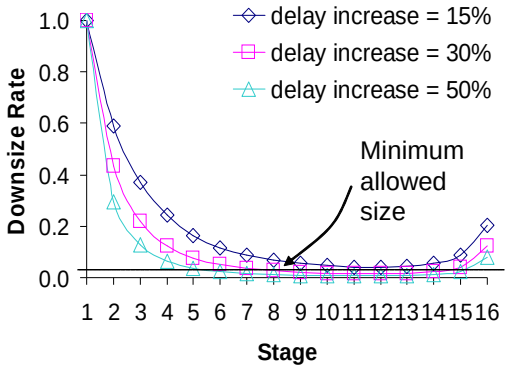
(a) Stage capacitance (chain of 6 inverters), for various delay increase rates (log scale)



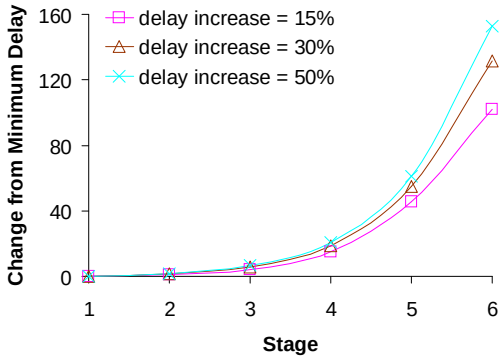
(b) Stage capacitance (chain of 16 inverters), for various delay increase rates (log scale)



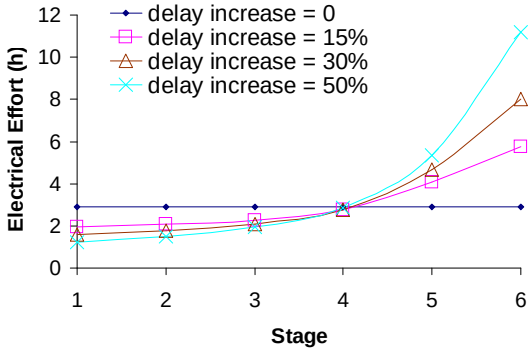
(c) Stage sizing factor (chain of 6 inverters) - ratio of gate capacitance to minimum delay capacitance, needed to meet the given delay increase rates value



(d) Stage sizing factor (chain of 16 inverters) - ratio of gate capacitance to minimum delay capacitance, needed to meet the given delay increase rates value



(e) Stage downsizing value - change in gate capacitance w.r.t minimum delay sizes to meet the given delay increase rates value



(f) Stage electrical effort (h), for various delay increase rates

Figure 6: **Inverter Chain** - sizing of the stages in an inverter chain

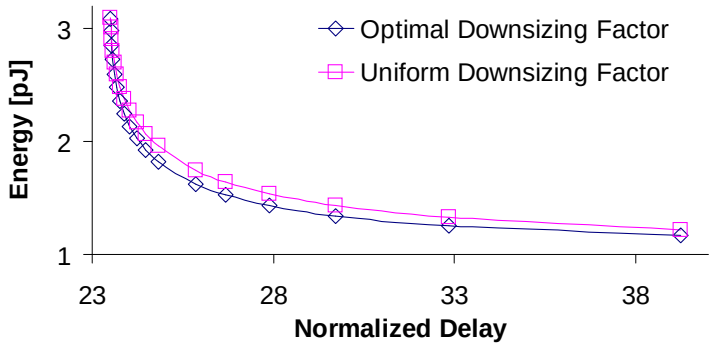


Figure 7: **Uniform vs. Optimal Downsizing**. Linear downsizing of an inverter chain in order to save energy by increasing the delay results in a non-optimal design - in this case 7% more energy could e save by tuning the sizing correctly.

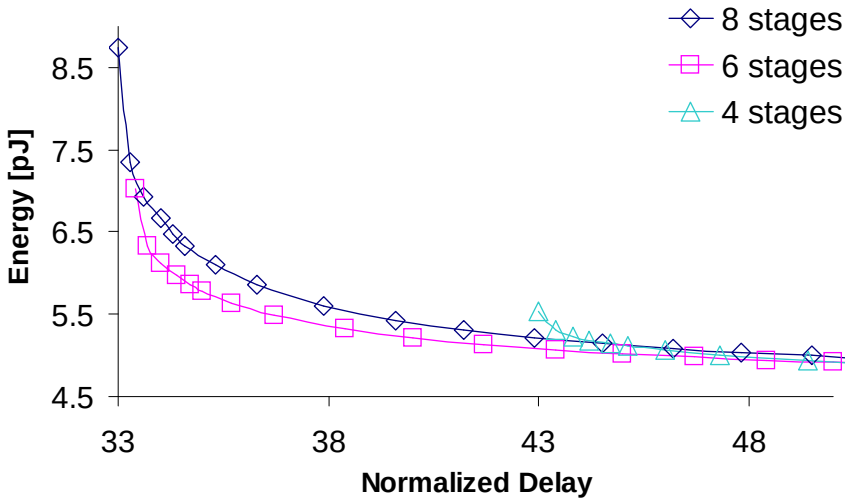
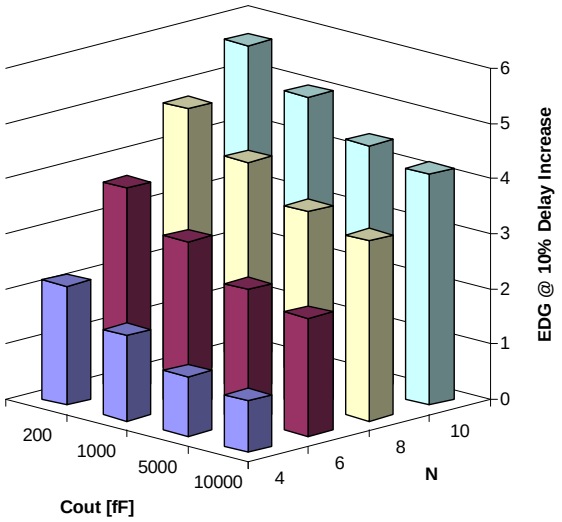


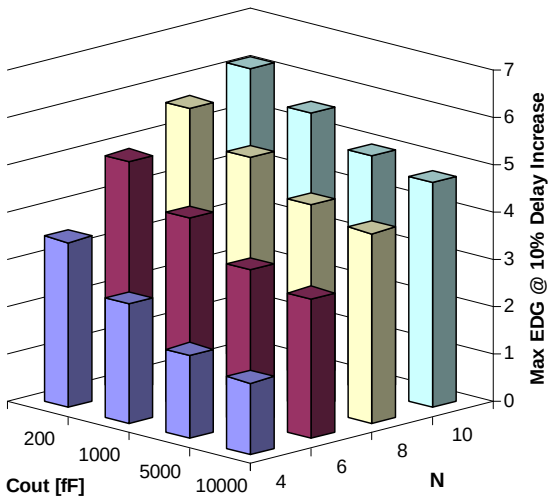
Figure 8: **Inverter Chain - Variable Length** - The chain length is varied in order to save a maximal amount of energy for each delay value

.

11 shows the sizing factor of each gate for various AF_B values, for a delay increase rate of 20%. We see that as AF_B increases, the sizing factor of gates 1 and 2 is increased, while the sizing factor of gates 5 and 6 is decreased.



(a) EDG Value of Various Inverter Chains at Delay Increase Rate of 10%



(b) EDG^{MAX} (analytical upper bound to EDG) of Various Inverter Chains at Delay Increase Rate of 10%

Figure 9: **Inverter Chain** - Comparison between energy delay gain and EDG^{MAX} (analytical upper bound to EDG)

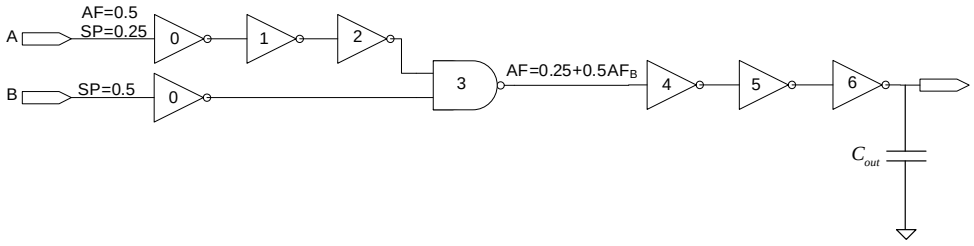


Figure 10: **Activity Effect on Sizing** the path from a to end is timing critical, and the activity of input b is varied

A similar observation holds for leakage dominant circuits, where the signal probability becomes the affecting parameter instead of the activity factor. $\bar{P_{leak_i}}$ in (15) depends on the signal probability. Therefore, it is expected that the sizing of each gate during the optimization process will be influenced by the signal probability at the gate input. For example, in an inverter, where the pmos transistor's size is twice the size of the nmos transistor, the leakage power of a single inverter can be estimated by:

$$\begin{aligned} \text{Inverter Leakage Power} &= SP \cdot C_{in} \cdot \frac{2}{3} \cdot P_{leak}(Pmos) \\ &+ (1 - SP) \cdot C_{in} \cdot \frac{1}{3} \cdot P_{leak}(Nmos) \end{aligned} \tag{41}$$

Where SP is the signal probability in the input of the inverter, C_{in} is the input capacitance of the inverter, and $P_{leak}(Nmos, Pmos)$ is the leakage power of Nmos and Pmos transistors respectively, per unit input capacitance. Figure 12 shows the sizes of the gates in a six stage inverter chain with input capacitance of 1ff and output load of 600ff with a small activity factor, when the delay increase rate is varied from 0% to 50%. The optimal sizing at each stage is clearly affected by the signal probability. Up to 50% difference in the sizing of the stages as a function of the signal probability can be observed (see delay increase of 50%, 4^{th} stage).

5.3. Comparing Analytical and Simulation-Based Optimization

In order to validate the correctness of the EDG optimization algorithm, the results of Section 5.1 are compared to simulation results. The simulation was performed using a proprietary circuit simulator combined with a proprietary numerical optimization environment, in a 32nm process. The circuit

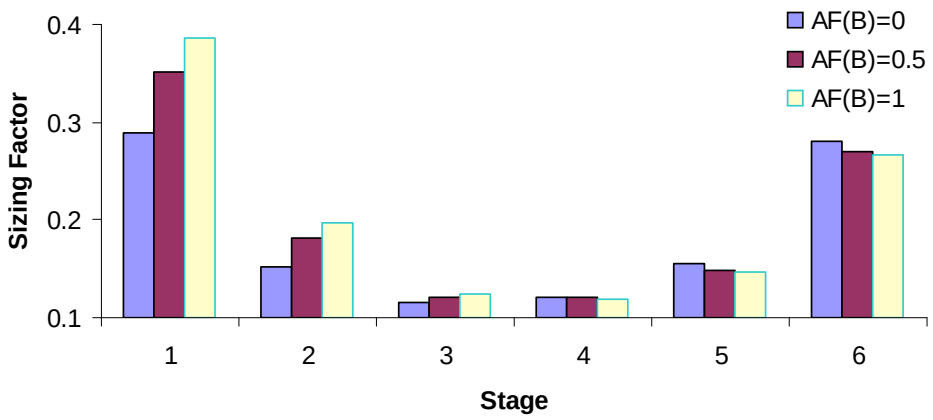


Figure 11: **Sizing Factor to Achieve 20% Delay Increase** as AF_b increases, the sizing factor of gates 1 and 2 is increased, while the sizing factor of gates 5 and 6 is decreased

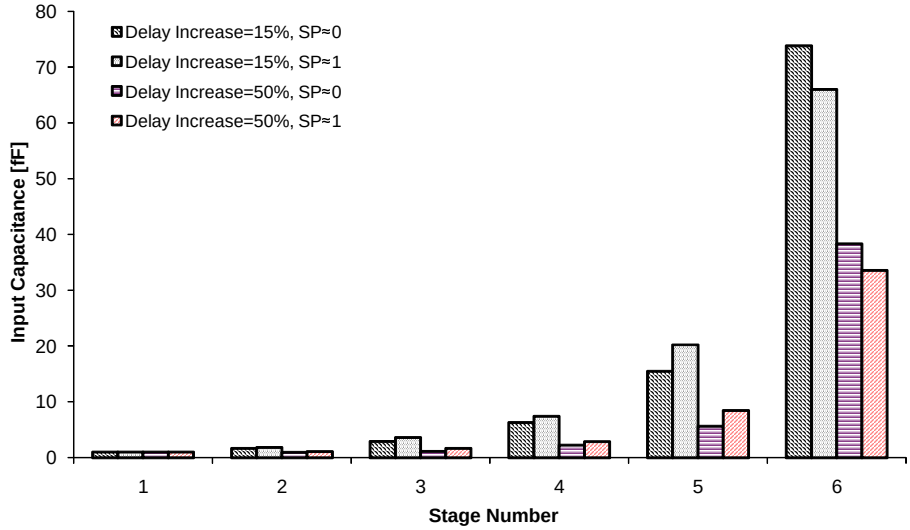


Figure 12: **Sizing of 6-stages Inverter Chain as a Function of SP** the sizing of the stages is sensitive to the signal probability changes, both for small and large delay increase values

was first optimized for minimum delay, which was used later as a reference. In order to get the EDG curves, the circuit was optimized by the simulation based tool for minimum energy, for several delay constraints.

Figure 13 presents the difference between the analytical computation (Section 5.1) and the simulation based optimization. The error is small, and ranges from a maximum of 7% to a minimum of $\tilde{0}$%. Obtaining the EDG curves using simulation based optimization is orders of magnitude slower than running the proposed analytical method. Table 1 compares the run time of simulation based optimization and the run time of the proposed analytical model for few inverter chain circuits. Note that simulation based optimization run time increases dramatically as the circuit complexity increases.

Circuit	Sim Based Optimization	Analytical Model Optimization
4-long Inverter Chain	240 sec	25 sec
8-long Inverter Chain	360 sec	40 sec
15-long Inverter Chain	1100 sec	70 sec

Table 1: **Comparison of Run Time - Simulation Based and Analytical Model Optimization**. The table compares the amount of time taken in order to generate an EDG plot consists of ten delay increase points.

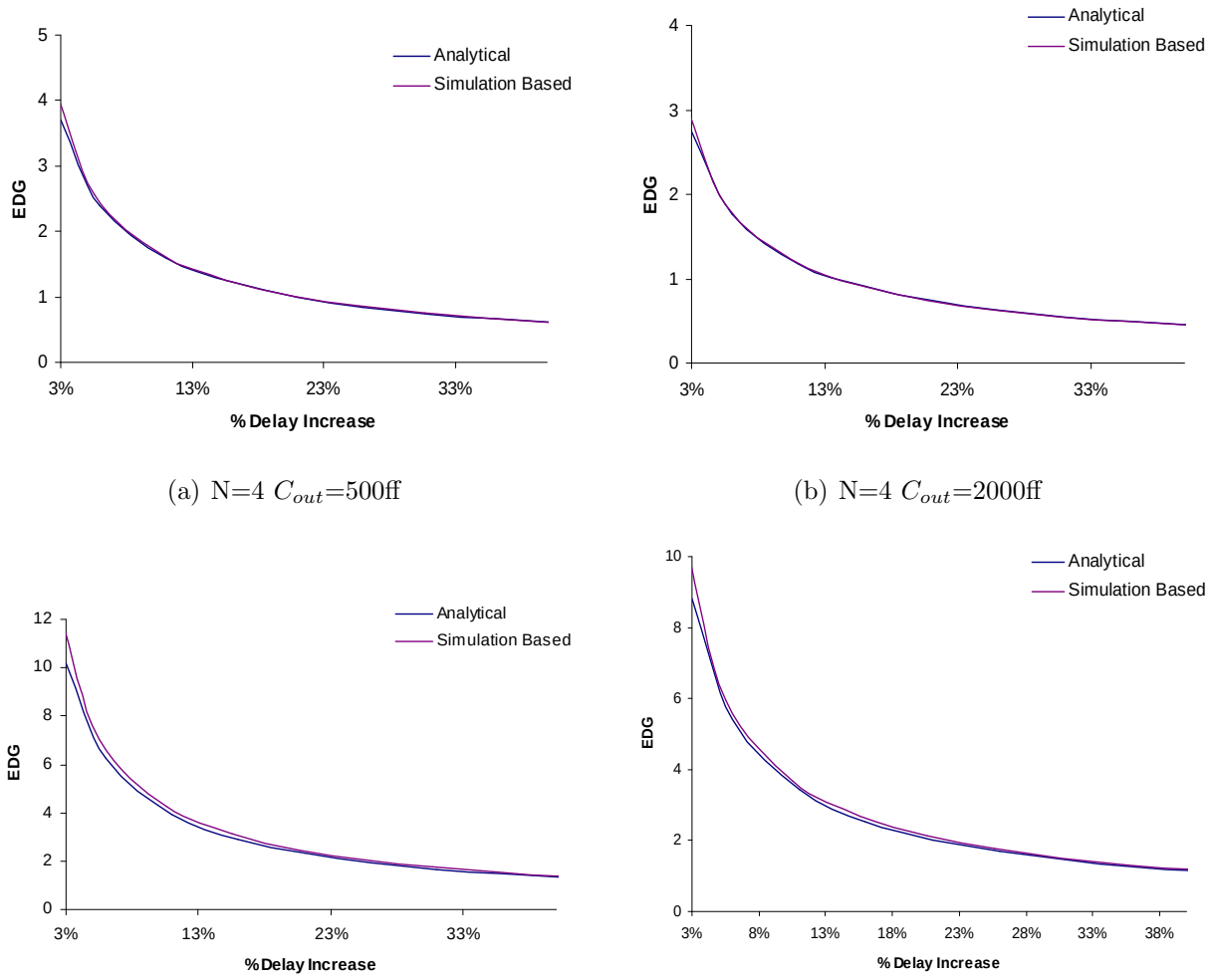
The analytical model was calibrated by computing the parasitics delay of an inverter (p) for the given technology, simply by comparing the output capacitance to the input capacitance of an unloaded inverter (see (8)).

6. Final Remarks and Conclusion

We have presented a design optimization framework that explores the power-performance space. The framework provides fast and accurate answers to the questions -

1. How much power can be saved by slowing down the circuit by x percent?
2. How to determine gate sizes for optimal power under a given delay constraint?

We introduced the energy/delay gain (EDG) as a metric for the amount of energy that can be saved as a function of increased delay. The method was demonstrated on a variety of circuits, exhibiting good correlation with



(a) N=4 C_{out}=500ff

(b) N=4 C_{out}=2000ff

(c) N=8 C_{out}=500ff

(d) N=8 C_{out}=2000ff

Figure 13: **Simulation Optimization of Inverter Chain with Comparison to Theoretical Computation**

accurate simulation-based optimizations. We have shown that around 25% dynamic energy can be gained when the delay constraint is relaxed by 5% in an optimal way, for circuits in 32nm technology which were initially designed for maximal operation speed. An upper bound of power savings in a given circuit can be obtained without optimization, in order to quickly assess

whether a downsizing effort may be justified for the circuit.

The method described in this work can be used by both circuit designers and EDA tools. Circuit designers can increase their intuition of the energy-delay tradeoff. The following rules of thumb can be derived from the experiments -

- **Minimum delay is power expensive.** By relaxing the delay, significant amount of dynamic energy could be saved. We have shown that under given conditions, for a 2-bit multiplexer up to 40% of dynamic energy could be saved when the delay constraint is relaxed by 10%.
- **A fixed uniform downsizing factor for all gates in the circuit would lead to an inefficient design in terms of energy.** The optimal downsizing factor is not uniform.
- **Increase delay by downsizing the "middle" gates.** In order to save energy with minimal impact on timing - the gates located in the middle (between he input and the load) are downsized the most. The downsizing factor increases as the delay constraint relaxes.
- **Increase delay by increasing the electrical effort towards the load.** Minimum delay design requires a constant tapering factor. Typically, a "fanout of 4" is used ([8]). Minimum energy design (when neglecting short circuit power) requires high tapering factor, that decreases the number of stages. When performance is compromised to save energy, the tapering factor of the stages must *increase* towards the external load. The tapering factor increases as the delay constraint is relaxed. Note that this result is applicable only when the external load is larger than the input capacitance.
- **Downsizing of the gates reduces both dynamic and leakage energy dissipation.** Both dynamic and leakage energy dissipation depend linearly on the size of the gates. By downsizing the gates, both dynamic and leakage energy are reduced.
- **The power optimization has to be performed under a given workload.** The activity factor and signal probability influence the optimized circuit's sizing. Different tests may result in different sizing. Using random tests, rather then typical tests to optimize the circuit may lead to sub-optimal design.

7. Acknowledgments

We would like to thank Yoad Yagil for his valuable inputs.

References

[1] V. Zyuban, P. N. Strenski *Balancing hardware intensity in microprocessor pipelines*. IBM J. RES. & DEV. VOL. 47 NO. 5/6 SEPTEMBER/NOVEMBER 2003

[2] V. Zyuban, P. Strenski, *Unified Methodology for Resolving power-Performance Tradeoffs at the Microarchitectural and Circuit Levels*, in Proc. of International Symposium on Low Power Electronics and Design, Monterey, CA, USA, pp. 166-171, Aug. 2002

[3] R. Gonzalez, B. Gordon, M. Horowitz, *Supply and Threshold Voltage Scaling for Low Power CMOS*, IEEE Journal of Solid-State Circuits, Vol. 32, No. 8, pp. 1210-1216, August 1997.

[4] H. Dao, B. Zeydel, V. Oklobdzija, *Energy optimization of Pipelined Digital Systems Using Circuit Sizing and Supply Scaling*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 14, No. 2, pp. 122-134, February 2006

[5] V. Oklobdzija, R. K. Krishnamurthy, *High-Performance Energy-Efficient Microprocessor Design*, Springer, 2006.

[6] Benini, L., De Micheli, G., Macii, E. *Designing low-power circuits: Practical recipes* IEEE Circuits and Systems Magazine 1, March 2001, 625

[7] V. Khandelwal, A. Srivastava *Leakage Control Through Fine-Grained Placement and Sizing of Sleep Transistors*, in Procs. of ICCAD 2004, pp 533 - 536.

[8] Ivan E. Sutherland, Robert F. Sproull, and David F. Harris *Logical Effort: Designing Fast CMOS Circuits*. Margan Kaufmann. ISBN 1558605576 1999

[9] Stephen Boyd, Lieven Vandenberghe *Convex Optimization*. Cambridge University Press, 2006

[10] Stephen Boyd, Seung Jean Kim, Lieven Vandenberghe, Arash Hassibi *A Tutorial on Geometric Programming* Revised for Optimization and Engineering, July 2005

[11] S. Boyd, S. Kim, D. Patil, M. Horowitz, *Digital Circuit Optimization via Geometric Programming*, Operations Research, Vol. 53, No. 6, pp. 899-932, November/December 2005

[12] Radu Zlatanovici and Borivoje Nikoli, *Power - Performance Optimization for Custom Digital Circuits*. PATMOS 2005, LNCS 3728, pp. 404-414

[13] Paul I.Penzes and Alain J.Martin, *Energy-Delay Efficiency of VLSI Computations*, GLSVLSI02, April, 2002

[14] D. Markovic, V. Stojanovic, B. Nikolic, M. Horowitz, R. Brodersen, *Methods for True Energy-Performance Optimization*, IEEE Journal of Solid-State Circuits, Vol. 39, No. 8, pp. 1282-1293, August 2004

[15] Almir Mutapcic, Kwangmoo Koh, Seungjean Kim, Lieven Vandenberghe and Stephen Boyd, GGPLAB: A Simple Matlab Toolbox for Geometric Programming, *http://www.stanford.edu/ boyd/ggplab/*, May, 2006

[16] Alain J. Martin, *Towards an energy complexity of computation*, Information Processing Letters 77, 181187, 2001

[17] Sasan Iman, Massoud Pedram, *Logic Synthesis for Low Power VLSI Designs*, Springer. ISBN: 978-0-7923-8076-4, 1998

[18] Alon Naveh, Efraim Rotem,Avi Mendelson, Simcha Gochman,Rajshree Chabukswar, Karthik Krishnan, Arun Kumar, *Power and Thermal Management in the Intel Core Duo Processor*, Intel Technology Journal, Volume 10, Issue 2, May 15, 2006,

[19] AMD Press Release, *AMD Phenom X4 9100e processor enables fullfeatured, sleek and quiet quad-core PCs*, AMD Press Resources Web Page, March 27, 2008

[20] A. Ghosh , S. Devadas , K. Keutzer , J. White, *Estimation of average switching activity in combinational and sequential circuits*, Proceedings of the 29th ACM/IEEE conference on Design automation, p.253-259, June 08-12, 1992, Anaheim, California, United States

[21] *International Technology Roadmap for Semiconductors*, 2007 Edition, http://www.itrs.net/Links/2007ITRS/Home2007.htm

[22] Rahman H., Chakrabarti C. *A leakage estimation and reduction technique for scaled CMOS logic circuits considering gate leakage*, Proceedings of the International Symposium on Circuits and Systems (ISCAS), pp. 297300, May 2004

[23] Zhanping Chen, Mark Johnson, Liqiong Wei, Kaushik Roy *Estimation of standby leakage power in CMOS circuits considering accurate modeling of transistor stacks*, Proceedings of the International Symposium on Low Power Electronics and Design, pp. 239-244, 1998

[24] Yongjun Xu, Zuying Luo, Xiaowei Li *A maximum total leakage current estimation method*, Proceedings of the International Symposium on Circuits and Systems (ISCAS), pp. 757-760, May 2004

[25] Chung-Ping Chen, Chu C.C.N., Wong D.F. *Fast and exact simultaneous gate and wire sizing by Lagrangian relaxation*, IEEE Trans. on CAD of Integrated Circuits and Systems 18(7), pp. 1014-1025, 1999