Part I

DIGITAL DESIGN AND POWER MANAGEMENT

1

DESIGN IN THE ENERGY–DELAY SPACE

Massimo Alioto

Department of Information Engineering, University of Siena, Siena, Italy

Elio Consoli and Gaetano Palumbo

Department of Electrical, Electronic and Systems Engineering, University of Catania, Catania, Italy

1.1 INTRODUCTION

In the past, the traditional constant-field scaling [1] has led CMOS technology to continuous improvements in the speed performances while maintaining constant power density. However, a fundamental limit of constant-field scaling manifests due to the nonscaling of subthreshold slope and the increase of gate leakage as long as the minimum feature size scales down [2,3]. Overall, the consequent continuous increase in energy consumption has become the major concern limiting the speed performances of VLSI Integrated Circuits [4], insomuch as, even for high-speed systems, designs undergo a "power limited" regime [5].

As a consequence, it is no longer possible to focus solely on optimizing the speed of circuits regardless their energy [6]. Rather, the achievement of energy efficiency, that is, finding the circuits designs allowing us to reach the desired speed under the

Advanced Circuits for Emerging Technologies, First Edition. Edited by Krzysztof Iniewski.

minimum dissipation, has become the primary target [7]. Thus, a deep understanding of the energy–delay (E–D) tradeoff and the related design issues is crucial.

In this chapter, energy and delay models of digital CMOS circuits are firstly presented (Section 1.2), since they constitute the base for any E–D-related optimization technique not fully relying on simulations. The theoretical background relative to exploration of the E–D space and the identification of the optimum, that is, energy efficient, designs is then reported (Section 1.3). Practical design approaches and the optimization of the various design knobs are discussed, together with exemplificative results relative to various circuits (Section 1.4). Finally, we deal with the slightly higher abstraction level of whole pipelined systems and the related energy-efficient design criteria (Section 1.5).

1.2 ENERGY AND DELAY MODELING

1.2.1 Delay: the Logical Effort as a Modeling Approach

From their basic structure, it is evident that CMOS logic gates can be simply modeled as decoupled RC blocks [8], as shown in Fig. 1.1.

The resistance of a MOS transistor is inversely proportional to its width W. When considering complex CMOS gates, the evaluation of the total equivalent resistance of pull-up (PUN) and pull-down (PDN) networks can be approximately performed by summing the resistances of stacked blocks of transistors and by summing the conductances of parallel blocks [9].

The equivalent capacitance at the input of a MOS transistor, C_G, is proportional to WL (L is the transistor channel length) and typically nearly equal to $C_{\text{ox}}WL$ [9]. The self-loading in a CMOS gate is due to diffusion capacitances and can be expressed as [7]

$$C_D = C_{D,A}WL_d + C_{D,P}(2W + 2L_d) \tag{1.1}$$

where L_d is the length of drain/source diffusions and $C_{D,A}$ ($C_{D,P}$) are the capacitances per unit area (perimeter) of drain/source-bulk junctions. By neglecting the $2L_dC_{D,P}$ term, C_D can be considered nearly proportional to W.

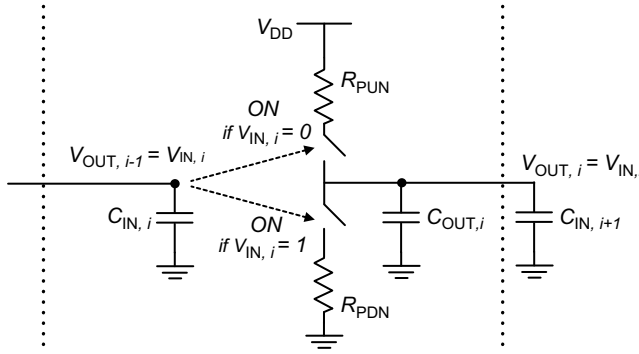

Figure 1.1. CMOS logic gates seen as decoupled *RC* blocks.

Summarizing, by considering a CMOS gate one has that

$$C_{\mathrm{IN}} \propto WL, C_{\mathrm{OUT}} \propto W, R_T \propto L/W \tag{1.2}$$

where C_{IN} is the capacitance of the input where the critical signal is applied, C_{OUT} is the output diffusion capacitance and R_T is the PUN/PDN resistance.

Usually, all the channel lengths are minimum and we can see the considered gate as a version scaled by a factor α (in terms of channel width) of a reference gate of the same type, called the "template" gate. Such a gate exhibits parameters $C_{\mathrm{IN,ref}}$, $C_{\mathrm{OUT,ref}}$, and $R_{T,\,\mathrm{ref}}$, and the following relationships hold [7]:

$$C_{\mathrm{IN}} = \alpha C_{\mathrm{IN,ref}}, \quad C_{\mathrm{OUT}} = \alpha C_{\mathrm{OUT,ref}}, \quad R_T = R_{T,\mathrm{ref}}/\alpha \tag{1.3}$$

Hence, any timing parameter of the gate can be expressed as [8]

$$t_D = KR_T(C_{\mathrm{OUT}} + C_L) = K\left(R_{T,\mathrm{ref}} C_{\mathrm{IN,ref}} \frac{C_L}{C_{IN}} + R_{T,\,ref} C_{\mathrm{OUT,ref}}\right) \tag{1.4}$$

where C_L is the external output load, and K depends on the kind of timing parameter (delay, fall/rise times) and on the slope of the input.

The RC model in (1.4) was revisited in [10] to obtain a new one normalized to (i.e., independent from) technology: the Logical Effort model. Basically, formula (1.4) is divided by $R_{\mathrm{INV}} C_{\mathrm{INV}}$, which is the product of the resistance and input capacitance of a symmetrical inverter. Once normalized, the timing parameter (e.g., delay or rise/fall time) of the gate, t_D, becomes

$$t_D = \tau(gh + p) = \tau(f + p) = \tau d \tag{1.5}$$

where the various quantities correspond to

$$\tau = KR_{\mathrm{INV}} C_{\mathrm{INV}} \tag{1.6}$$

$$g = \frac{R_{T,\,\mathrm{ref}} C_{\mathrm{IN,\,ref}}}{R_{\mathrm{INV}} C_{\mathrm{INV}}} \tag{1.7}$$

$$h = \frac{C_L}{C_{\mathrm{IN}}} \tag{1.8}$$

$$p = \frac{R_{T,\,\mathrm{ref}} C_{\mathrm{OUT,\,ref}}}{R_{\mathrm{INV}} C_{\mathrm{INV}}} \tag{1.9}$$

The parameter τ allows to normalize t_D to technology. The parameter g is called "logical effort" and is a feature dependent on the gate's topology and hence not affected by its absolute sizing. The parameter h is called "electrical effort," and it is equal to the fanout of the gate. The parameter p is called "parasitic delay" and represents the intrinsic delay contribution due to the self-loading. As for g, p, is a feature dependent on the gate's topology and hence not affected by its absolute sizing. Finally, the product $f = gh$ is called "stage effort."

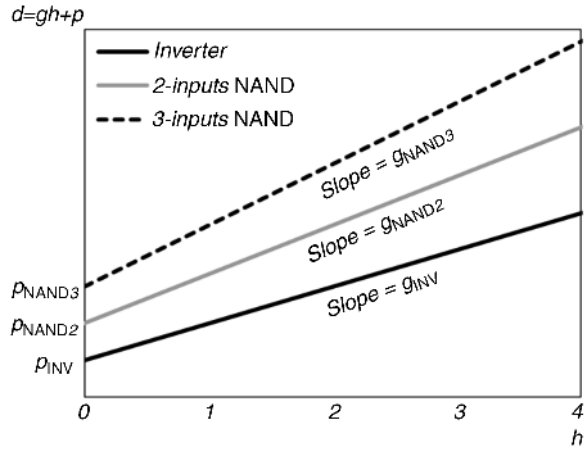

Figure 1.2. Geometrical interpretation of logical effort and parasitic delay.

It is apparent that the normalized timing parameter d is a linear function of h, as shown in Fig. 1.2. The logical effort, g, represents the slope of such a line, whereas the parasitic delay p is the minimum achievable value of d, obtained for $h = 0$, that is, for zero external load or for $C_{\mathrm{IN}} \gg C_{\mathrm{L}}$.

The Logical Effort model is valid also in the case of nonstatic CMOS gates, such as the dynamic ones and those including pass-transistors (PTs) and transmission gates (TGs). When considering dynamic gates, one often has to deal with keepers introducing a current contention with the evaluation path in the gate. A multiplicative factor $r>1$ can be introduced to modify both parameters g and p, whose value is [10]

$$r = \frac{1}{1 - \frac{r_{\mathrm{eval}}}{r_{\mathrm{kpr}}}} \tag{1.10}$$

where r_{eval} is the equivalent resistance of the evaluation path in the dynamic gate, and r_{kpr} is the resistance of the keeper. Also TGs and PTs can be straightforwardly introduced in the Logical Effort framework. The only limitation is that (a chain of) TGs (or PTs) have to be included in an initial gate with driving capability, that is, connected to V_{DD} and/or GND [10].

The model described so far suffers from some limitations:

(a) The evaluation of equivalent resistances requires several approximations to manage the various effects arising in deep-submicron technologies and influencing the I–V behavior of MOS transistors [11].

(b) The model in (1.4)–(1.5) deals with the self-loading effect through a single capacitance C_{OUT}. However, when the PUN and/or PDN are made up by stacked (blocks of) transistors, the capacitances in their internal nodes give a further contribution to the parasitic delay [12].

The delay and rise/fall times of CMOS gates both significantly depend on the input transition time (or slope), which is neglected in (1.5).

Starting from the basic estimation of g and p parameters [10], which can be straightforwardly carried out by analyzing the gates topology, several attempts have been made to develop model extensions in order to capture the above effects, although they have resulted in quite complex models.

Nevertheless, apart from the necessity to model the input slope impact, the general applicability of (1.5) is still retained when referring to a specific kind of timing parameter (delay, rise/fall times) and to one of the inputs of a logic gate. Therefore, one can characterize a logic gate through simulations as shown in [10,12] to extract accurate estimations of g and p.

The input slope impact can be quite accurately modeled with a further linear term as in the following [13]

$$d = gh + p + \eta d_{\text{in}} \tag{1.11}$$

where η is an additional parameter to be characterized, and d_{in} is the normalized (according to Logical Effort approach) input rise/fall time, that is, the normalized output rise/fall time of the gate driving the considered one.

1.2.2 Delay: the Logical Effort as an Optimization Approach

So far we have discussed the modeling potentials of Logical Effort approach. Actually, the Logical Effort theory also leads to useful equations allowing to maximize the speed of a logic path constituted by several gates, that is, to size them in order to minimize the overall path delay [10].

In the following, as done elsewhere, this theory is reported by focusing on the delay model in (1.5), which does not account for input slope. Indeed, although the Logical Effort modeling accuracy is weakened by this lack, we will show that the minimum delay condition is achieved when the stage efforts of the various gates in the path are equal. This means that the minimum delay condition is achieved when the input and output slopes of the gates in the path are quite similar. Under this condition, the original Logical Effort model in (1.5) is sufficiently accurate [10,12].

Let us consider a multistage network comprising a path made up of N-cascaded logic gates, the ith of which featured by parameters g_i, p_i, and

$$h_i = \frac{C_{L,i}}{C_{\text{IN},i}} = \frac{C_{\text{IN},i+1} + C_{\text{off},i}}{C_{\text{IN},i}} \tag{1.12}$$

where $C_{\text{IN},i}$ and $C_{\text{IN},i+1}$ are the input capacitances of the ith and $(i+1)$th gate in the path, respectively, while $C_{\text{off},i}$ is the overall capacitance of other gates loading stage i but not belonging to the path under analysis, as shown in Fig. 1.3. The "path logical effort," G, and "path parasitic delay," P, can be defined as

$$G = \prod_{i=1}^{N} g_i \tag{1.13}$$

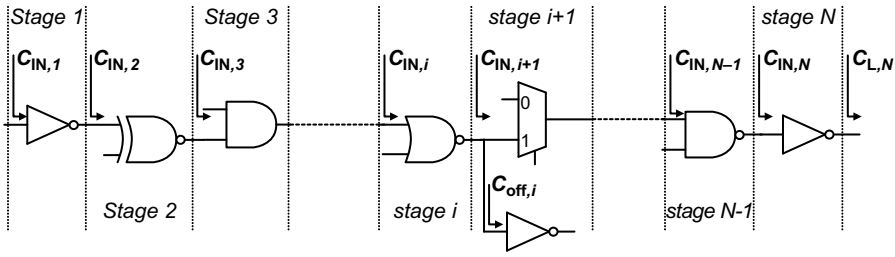

Figure 1.3. Multistage path.

$$P = \sum_{i=1}^{N} p_i \tag{1.14}$$

and, by defining the "branching effort" b_i of the ith stage as the proportion between the total load of gate i and the fraction lying on the considered path,

$$b_i = \frac{C_{\mathrm{IN},i+1} + C_{\mathrm{off},i}}{C_{\mathrm{IN},i+1}} \geq 1 \tag{1.15}$$

we can also introduce the "path electrical effort," H, and the "path branching effort," B, of the entire path through the following formulas:

$$HB = \prod_{i=1}^{N} h_i \tag{1.16}$$

$$H = \frac{C_{L,N}}{C_{\mathrm{IN},1}} \tag{1.17}$$

$$B = \prod_{i=1}^{N} b_i \tag{1.18}$$

being $C_{L,N}$ and $C_{\mathrm{IN},1}$ the final load and the first stage input capacitance, respectively.

Finally, the overall "path effort" F is equal to

$$F = \prod_{i=1}^{N} g_i h_i = \prod_{i=1}^{N} f_i = GBH. \tag{1.19}$$

The total normalized delay of the considered path is

$$D = \sum_{i=1}^{N} (g_i h_i + p_i) \tag{1.20}$$

and, assuming for the moment that not only g_i and p_i, but also b_i, are constant parameters (although this is not true in general), one has that D is a function only of the capacitive gains of the various stages on the path.

As previously anticipated, the Logical Effort approach can serve also as an optimization method to minimize delay. In particular, considering that

$$h_1 = \frac{H}{h_2 h_3 \dots h_N} \tag{1.21}$$

the condition for minimum path delay can be written as

$$\frac{\partial D}{\partial h_i} = \frac{\partial \left(g_1 \frac{H}{h_2 h_3 \cdots h_N} + \sum_{i=2}^{N} (g_i h_i + p_i) \right)}{\partial h_i} = g_i - \frac{g_1 H}{h_1 (h_2 h_3 \cdots h_N)} = 0 \tag{1.22}$$

which leads to

$$g_1 h_1 = g_i h_i \quad \forall i \tag{1.23}$$

that is, the stage effort has to be the same for all stages in the path. Moreover, according to (1.19) and (1.23), the optimum stage effort is equal to

$$f_{\text{opt}} = \sqrt[N]{GBH}. \tag{1.24}$$

According to the previous considerations, parasitic delays do not enter in the optimization and, considering that the final load and the first stage input capacitance are known, the minimum achievable delay of the path with fixed topology and stages number N is known *a priori*, and it is equal to

$$D_{\text{opt}} = \sqrt[N]{GBH} + P \tag{1.25}$$

where G, B, and H have fixed value independently from the absolute sizing of the various stages (true only if g_i, b_i, and p_i can be assumed as constant).

The Logical Effort can be used as a method to size gates in order to minimize delay given that, according to (1.23) and (1.24), it is sufficient to set

$$f_i = \sqrt[N]{GHB} \quad \forall i \tag{1.26}$$

leading to

$$C_{\text{IN},i} = \frac{g_i b_i C_{IN,i+1}}{\sqrt[N]{GHB}} \quad \forall i \tag{1.27}$$

which are a set of relationships that can be applied by starting from the Nth gate ($C_{L,N}$ is known) and proceeding backward along the path or starting from the first gate ($C_{\text{IN},1}$ is known) and proceeding onward along the path.

In practical cases, this condition of constants g_i, b_i, and p_i cannot be satisfied for several reasons, which are listed in the following.

1. The factor r in (1.10) is a function of the gate and keeper absolute sizes when a constant ratio between their driving capabilities is not maintained.
2. The branching effect in (1.15) due to gate and/or diffusion capacitances of transistors outside the path can often be a function of the absolute size of the ith gate

itself. This happens when a constant proportion between the absolute values of $C_{IN,i+1}$ and $C_{off,i}$ is not maintained.

3. Global interconnections can be modeled as equivalent *RC* ladder blocks and hence handled as done for stacked transistors and TGs/PTs. However, their length is normally fixed and hence the resistive and capacitive contributions they introduce lead to g and b values that are functions of the absolute size of the gates driving such interconnections.
4. Lumped capacitances associated with local interconnections in each of the internal nodes in a circuit lead to additional delay contributions. They can be subdivided in a contribution given by the gate driving the considered node (affecting parasitic delay), in a contribution given by the gates loading the considered node (affecting electrical effort) and in a constant contribution (affecting branching effort). The latter contribution is gate-size dependent, while the first two ones lead to complex nonlinear dependencies, and a linearization is not always feasible.

It is apparent that in all these cases several nonlinearities emerge and do not allow the optimization described in (1.23)–(1.27) to be straightforwardly applied. Therefore, in order to minimize the delay of paths including complex branching effects and the impact of interconnections, a need for iterative procedures arises, thereby weakening the logical effort handiness.

1.2.3 Energy: A Comprehensive Model

Being the optimization of circuits from the joint speed-consumption perspective the focus of this chapter, it is necessary to clarify the metrics that will be used to quantify the consumption at the abstraction level this chapter deals with, that is, the transistor-level one. In particular, two metrics are available: power and energy [14].

Both metrics are actually interchangeable and choosing one or another is simply a matter of convention as long as transient (i.e., dynamic and short-circuit) and static (i.e., leakage) dissipative contributions are properly weighed [15]. In the following, energy is chosen as the metric for circuits consumption. This implies that transient contributions relative to a generic circuit operation have to be simply summed, whereas static leakage-related power has to be multiplied by the time between successive operations (e.g., the duration of a clock cycle in a pipelined system) and summed to the previous transient contribution to obtain the overall energy dissipation.

In the following, a model accounting for the above contributions [16] is reported. This model aims at the extraction of a factor n featuring a logic gate and such that the overall gate energy, E, can be simply expressed as linearly proportional to the input capacitance, C_{IN}, that is, to the gate size

$$E = \chi C_{IN}. \tag{1.28}$$

Such a model intentionally excludes the energy dissipated in charging/discharging the load C_L, but includes that dissipated in charging/discharging C_{IN}. Again, it is simply a matter of convention.

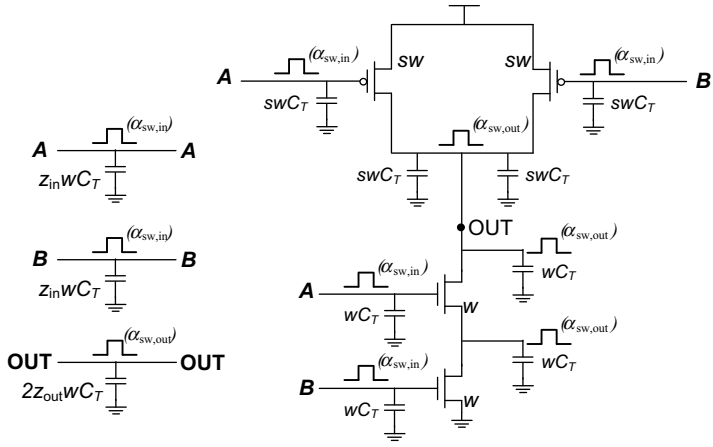

Figure 1.4. Capacitive contributions determining dynamic energy in a gate.

Let us consider a static CMOS gate such as the 2-inputs NAND shown in Fig. 1.4, where also the various capacitive contributions determining the dynamic dissipation are depicted. One can distinguish among capacitances lying in the input nodes and switching according to the transition probability of the inputs, and capacitances lying in the output node (or in the internal ones featuring stacked structures) and switching according to the transition probability of the output (internal) node. Moreover, each of these capacitances is made up by transistors related contributions (gate capacitances for the input nodes and diffusions capacitances for the output and/or internal nodes) and parasitic capacitances due to local wires.

Accordingly, the average dynamic energy (in a clock cycle) of a CMOS gate can be expressed as

$$E_{\mathrm{DYN}} = \left[(1 + s + z_{\mathrm{in}})\,\alpha_{\mathrm{sw,\,in}} + (1 + s + z_{\mathrm{out}})\,\alpha_{\mathrm{sw,\,out}}\right] w C_T m V_{\mathrm{DD}}^2 \tag{1.29}$$

where (see Fig. 1.4 for exemplification):

- w is the normalized width (with respect to the minimum feasible value $W_{\min}$ imposed by the technology) of each NMOS transistor inside the gate (assuming that all NMOS have the same width and minimum lengths);
- C_T is the gate capacitive contribution relative to a minimum sized transistor. It can be defined as $C_{\mathrm{INV}}/3$, where C_{INV} is the input capacitance of a symmetrical minimum inverter (i.e., with $W_{\mathrm{PMOS}} = 2W_{\mathrm{NMOS}} = 2W_{\min}$);
- s is a multiplicative factor that defines the widths of PMOS (again all equal and with minimum lengths) with respect to the NMOS ones, thus leading to a certain skew in the speed of PUN and PDN [10];
- m is the number of inputs of the gate;

- $\alpha_{\text{sw,in}}$ and $\alpha_{\text{sw,out}}$ are the activity factors weighing the static probabilities of a full $0 \rightarrow 1 \rightarrow 0$ transition in a clock cycle [17] for the input and output/internal nodes of the gate (for the moment we assume a unique $\alpha_{\text{sw,in}}$ value for all the inputs and a unique $\alpha_{\text{sw,out}}$ value for output and internal nodes);
- we assumed that gate and diffusion (drain-bulk and source-bulk) capacitances are nearly equal [12];
- z_{in} and z_{out} weigh those local parasitic capacitive contributions at the input and at the output of the gate that are dependent on the size w of the gate itself. Although the dependence of such parasitics on w is formally complex and nonlinear, linear fittings can be extracted without seriously compromising the estimation of lumped local wires capacitances. Hence, the overall local wires capacitance in a generic node j, $C_{\text{par},j}$, can be expressed as [16]

$$C_{\text{par},j} = z_{\text{out},i-1,j} W_{i-1} C_T + z_{\text{in},i,j} w_i C_T \tag{1.30}$$

  being j the node at the output and the input of the $(i-1)$th and the ith stage, respectively.
- we have inherently assumed that each transistor contributes to energy consumption with a single gate and a single parasitic capacitance (the approximation of considering a single intermodal capacitance for each stacked transistor is simple but reasonably accurate).

A similar analysis concerning the static dissipation of a CMOS gate can be carried out and the average energy (in a clock cycle) due to subthreshold and gate leakage can be expressed as

$$E_{\text{STAT}} = w \left(\beta_{\text{sub},n} \frac{\rho_{\text{sub},n}}{T_{\text{sub},n}} + s\beta_{\text{sub},p} \frac{\rho_{\text{sub},p}}{T_{\text{sub},p}} + \frac{\rho_{\text{gate},n}}{T_{\text{gate},n}} + s\frac{\rho_{\text{gate},p}}{T_{\text{gate},p}} \right) V_{\text{DD}} T_{\text{CK}} \theta \tag{1.31}$$

where

- $\rho_{\text{sub},n}$ and $\rho_{\text{sub},p}$ ($\rho_{\text{gate},n}$ and $\rho_{\text{gate},p}$) are parameters depending on technology and approximately constant for any gate. They include the dependences of the subthreshold (gate) leakage current of a single transistor on threshold voltage, on the applied biases (assuming $V_{\text{GS}} = 0$ and $V_{DS} = V_{DD}$), on the temperature and on technology parameters for a NMOS and PMOS, respectively;
- $T_{\text{sub},n}$ and $T_{\text{sub},p}$ ($T_{\text{gate},n}$ and $T_{\text{gate},p}$) are factors that include the effect of the PDN and PUN topologies on their subthreshold (gate) leakage currents, respectively (by averaging out the various currents for each inputs combination).
- $\beta_{\text{sub},n}$ and $\beta_{\text{sub},p}$ average the subthreshold leakage currents of PDN and PUN according to static probabilities of logic values at input and output nodes of the gate (obviously $\beta_{\text{sub},n} + \beta_{\text{sub},p} = 1$);
- T_{CK} is the clock period duration;
- θ is a factor to include the relation between the durations of active and inactive modes (or standby) for the part of the system where the considered gate lies.

Basically, it is a correction factor leading to an effective clock period, $T_{CK}\theta$, which properly weighs the impact of static dissipation compared to dynamic one.

The above expressions (1.29) and (1.31) can be further complicated to more accurately model some effects while still remaining proportional to the parameter identifying the gate size, that is, w. For instance, (1.29) and (1.31) can be easily generalized to deal with gates with nonminimum channel lengths, with nonstatic (e.g., dynamic) gates, to more accurately weigh the impact of internodal capacitances on dynamic energy and of stacking effect on leakage, to consider the cases where some NMOS (PMOS) transistor within the PDN (PUN) has a width proportional but not equal to w, and so on. Hence, such models do not lead to any loss of generality. Furthermore, as already discussed for the Logical Effort model, many of the parameters in (1.29)–(1.31) can be accurately characterized through simulations.

Once E_{DYN} and E_{STAT} have been found, the overall energy dissipation of the gate is

$$E = E_{DYN} + E_{STAT}. \tag{1.32}$$

According to the previous definitions, C_{IN} can be expressed as

$$C_{IN} = (1 + s + z_{in})wC_T. \tag{1.33}$$

It is worth noting that this is the same value entering in the definition of Logical Effort parameters g and h, that is, it is the input capacitance seen at one of the gate inputs.

Finally, the parameter $\chi = E/C_{IN}$ can be expressed as

$$\chi = \left(\alpha_{sw,in} + \alpha_{sw,out}\frac{(1 + s + z_{out})}{(1 + s + z_{in})}\right) m V_{DD}^2 + \frac{\left(\beta_{sub,n}\frac{\rho_{sub,n}}{T_{sub,n}} + s\beta_{sub,p}\frac{\rho_{sub,p}}{T_{sub,p}} + \frac{\rho_{gate,n}}{T_{gate,n}} + s\frac{\rho_{gate,p}}{T_{gate,p}}\right)}{(1 + s + z_{in})C_T} V_{DD}T_{CK}\theta. \tag{1.34}$$

The above model neglects short-circuit dissipation. Given the increasing V_{TH}/V_{DD} ratios, this contribution tends to relatively decrease with technology scaling [9]. Nevertheless, when the input rise/fall times are quite large, the impact of short-circuit energy can be nonnegligible.

Differently from the dynamic and leakage ones, short-circuit contribution cannot be approximated as linearly dependent on the gate size. Indeed, it increases with gate size for three reasons:

- for the linear dependence of the PDN and PUN currents on w;
- for the approximately proportional dependence on the input rise/fall time, that is, on the output rise/fall time of the preceding gate [9];
- for the approximately inverse dependence on the output rise/fall time of the gate itself [9].

The last two terms can be assumed (by neglecting the parasitic delays in the computation of input rise/fall times) as nearly linearly dependent on w.

Overall, the short-circuit dissipation can be equaled to

$$E_{SC} = \frac{d_{in}}{d_{out}} \rho_{sc} [(T_{sc,n} + sT_{sc,p}) \alpha_{sw,out}] w \tag{1.35}$$

where d_{in} and d_{out} are input and output rise/fall times according to Logical Effort model, while parameters $T_{sc,n}$ and $T_{sc,p}$ average the various possible output transition cases according to PDN and PUN topologies. Finally ρ_{sc} is a further parameter accounting for the impact of technology and V_{DD}.

1.3 ENERGY–DELAY SPACE ANALYSIS AND HARDWARE-INTENSITY

1.3.1 The Energy-Efficient Curve

For a digital circuit under a fixed supply voltage V_{DD} and whose last stage is loaded with a capacitance C_L, the "energy-efficient curve" (EEC) is made up by the design points exhibiting the minimum delay for a fixed energy dissipation or, equivalently, the minimum energy consumption for a fixed delay [18,19]. By definition, other design points above the EEC lead to a needlessly higher energy under the same speed performances, as shown in Fig. 1.5.

As previously stated, we adopt the convention of considering the input capacitance of (the first stage of) the circuit, C_{IN}, as a further design variable to be optimized, and including (excluding) the energy dissipated in charging/discharging C_{IN} (C_L).

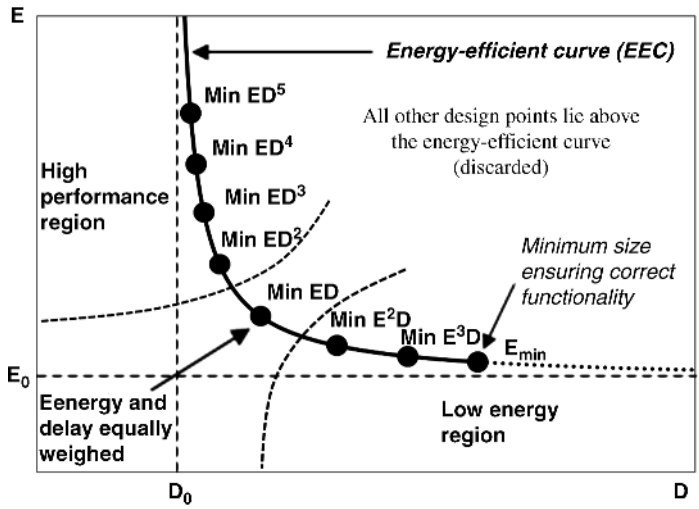

Figure 1.5. Energy-efficient curve and designs optimizing the metrics $E^i D^j$.

This assumption is different from that adopted in [7,20–22] and, while it was a simple matter of convention when referring to the modeling of the energy of a circuit, we will show that it becomes a necessary care when the target is the full exploration of the *E–D* potentials of a topology.

In [19] it was predicted that the EEC of any circuit has a hyperbolic shape

$$(E - E_0)(D - D_0) = E_0 D_0 \tag{1.36}$$

being E_0 and D_0 the minimum energy and minimum delay asymptotes, respectively, as shown in Fig. 1.5. Actually, substantial deviation from (1.36) are found when analyzing real circuits and hence a correction factor γ (typically $0 < \gamma < 1$) can be introduced to fit real data [20,21]

$$(E - E_0)(D - D_0) = \gamma E_0 D_0 \tag{1.37}$$

Despite our assumptions of including the dissipation related to a fully optimizable C_{IN} and excluding that relative to the load C_L differ from those in [20,21], the general character of (1.37) is retained. In particular, looking at the generic EEC depicted in Fig. 1.5, one has that:

1. There is a minimum energy value, $E_{\min}$, that is achievable with the minimum transistors sizes allowing correct operation. This implies that in an extrapolated EEC, the points between E_0 and $E_{\min}$ have not a physical correspondence, as shown in Fig. 1.5.
2. Regarding delay, the value D_0 can be approached only asymptotically through transistor sizing, and measures the maximum speed potential of a specific topology. More specifically, one can indefinitely trade energy for delay by increasing C_{IN}. On the contrary, if C_{IN} is fixed [7,20–22], a minimum delay for a given load is actually reachable and corresponds to the Logical Effort sizing. Nevertheless, also the asymptotic value D_0 under a varying C_{IN} can be estimated through Logical Effort, and it is the parasitic delay P.

As concerns parameter γ in (1.37) and the actual analytical expression of the EEC under our assumption, analytical calculations can be carried out only for a single logic gate [16].

Indeed, according to Logical Effort model, one has

$$\frac{D - D_0}{D_0} = \frac{gh}{p} = \frac{g}{p}\frac{C_L}{C_{\text{IN}}}. \tag{1.38}$$

As concerns the energy, by adopting the approximation in (1.28) one has

$$\frac{E - E_0}{E_0} = \frac{\chi C_{\text{IN}} - \chi C_{\text{IN, min}}}{\chi C_{\text{IN, min}}} = \frac{C_{\text{IN}} - C_{\text{IN, min}}}{C_{\text{IN, min}}} \tag{1.39}$$

being $C_{\text{IN,min}}$ the minimum input capacitance of the gate (i.e., when its transistors are all minimum sized).

By referring to (1.37) and using (1.38), (1.39), the resulting expression for γ is

$$\gamma = \frac{gC_L}{pC_{\text{IN, min}}} - \frac{D-P}{P} = \frac{gC_L}{D_0E_0} - \frac{D-D_0}{D_0} \tag{1.40}$$

The above formula indicates that, under our assumptions, formula (1.37) can be applied with a value of γ that is dependent on the variable D, that is to say the EEC is not a pure hyperbole. However, γ can be approximated in a sufficiently accurate way by its first term, $gC_L/pC_{\text{IN,min}}$ as long as the delay is not much higher than $D_0 = p$.

Nevertheless, when dealing with circuits made up by more than one gate, no analytical expression can be determined for γ, and, in such a case, it is consistent to assume γ as a constant parameter in (1.37).

1.3.2 Energy–Delay Metrics and Hardware Intensity

In the last two decades digital circuit designers have become familiar with the use of composite energy–delay metrics to effectively translate the more and more stringent constraints on the speed performances while not disregarding the energy dissipation.

The first (and at first glance the most appropriate) composite metric to be introduced is the simple ED product, which equally weighs the two quantities. Another popular metric is the ED^2 product where speed has priority over energy. The latter metric is claimed to have useful properties such as a nearly zero sensitivity on the supply voltage [23].

However, although designs optimizing (i.e., minimizing) the above metrics are maximally efficient for a given delay (or energy), it is clear that a generalization is required when analyzing and/or designing a circuit over the entire spectrum of the delay (energy) values it can achieve.

Hence, the general class of metrics E^iD^j, or equivalently ED^η (being η equal to j/i) as originally presented in [19], are introduced. By varying the exponents $i \geq 0$ and $j \geq 0$ ($\eta \geq 0$), any tradeoff between energy and delay can be explored. The extreme cases are obtained when $j/i = 0$ ($\eta = 0$) and when $j/i = \infty$ ($\eta = \infty$), which, once optimized, represent the designs having the minimum possible energy and delay, respectively.

Turning back to the EEC introduced before, one has that a design solution minimizing a metric E^iD^j (ED^η), lies in the EEC [19], that is, this curve is made up of all points that minimize E^iD^j (ED^η), for some i and j (η), as shown in Fig. 1.5.

The demonstration of this assertion is quite simple and intuitive. Indeed, considering a circuit under a fixed load and supply voltage, both its delay and energy are functions of its sizing W (W is an array containing the sizes of transistors in all circuit gates). A design minimizing an E^iD^j metric for some (i, j) has a delay D^* which is obtained with a certain size W^* (i.e., $D^* = D(W^*)$). Since the size W^* minimizes a product E^iD^j, in which the energy is taken into account with $i \geq 0$, the value $E^* = E(W^*)$ of this design will be the minimum among all the designs exhibiting a delay $D = D^*$ and thus it lies on the EEC. More rigorous analytical proofs can be found in [19].

From the above considerations, the indexes i and $j(\eta)$ identify cost functions for optimizing hardware under a fixed load and supply voltage, and, according to [20,21,24], the value j/i (η) is defined "hardware intensity." Basically, $j/i(\eta)$ quantifies the effort to be spent in sizing a circuit to optimize the speed of the circuit at the expense of its energy consumption. The higher $j/i(\eta)$, the higher the effort to further optimize speed. The region of the E–D design space where metrics with $j>i(\eta>1)$ are minimized is hence called the high-performance one, while the region where metrics with $j<i(\eta<1)$ are minimized is called the low energy one. The former is featured by lower and lower delay gains achieved at the cost of larger and larger increments in energy as long as the delay itself diminishes. Analogous considerations are valid for the low energy region.

The graphical interpretation of hardware intensity is shown in Fig. 1.6 [21,24]. The solid line plots a typical EEC for a generic circuit. Dotted curves show several contours of the cost function $E^i D^j$ for three values of the hardware intensity. The point in the E–D space at which the EEC tangents the lowest of the contours corresponds to the energy-efficient implementation of the circuit for that specific hardware intensity value [20,21].

Accordingly, the analytical interpretation of hardware intensity is related to the energy-to-delay sensitivity evaluated in correspondence of the design points optimizing the $E^i D^j (ED^\eta)$ metrics [16,20,21].

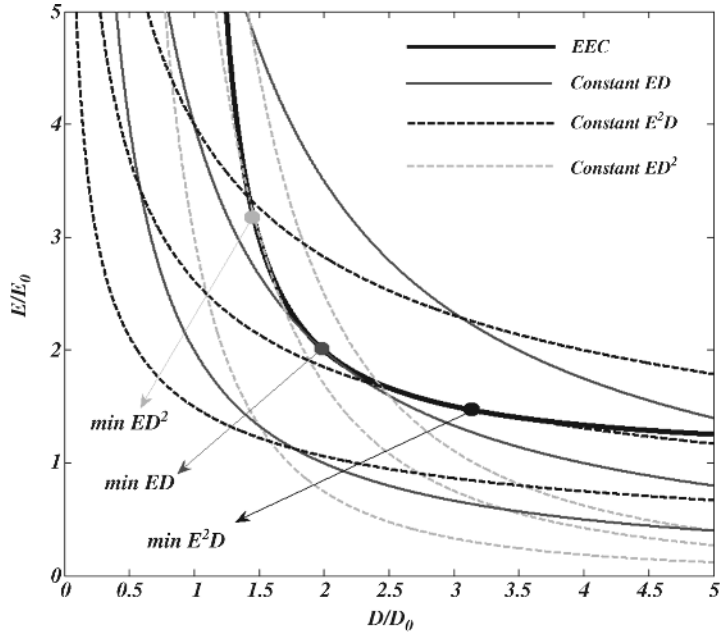

Figure 1.6. Typical energy-efficient curve and constant cost function contours for $j/i = 1.0$, $j/i = 0.5$, and $j/i = 2.0$.

Indeed, by referring to the former ones, the design point minimizing $E^i D^j$ for a given (i, j) leads to a zero derivative of $E^i D^j$ with respect to D and E [16,19]

$$\frac{\partial \left(E^i D^j\right)}{\partial D}_{E^i D^i_{\min}} = \left(iE^{i-1}D^j \frac{\partial E}{\partial D} + jE^i D^{j-1}\right)_{E^i D^j_{\min}} = 0 \tag{1.41}$$

$$\frac{\partial (E^i D^j)}{\partial E}_{E^i D^j_{\min}} = \left(iE^{i-1}D^j + jE^i D^{j-1}\frac{\partial D}{\partial E}\right)_{E^i D^j_{\min}} = 0. \tag{1.42}$$

Solving the set of Eqs. (1.41) and (1.42), one finds

$$S^E_{D_{E^i D^j_{\min}}} = \left(\frac{\partial E}{\partial D}\frac{D}{E}\right)_{E^i D^j_{\min}} = -\frac{j}{i}. \tag{1.43}$$

When carrying out analogous calculations by referring to the ED^η metrics, the result is simply $-\eta$. Anyhow, the adoption of the two indexes i and j allows for better clarifying the E–D tradeoff when the generic $E^i D^j$ FOM is minimized. Indeed, in the neighborhood of the optimum $E^i D^j$ design, a j% speed increase is traded for a i% energy increment and vice versa. Finally, from (1.43) it is apparent that metrics leading to the same j/i ratio are not distinguishable.

1.3.3 Voltage Intensity and Generalization of the Sensitivity Criterion

So far we have focused on hardware, that is, transistors sizing, optimization. However, other tuning variables, such as the supply voltage V_{DD} and the transistors threshold voltages, are available in the circuital level design.

As concerns supply voltage, by introducing the dimensionless derivatives of energy and delay with respect to V_{DD}, henceforth referred as v,

$$E_v = \frac{v}{E}\frac{\partial E}{\partial v} \tag{1.44}$$

$$D_v = -\frac{v}{D}\frac{\partial D}{\partial v} \tag{1.45}$$

and taking their ratio, one can define "voltage intensity," θ, as the energy-to-delay sensitivity relative to the variation of v at a fixed hardware intensity η (i.e., j/i) [20,21]. Hence, just like η represents the negative energy (delay) relative gain at the cost of a relative increase in delay (energy), achievable by restructuring hardware, that is, sizing w, under a fixed v [20,21,26]

$$\eta = -\frac{\partial E}{\partial D}\frac{D}{E}\Big]_{w\text{ variable}-v\text{ fixed}} \tag{1.46}$$

analogously, θ represents the energy (delay) relative increase (decrease), achievable by increasing v under a fixed w [20,21,26]

$$\theta = -\frac{\partial E}{\partial D}\frac{D}{E}\Bigg]_{v\ \text{variable}-W\ \text{fixed}} \tag{1.47}$$

The E_v and D_v values cannot be simply determined through classical $E \propto V_{DD}^2$ and $D \propto 1/(V_{DD} - V_{TH})^2$, given the impact of leakage and short-circuit currents on energy and the complexity of $I_D = f(V_{GS}, V_{DS})$ relationship featuring nanometer transistors. Therefore, it is necessary to develop comprehensive models of energy and delay as functions of the V_{DD} value [25] (similarly to those relative to transistors sizing that were discussed in the previous section) or extract E_v and D_v for the various gates in a circuit through simulations. To have an idea of the main trend, according to experimental results [21], the voltage intensity θ almost linearly increases with V_{DD} for typical CMOS circuits.

The most important aspect of this discussion is that hardware and voltage intensities are related when optimizing a circuit in the *E–D* space.

If we consider a circuit (like a pipeline stage) that has to satisfy a given maximum delay constraint, such a requirement can be achieved at different combinations of the η and θ values. However, the energy-efficient implementation, that is, that with the minimum energy, is the one featured by

$$\eta = \theta. \tag{1.48}$$

Indeed, energy and delay are functions of the variables (w, v), and, by solving the problem of minimizing $E(w, v)$ under the constraint $D(w, v) = D^*$, one finds [16,20,21,26]

$$\frac{\partial D}{\partial w}\frac{\partial E}{\partial v} = \frac{\partial D}{\partial v}\frac{\partial E}{\partial w} \tag{1.49}$$

which means $\eta = \theta$. Hence, for an optimal balance between the supply voltage and the transistors sizing, the relative speed gain achieved at the cost of a given relative energy increase due to an increment in the supply voltage must equal the relative speed gain achieved at the cost of a given relative energy increase due to a larger transistors sizing [21]. This result disproves the common misconception that the lowest energy can be achieved by designing circuit for the highest speed and then reducing the power supply up to the lowest value that satisfies the delay requirement [21].

Further generalizing the above analysis to any kind of design variable, for example, like threshold voltages [27,28], and to the sensitivity of energy to delay with respect to a change in that variable, as in (1.46) and (1.47), the minimum energy under a given delay constraint is achieved when [22]

$$S_x(X) = \frac{\partial E}{\partial D}\frac{D}{E}\Bigg]_{x\ \text{variable}} = S_y(Y) = \frac{\partial E}{\partial D}\frac{D}{E}\Bigg]_{y\ \text{variable}} \quad \forall x,\ y \tag{1.50}$$

being x and y design variables, that is, the energy-efficient corresponds to the design with $x = X$, $y = Y$, and so on.

1.4 ENERGY-EFFICIENT DESIGN OF DIGITAL CIRCUITS

In this section we discuss practical optimization techniques to achieve the energy-efficient design of digital circuits at the circuit level, by considering various levels of complexity. In particular, we first provide some preliminary remarks concerning the role played by the input capacitance of the circuit and the definition of design space bounds, both essential regardless of the actually employed optimization technique. Then, we consider the case of simple basic blocks whose complexity allows a simulations-based optimization and end with large designs that can be dealt with by resorting to convex optimization and exploiting simple *E–D* models.

1.4.1 The Role of the Input Capacitance

As shown in recent works [7,26], when dealing with the issue of energy-efficient design, the input capacitance, C_{IN}, of a logic circuit cannot be simply assumed as fixed. Granted that the adopted C_{IN} value is also related with the architectural-level design strategies [26], differently from [7,20–22], here we consider C_{IN} (i.e., the transistors sizes determining its value) as an additional design variable to be fully optimized like all the other transistors sizes. Indeed, an effective exploration of the *E–D* space to achieve the required *E–D* tradeoff strongly depends on C_{IN}.

A second assumption, differently to [7,20–22,26] is that of including the energy dissipated in the charge and discharge of the C_{IN} and to exclude the energy dissipated in the charge/discharge of the external output load, C_L. Indeed, the first term is inherently related to the adopted circuit sizing (here C_{IN} is a further design knob), whereas the latter term does not depend on the features of the topology [29,30].

It is worth opportunely addressing the consequences of the C_{IN} optimization within a wide range of exploration [18]:

- In general, a throughput increment can be achieved by means of an increase in the degree of parallelism and/or a more critical sizing of all the gates in the logic paths (e.g., when the serial part of code is dominant and parallelization is not so effective). In the latter case, if C_{IN} is increased with respect to medium values, it means that the topology is being sized to achieve a high speed (increasing the energy consumption). Even if the circuit imposes a larger load on the preceding logic stage (e.g., in a pipeline), in high-speed applications the speed penalty of the preceding logic stages could be exceeded by the speed improvement in the considered topology. This tradeoff cannot be explored if one does not assume a fully variable C_{IN}.
- Conversely, when sizing to achieve low-power, low-speed operation, C_{IN} can be strongly reduced. Indeed, granted that the above tradeoff is still valid, the low-power applications are typically featured by long cycle times and hence can easily tolerate slower stages and high logic depths (e.g., when no parallelization is adopted and the processing is actually done serially through single deep paths). In such a context, a slower topology can be tolerated in favor of its smaller energy dissipation.

Obviously, there always exist practical limits on the adoptable C_{IN} values. Nevertheless, once the full EEC is extracted, the designer can easily select the portion of interest according to practical constraints in terms of maximum allowed C_{IN}.

Finally, it is worth highlighting that, when referring to the "first stage," we mean the first gate in the path of the circuit whose sizing is assumed as a reference in terms of timing criticality. Indeed, several input-to-output paths coexist in a circuit composed by more than a single gate and, being the delay of the circuit identified with the maximum among the delays in its various input-to-output paths, the target must obviously be that of equaling these delays. Among the various paths, it is then possible to identify one (typically the longest) that can be used as the reference to identify the C_{IN} of the circuit. Note that, since C_{IN} is fully varied and the optimization targets the equality of the various concurrent delays, the input capacitances of the first stages of all the other paths will be optimized and fully explored as well.

1.4.2 Definition of Design Space Bounds

Regardless of the methodology actually employed for EEC extraction, one first needs to define practical design space bounds allowing one to limit the space of solutions. As will be shown successively, this issue is particularly important in the case of simulations-based procedures and nonlinear optimizations. In these cases, a larger and larger computational effort is required if the design space bounds are not properly defined. On the contrary, this issue becomes less relevant when one adopts simple *E–D* models leading to a convex optimization problem.

At the same time, one must be sure to catch the optimum sizings actually leading to the desired energy–delay tradeoffs, that is, one must guarantee that the selected bounds strictly contain the searched optimum sizings.

In [26] it is shown that Logical Effort designs lie above the EEC, that is, they are not the most efficient possible designs. Even if, unlike [26], the C_{IN}-related dissipation is here included and C_{IN} is assumed as a design variable, the same result still holds[1]. Nevertheless, the energy-to-delay sensitivity of Logical Effort designs can be exploited to determine design space bounds.

More specifically, one can be interested in the portion of the EEC up to a certain minimum-E^iD^j design point with $j/i = X$, that is, the portion of the EEC made up by energy-efficient designs that minimize FOMs with j/i less or equal than X[2]. In such a case, the design bounds can be defined through the "limiting" Logical Effort sizing

[1] Indeed, as explained in the previous paragraph, the minimum energy under a given speed constraint is reached when the sensitivity with respect to "all" the tuning variables is the same. Logical Effort designs are featured by an infinite energy-to-delay sensitivity with respect to the sizes of internal transistors (since delay cannot be further reduced given a fixed C_{IN}), but not with respect to the size of transistors defining C_{IN}. Hence, the condition in (1.50) is not satisfied for Logical Effort designs, which thus are not energy-efficient [26]. Only when C_{IN} approaches infinity, the Logical Effort design will be featured by an equal (and infinite) energy-to-delay sensitivity with respect to all the tuning variables.

[2] It is worth noting that if the searched X is not large enough (say, smaller than 3), the bounds determined through Logical Effort will not be much close to the minimum-E^iD^j design with $j/i = X$.

exhibiting an energy-to-delay sensitivity with respect to C_{IN} equal to $X.$, that is, the upper bound of C_{IN}, $C_{\mathrm{IN,max}}$, is the value which satisfies [16]

$$S_D^E]_{C_{\mathrm{IN}}} = \frac{S_{C_{\mathrm{IN}}}^E}{S_{C_{\mathrm{IN}}}^D} = -\frac{j}{i} = -X. \tag{1.51}$$

The definition of $C_{\mathrm{IN,max}}$ also leads to the definition of the upper bounds for the other design variables (i.e., transistors sizes) that are determined by the Logical Effort sizing with $C_{\mathrm{IN}} = C_{\mathrm{IN,max}}$.

The sensitivity in (1.51) can be analytically evaluated thanks to the property of Logical Effort designs. In particular, as discussed in Section 1.2, given C_{IN} and C_L, the optimized delay D_{TOT} of a circuit simply made up by a path of N cascaded gates is

$$D_{\mathrm{TOT}} = N\sqrt[N]{GBH} + P = N\sqrt[N]{F} + P \tag{1.52}$$

which can be rewritten as

$$D_{\mathrm{TOT}} = p\,(1+k) \tag{1.53}$$

where

$$k = \frac{N\sqrt[N]{GB}\sqrt[N]{C_L}}{p\sqrt[N]{C_{\mathrm{IN}}}} \tag{1.54}$$

is the relative delay increment with respect to the ideal and practically inaccessible minimum path delay (i.e., the path parasitic delay P).

From (1.53) and (1.54), the sensitivity of the optimized path delay, D_{TOT} to C_{IN}, is given by

$$S_{C_{\mathrm{IN}}}^{D_{\mathrm{TOT}}} = \frac{\partial D_{\mathrm{TOT}}}{\partial C_{\mathrm{IN}}}\frac{C_{\mathrm{IN}}}{D_{\mathrm{TOT}}} = -\frac{1}{N}\frac{k}{k+1} \tag{1.55}$$

which is a function of the only C_{IN}.

As for the delay D_{TOT}, it is possible to univocally determine the energy E_{TOT} of a single path circuit sized through Logical Effort for a given C_{IN} and C_L. According to (1.27) and (1.28), the input capacitance, C_N, and the energy, E_N, of the Nth gate are respectively given by

$$C_N = \frac{g_N b_N \sqrt[N]{C_{\mathrm{IN}}}}{\sqrt[N]{GBC_L}} C_L \tag{1.56}$$

$$E_N = \chi_N C_N. \tag{1.57}$$

By iterating the above reasoning and going backward through the path, one finds that the input capacitance and energy of the ith gate (for the Logical Effort design) are

$$C_i = \frac{\left(\prod_{j=i}^{N} g_i\right)\left(\prod_{j=i}^{N} b_i\right)(C_{\mathrm{IN}})^{\frac{N-i+1}{N}}}{(GBC_L)^{\frac{N-i+1}{N}}} C_L \tag{1.58}$$

$$E_i = \chi_i C_i \tag{1.59}$$

and $C_1 = C_{\mathrm{IN}}$.

Therefore, the overall dissipation of the reference path is

$$E_{\mathrm{TOT}} = \sum_{i=1}^{N} \left[\chi_i \frac{\left(\prod_{j=i}^{N} g_i\right)\left(\prod_{j=i}^{N} b_i\right)(C_{\mathrm{IN}})^{\frac{N-i+1}{N}}}{(GBC_L)^{\frac{N-i+1}{N}}} C_L \right]. \tag{1.60}$$

Although one cannot attain to a simple expression like (1.55), also the sensitivity of the overall energy E_{TOT} to C_{IN} can be again expressed as a function of the only C_{IN}

$$S_{C_{IN}}^{E_{\mathrm{TOT}}} = \frac{\sum_{i=1}^{N} \left[\chi_i \frac{\left(\prod_{j=i}^{N} g_i\right)\left(\prod_{j=i}^{N} b_i\right)(C_{\mathrm{IN}})^{\frac{N-i+1}{N}}}{(GBC_L)^{\frac{N-i+1}{N}}} \left(\frac{N-i+1}{N}\right) C_L \right]}{\sum_{i=1}^{N} \left[\chi_i \frac{\left(\prod_{j=i}^{N} g_i\right)\left(\prod_{j=i}^{N} b_i\right)(C_{\mathrm{IN}})^{\frac{N-i+1}{N}}}{(GBC_L)^{\frac{N-i+1}{N}}} C_L \right]} \tag{1.61}$$

Finally, (1.55) and (1.61) can be combined to evaluate (1.51) and determine $C_{\mathrm{IN,max}}$.

Unfortunately, formula (1.51) cannot be always applied straightforwardly given that g_i, h_i, b_i, and p_i are often not available in a closed-form as functions of C_{IN}.[3] Rather, g_i, h_i, b_i, and p_i themselves can be found only by numerically solving a set of complex nonlinear equations when applying the Logical Effort method for a given C_{IN} (see footnote 3).

Furthermore, when the circuit is not simply made up by a single path, also the energy of the circuit is not simply that in (1.60) (see footnote 3), and it is not always possible to find closed form relationships describing the energy of the other gates as functions of C_{IN}. Nevertheless, one has to keep in mind that, when sizing for maximum speed, the energy still depends on the only variable C_{IN}.

Therefore, the need for iterative procedures arises. For instance, one can adopt the following cycle for increasing C_{IN} [16,18]:

[3]There are three main reasons for this issue.

1. The various sources of nonlinearities listed in the second paragraph, which implies the need for iterative procedures to be solved to determine the Logical Effort sizing.
2. The fact that not all the transistors in the circuit have to be considered as variables to be optimized. Actually, only transistors lying in input-to-output paths should represent variables to be optimized in the E–D space, since they affect both consumption and speed. On the contrary, there can exist some parts of the circuit whose size must be simply the minimum one guaranteeing a correct operation, since they affect only energy. This is the case for instance of keepers, pulse generators, and so on. However, these gates have a size dependent on the design variables to be optimized (to guarantee the correct operation) and hence affect b_i in a nonlinear way.
3. The possible presence of reconvergent paths or multiple outputs. Indeed, transistors in the paths that lie nearby to the path assumed as the reference one will affect speed too, since, as previously explained, they must be sized so that all concurrent paths exhibit the same delay (for this reason, their sizes must be considered as design variables to be optimized in the E–D space exploration successive to the definition of design space bounds). When formulating the Logical Effort equations, besides satisfying (1.26) for stages in the reference path, additional equations arise that are relative to the equality of the various concurrent paths delays. This makes the problem of finding the minimum delay design even more complex and nonlinear.

(a) under the current C_{IN} (re)apply the Logical Effort method to find the transistor sizes leading to the minimum delay of all the concurrent paths in the circuit (a nonlinear set of equations must be solved, see note 3);
(b) (re)simulate energy and delay;
(c) (re)extrapolate the E_{TOT} versus C_{IN} and D_{TOT} versus C_{IN} fitted curves and (re)compute the sensitivity (1.51) around the current C_{IN} value;

(re)compare such sensitivity with the desired one $-j/i$. If $\left| S_{D_{TOT}}^{E_{TOT}} \right]_{C_{IN}} | < \left| \frac{j}{i} \right|$, C_{IN} is increased and cycle comes back to (a). Otherwise, cycle stops and $C_{IN,max}$, together with the overall design space bounds, is found.

To exemplify the above procedure, we consider a 4-bit Ripple–Carry Adder in a 65-nm technology, whose schematic is shown in Fig. 1.7, under a load equal to 16 minimum inverters and $V_{DD} = 1V$. In Fig. 1.8, we show the energy-to-delay sensitivity relative to the variation of C_{IN}. The x-axis corresponds to the value of the transistor width w_1 (normalized to the minimum W_{min}) determining the size of the first stage of the circuit, that is, C_{IN}, while other four transistors widths are selected as further tuning variables, $w_2 - w_5$ (see Fig. 1.7 and [16] for details).

By inspection of Fig. 1.8, according to the above-discussed procedure, one has that the minimization of the ED^3 metric requires $w_1 > 15$, while the minimization of the ED^4 metric requires $w_1 > 31$. The corresponding bounds on the other variables $[w_2, w_3, w_4, w_5]$ are [17,18,17,7] for the ED^3 metric and [31,30,25,9] for the ED^4 metric [16]. These bounds are very close to the transistors sizes actually optimizing the two metrics, which are equal to [15,17,16,6] and [29,30,18,10], respectively [16].

Summarizing, these results confirm the effectiveness of such a procedure, which aims at practically bounding the design space through the analysis of the energy-to-delay sensitivity relative to the variation of C_{IN} in minimum delay (i.e., Logical Effort based) designs.

1.4.3 Simulations-Based Optimization of Small Size Circuits

When dealing with small circuits featured by few design variables (i.e., simple basic circuit blocks), the energy-efficient optimization can be carried out by employing a simulations-based procedure, allowing to evaluate both energy and delay with the maximum possible degree of accuracy [16,18,31,32]. Obviously, given that simulations are time consuming, the accuracy in *E–D* estimation is traded for a nonextensive exploration of all the possible design solutions and hence some sort of algorithm have to be applied to reduce the computational effort but still allowing to reach the optimum points.

As a useful consequence of the properties of the $E^i D^j$ metrics discussed in the previous section, from a practical perspective the EEC of a circuit can be extracted by simply minimizing $E^i D^j$ for a limited number of pairs (i, j) and interpolating such optimum points. In particular:

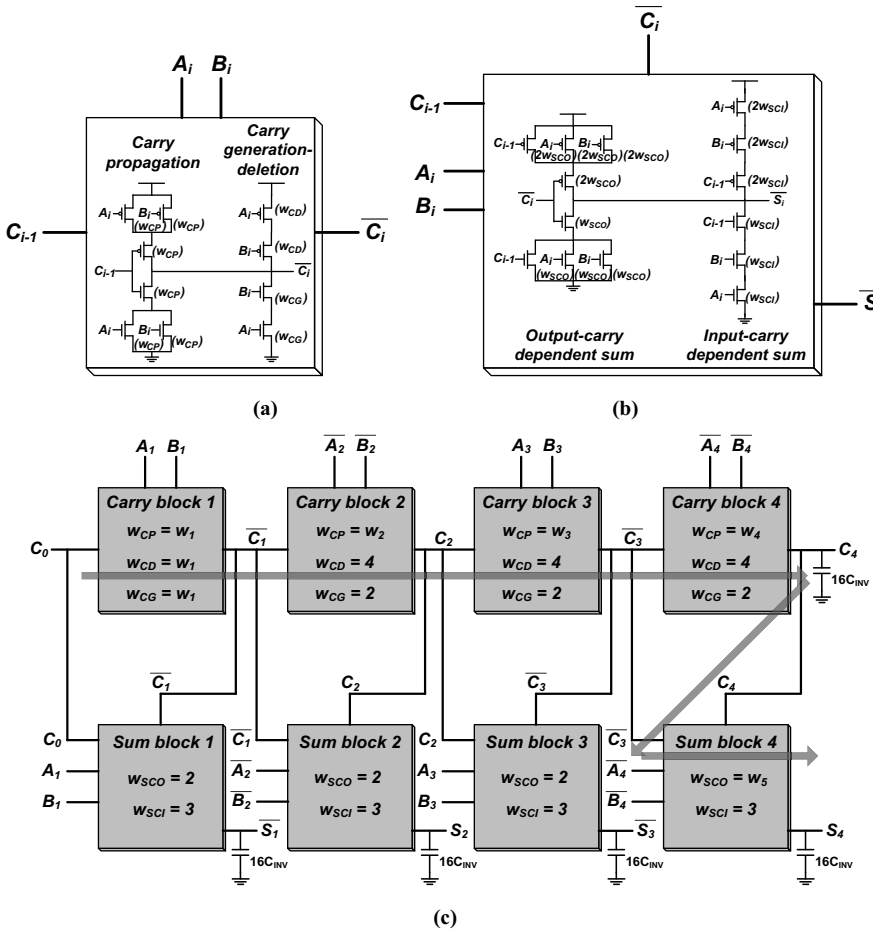

Figure 1.7. Four-bit RCA: carry block (a), sum block (b), whole structure (c).

1. A binary search can be employed to identify minimum-$E^i D^j$ designs because in a simulations-based framework it is worth assuming that $E^i D^j$ functionals are nearly convex in the design space [18]. Anyhow, more complex search criteria can be adopted as well.
2. The design space to be explored can be progressively reduced. Indeed, assuming $j_1/i_1 < j_2/i_2$, a design optimizing $E^{i_1} D^{j_1}$ will be always featured by a sizing smaller than that optimizing $E^{i_2} D^{j_2}$. Therefore, one can start from the metric with an highest j/i ratio, and, once it is optimized with a sizing W', the optimization of the successive (in terms of decreasing j/i value) FOM will be constrained by bounding the design space with the sizing W', and so on [18].

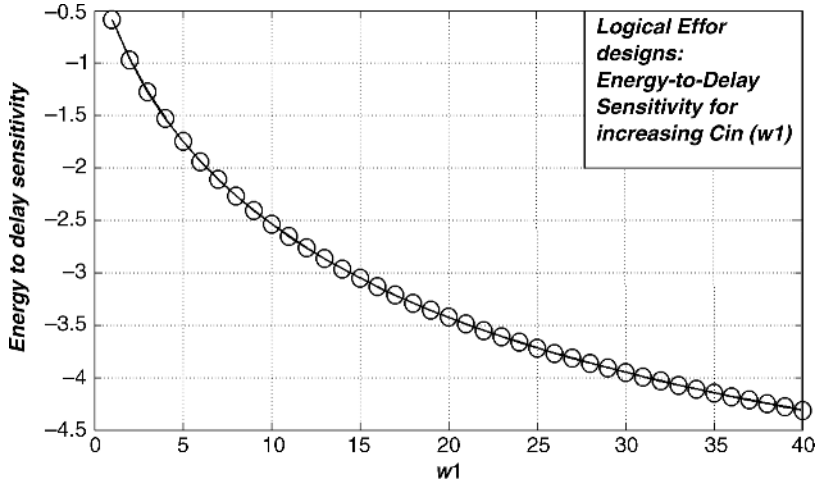

Figure 1.8. Four-bit RCA: energy-to-delay sensitivity of Logical Effort designs as a function of the first stage size.

To exemplify the above search algorithm, we report the results relative to the simulations-based extraction of the EEC for the 4-bit adder previously mentioned. In Fig. 1.9, the design points explored in the search space are depicted with small circles, while the energy-efficient ones minimizing some $E^i D^j$ metrics are highlighted. It is

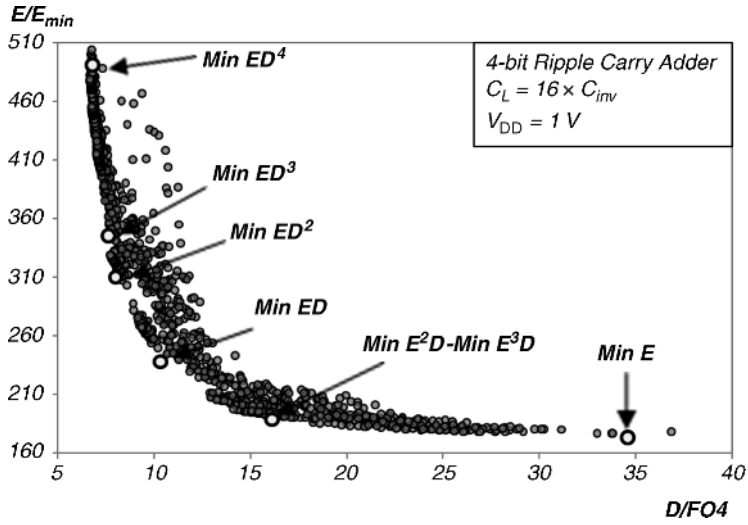

Figure 1.9. Energy-delay space exploration for the 4-bit RCA.

TABLE 1.1. 4-Bit RCA: Minimum E^iD^j Designs

Sizing ↓	D [FO4]	E (E_{min})	$S_D{}^E$	$-j/i$
Min ED^4	6.79	490.89	−4.24	−4.00
Min ED^3	7.62	345.12	−2.78	−3.00
Min ED^2	7.99	310.12	−1.85	−2.00
Min ED	10.32	238.00	−0.92	−1.00
Min $\boldsymbol{E^2D}$	16.11	188.62	−0.37	−0.50
Min $\boldsymbol{E^3D}$	16.11	188.62	−0.37	−0.33
Min E	34.59	173.43	−0.08	−0.00

apparent that the explored designs crowd near the EEC, thus highlighting the search algorithm effectiveness.

As a further validation, we also evaluate the energy-to-delay sensitivity in the minimum E^iD^j points and compare with the theoretically expected $-j/i$ value, as shown in Table 1.1. Results again confirm that the described search algorithm allows one to fairly well identify the minimum E^iD^j points.

1.4.4 Nonlinear and Convex Optimization of Large Size Circuits

When dealing with circuits of large size, that is to say featured by several tens to several thousands design variables, a simulations-based optimization becomes infeasible because of its prohibitive computational effort and a design space exploration based on compact *E–D* models is required.

To give an idea, the full *E–D* space exploration of a simple buffered 2:1 multiplexer, featured by five design variables (transistors widths swept with a W_{min} step), takes nearly a minute on a current desktop computer when using the *E–D* models in Section 1.2 and the previous procedure to determine the design space bounds. The tens of millions designs explored are shown in Fig. 1.10. Considering larger circuits, the complexity grows exponentially and a full exploration soon becomes infeasible.

If the objective function to be minimized (e.g., energy) and constraints functions to be satisfied (e.g., delay related) have not any special feature (e.g., convexity), the optimization problem is said a "nonlinear optimization" or a "nonlinear programming" [33]. This is actually the case when both energy and delay are very accurately modeled by accounting for several effects even in complex ways (e.g., short-circuit currents, impact of input slope on the delay, dependence of leakage on the threshold voltages, etc.).

As long as the design variables are no more than several tens, global optimization algorithms, ensuring that the true global optimum solution is found, can be applied while still maintaining the computational effort feasible, that is, from hours to no more than few days [33]. Obviously, the accuracy in *E–D* estimation is not maximum as in the simulations-based case, but, on the other hand, a much broader exploration of the design space can be performed in a comparable time [34]. Note that in such a case, the definition of proper design space bounds, which can be accomplished by resorting to the previously described method, has still a great importance as in the simulations-based case.

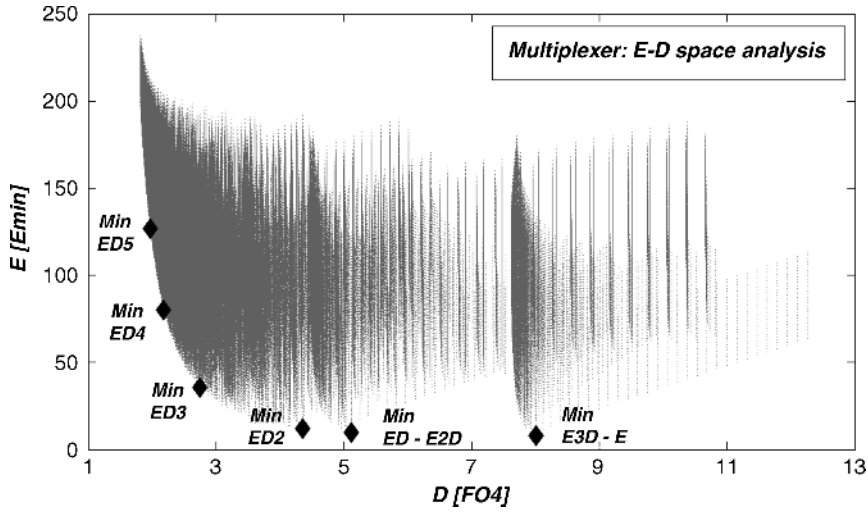

Figure 1.10. Full *E–D* space exploration for a buffered 2:1 multiplexer.

When dealing with circuits featured by more than 100 design variables, a nonlinear programming does no longer allow to reliably determine the optimum solution of the optimization problem. Therefore, the focus must be on the adoption of the most accurate possible *E–D* models leading to optimization problems that can be reliably solved (i.e., assuring the global optimum is found) in a feasible time.

A class of problems that can be reliably and fast solved is the "convex optimization," where both the objective and constraint functions are convex [33]. There is in general no analytical formula for the solution of convex optimization problems, but there are very effective methods for solving them like interior-point methods [33] or other custom methods. For instance, the method proposed in [35] is claimed to size circuit with a million gates in nearly 1 h. Furthermore, thanks to the properties of the above solving methods, the definition of practical design space bounds as well as that of the initial point from which start the optimization, become irrelevant.

Hence, it is apparent that as long as the optimization problem can be formulated in a convex form, the required computational effort is incomparably lower than that required in the previous cases. The other side of the coin is that the formulation itself requires a simplification of the *E–D* models that lowers the accuracy in their estimation. Nevertheless, this is the only feasible approach when the circuit size is large enough.

A class of convex optimization problems that really well suits the problem of digital gate sizing (e.g., to determine the energy-efficient designs as in our case) is that called "generalized geometric programming (GGP)," where the objective and constraint functions take the special form of "generalized posynomials" ("monomials" for the equality constraints). Details and a full mathematical treatment of convex optimization and GGP problems can be found in [33,36].

A comprehensive list concerning the applicability of GGPs to the design of digital circuits can be found in [37]. It includes the following:

- the minimization of energy/power (or area) of logic circuits under speed (e.g., delay, clock frequency) constraints, that is, the energy-efficient design;
- wires sizing in *RC* tree networks;
- statistical optimization under PVT variations.

As previously discussed, energy and delay have to be modeled as most accurately as possible through generalized posynomials. As concerns delay, *RC*-based models linearly including the impact of input slope, as that shown in Section 1.2, are typically adopted [38–40], while energy is typically modeled as proportional to gates sizes, as in (1.28).

1.5 DESIGN OF ENERGY-EFFICIENT PIPELINED SYSTEMS

When dealing with custom datapaths, the design of energy-efficient pipelined systems is essential to achieve the desired throughput (or clock frequency) while paying the lowest possible energy consumption.

Convex optimization methods allow to deal with any kind of digital circuit featured by several concurrent constraints, as in the case of pipelined systems. However, simply formulating the problem as (for instance) a GGP and solving it by relying upon the related mathematics, makes one lose sight of the relevant aspects pertinent to the design of an energy-efficient pipeline. In such sense, the state of the art is represented by the papers from Zyuban and Strenski's [20,21] and a subsequent work [26] drawing inspiration from the former ones and attempting to solve the related issues.

In this section, we refer to pipelines that are made up of pipeline stages (e.g., fetch, decode, execute stages in a processor). In turn, pipeline stages are made up of circuit blocks of different complexity (e.g., a flip–flop, an adder, a multiplier, etc.). Finally, a block is constituted by a number of basic logic gates (e.g., inverters, NAND gates, NOR gates, etc.).

1.5.1 Zyuban and Strenski's Hardware-Voltage Intensity Criteria

According to (1.48), the minimum energy of a single circuit under a given delay constraint is achieved when hardware, η, and voltage, θ, intensities are equal. The analysis can be extended to the cases of:

(a) A composite pipeline stage made up of several blocks (see Fig. 1.11a). The speed constraint is expressed in terms of the overall stage delay, as in the case of a single circuit. However, here we are separately targeting the energy and delay contributions from the various underlying blocks.
(b) A multistage pipeline with composite stages (see Fig. 1.11b), that is, various pipeline stages subject to the same delay constraint.

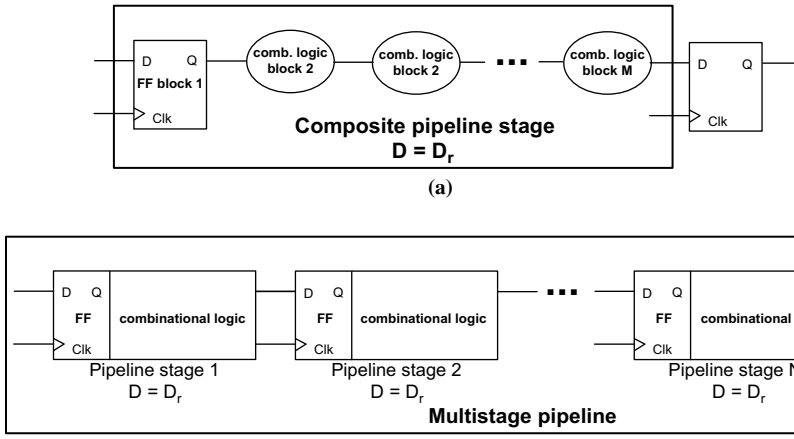

Figure 1.11. Composite pipeline stage (a) and multistage pipeline (b).

(c) A multistage pipeline with composite stages, that is, various pipeline stages subject to the same delay constraint, where the energy and delay contributions from the various underlying blocks are separately targeted.

1.5.1.1 A Composite Pipeline Stage. In any conventional pipeline, at least two independent blocks (latches and logic) can be distinguished, and these are usually designed and tuned independently of each other. Consequently, different blocks in the same pipeline stage may have different values for the optimal hardware intensity.

Assuming the pipeline stage is made up of M blocks, we have to minimize the overall energy

$$E(w_1, w_2, \ldots, w_M, v) = \sum_{i=1}^{M} E_i(w_i, v) \tag{1.62}$$

being w_i the sizes of the various blocks and v the supply voltage, under the constraint that the overall delay is equal to a given value

$$D(w_1, w_2, \ldots, w_M, v) = \sum_{i=1}^{M} D_i(w_i, v) = D_r. \tag{1.63}$$

The solution of the problem can be easily found by using Lagrange multipliers [26], and corresponds to the condition

$$\frac{e_i}{d_i}\eta_i = \theta, \quad \forall i = 1, \ldots, M \tag{1.64}$$

where $e_i = E_i/E$ and $d_i = D_i/D$ are the energy and delay percentages of the ith block relative to the entire pipeline stage, η_i is the hardware intensity of the ith block, and θ is

the stage voltage intensity, that is,

$$\eta_i = -\frac{\partial E_i}{\partial D_i}\frac{D_i}{E_i}\Bigg]_{w_i \text{ variable}/(w_1, w_2, \ldots, w_{i-1}, w_{i+1}, \ldots, w_M, v)\text{fixed}} \tag{1.65}$$

$$\theta = -\frac{\partial E}{\partial D}\frac{D}{E}\Bigg]_{v \text{ variable}/(w_1, w_2, \ldots, w_M)\text{fixed}}. \tag{1.66}$$

Thus, in a pipeline stage with multiple blocks designed independently, blocks that have lower energy weight and higher delay weight should be designed more aggressively than blocks with lower delay weight and higher energy weight.

The aggregate hardware intensity of the whole pipeline stage cannot be in general related to the hardware intensities of the underlying blocks, given that one has [21]

$$\frac{\partial E}{E} = -\sum\nolimits_{i=1}^{M}\left[\frac{e_i}{d_i}\eta_i\frac{\partial D_i}{D}\right]. \tag{1.67}$$

However, when condition (1.64) is satisfied, from (1.67) one finds that the aggregate hardware intensity of the whole pipeline stage is equal to those of the various blocks, that is,

$$\eta = -\frac{\partial E}{\partial D}\frac{D}{E}\Bigg]_{(w_1, w_2, \ldots, w_M)\text{variable}/v \text{ fixed}} = \frac{e_i}{d_i}\eta_i = \theta, \quad \forall i = 1, \ldots, M \tag{1.68}$$

1.5.1.2 A Multistage Pipeline. Practically, different stages of the pipeline usually have different amounts of complexity, and it would be incorrect to tune all of them for the same value of hardware intensity.

Assuming the pipeline is made up of N stages, we have to minimize the overall energy

$$E(w_1, w_2, \ldots, w_N, v) = \sum\nolimits_{i=1}^{N} E_i(w_i, v) \tag{1.69}$$

being W_i the sizes of the various stages, under the constraint that the delays of the various stages are all equal to a given value

$$D_i(w_i, v) = D_r, \quad \forall i = 1, \ldots, N. \tag{1.70}$$

Note that each ith stage is in turn made up of M_i blocks and hence the sizing W_i should be more properly expressed as

$$W_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,M_i}). \tag{1.71}$$

The solution of the problem can be again easily found by using Lagrange multipliers [26], and corresponds to the conditions

$$\sum\nolimits_{i=1}^{N} e_i\eta_i = \theta, \quad \forall i = 1, \ldots, N. \tag{1.72}$$

The above relationship can be used to reevaluate the choice of the power-supply voltage and the clock-cycle target, and possibly the partitioning of the pipeline into stages.

This time the aggregate hardware intensity of the whole multistage pipeline can be computed from the hardware intensities of the various stages and corresponds to the left side of (1.72) Eq. [21], that is,

$$\eta = -\left.\frac{\partial E}{\partial D}\frac{D}{E}\right]_{(w_1, w_2, \ldots, w_N)\text{variable}/v\text{ fixed}} = \sum_{i=1}^{N} e_i \eta_i \tag{1.73}$$

1.5.1.3 A Multistage Pipeline with Composite Stages. Assuming the pipeline is made up of N composite stages and the ith stage is made up of M_i blocks, we have to minimize the overall energy

$$E\left(w_{1,1}, w_{1,2}, \ldots, w_{1,M_1}, w_{2,1}, w_{2,2}, \ldots, w_{2,M_2}, w_{N,1}, w_{N,2}, \ldots, w_{N,M_N} v\right) = \sum_{i=1}^{N} \left\{ \sum_{j=1}^{M_i} \left[E_{i,j}\left(w_{i,j}, v\right)\right]\right\} \tag{1.74}$$

where the subscripts i and j refer to the ith pipeline stage and to the jth block within it, under the constraint that the overall delays of the various stages are all equal to a given value

$$D_i(w_{i,1}, w_{i,2}, \ldots, w_{i,M_i}, v) = \sum_{j=1}^{M_i} [D_{i,j}(w_{i,j}, v)] = D_r. \tag{1.75}$$

The solution of the problem, as in the previous cases, can be found by using Lagrange multipliers and corresponds to the conditions

$$\frac{e_{i,j}}{d_{i,j}}\eta_{i,j} = \frac{e_{i,k}}{d_{i,k}}\eta_{i,k}, \quad \forall j, k = 1, \ldots, M_i \tag{1.76}$$

$$\sum_{i=1}^{N} \frac{e_{i,j}}{d_{i,j}}\eta_{i,j} = \theta, \quad \forall j = 1, \ldots, M_i. \tag{1.77}$$

Again, the aggregate hardware intensity of the whole pipeline stage cannot be in general related to the hardware intensities of the underlying blocks, given that one has [21]

$$\frac{\partial E}{E} = -\sum_{i=1}^{N} \left\{ \sum_{j=1}^{M_i} \left[\frac{e_{i,j}}{d_{i,j}}\eta_{i,j}\frac{\partial D_{i,j}}{D}\right]\right\}. \tag{1.78}$$

However, when condition (1.76) is satisfied, from (1.78) one finds that the aggregate hardware intensity of the whole multistage pipeline is equal to

$$\eta = \sum_{i=1}^{N} \frac{e_{i,j}}{d_{i,j}}\eta_{i,j}, \quad \forall j = 1, \ldots, M_i. \tag{1.79}$$

1.5.2 Practical Guidelines to Design Energy-Efficient Pipelines

The optimal criteria given by Zyuban and Strenski have two primary limitations: their hard-to-use coarse-tuning approach and the restricted assumption of energy and delay dependency among blocks/stages [26].

Indeed, the optimal criteria are difficult to apply, and their application is mainly suited for the verification of design optimality, since, given a design solution, these criteria can be used to determine if the design is optimal. However, if the design is not optimal, the criteria may suggest modifications to energy, delay, hardware intensity, or supply voltage, but it is not immediately clear how to change each of these quantities [26].

The other limitation arises since these optimal criteria are derived assuming that changes in a particular block/stage do not affect the energy and delay of neighboring ones. While this assumption can be justified in coarse tuning of circuits, it is generally not true for a pipeline stage where the input (output) capacitances of each stage/block affect the performance of the preceding stage/block (of the stage/block itself). Therefore, the energy and delay dependencies between adjacent blocks/stages should be added to the previous derivations. However, due to the nonanalytical form of these dependencies, their inclusion does not lead to an analytical solution [26].

To partially overcome the above issues, a thorough methodology consisting of several iterative steps has been proposed in [26]. This methodology targets the minimization of the energy of a multistage pipeline under a given delay constraint and without neglecting the mutual influence between the design of the various stages. In this case, the stages are treated as unique blocks, that is, the previous analysis relative to the energy-efficient design of a stage considered as the composition of several blocks is not considered.

The convention adopted in [26] is to exclude the energy dissipation related with the charge/discharge of the input capacitance of a stage and to include that related with the charge/discharge of the output load capacitance.

The iterative procedure leading to the optimum designs of all the combined pipeline stages is based on the optimization of the stages themselves under various input/output capacitances conditions. Indeed, three different optimizations can be performed on a single stage:

1. The stage can be designed to achieve the minimum energy under a given delay constraint and with a fixed input and load capacitances. This is the problem discussed in the rest of this chapter and can be dealt with by resorting to simulations- or models-based optimizations (e.g., with generalized geometric programming). When exploring different delay constraints, an energy-efficient curve can be extracted and it reaches a well-defined minimum delay point corresponding to the Logical Effort design. This case is exemplified in the case of a 64-bit Kogge–Stone adder in Fig. 1.12 [26].
2. Given the convention adopted on input capacitance related dissipation, the delay of the stage can be improved without worsening energy by simply increasing the input capacitance as shown in the case of the 64-bit Kogge–Stone adder in Fig. 1.13 [26]. Obviously, such an increase negatively affects the delay of the stage preceding the considered one given that its load increases.

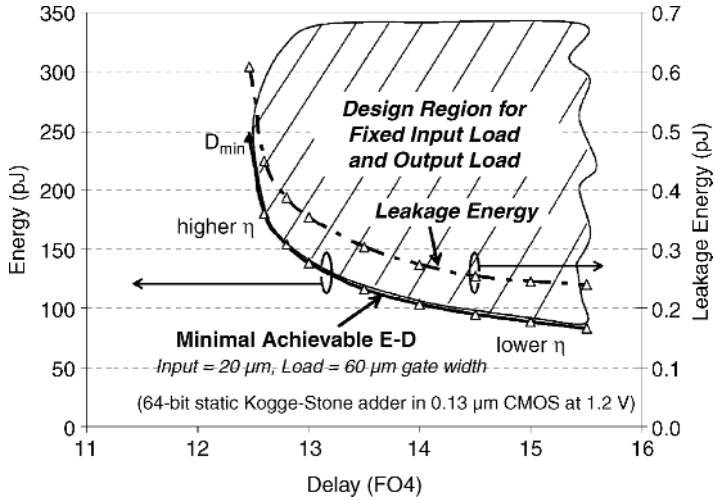

Figure 1.12. 64-bit Kogge–Stone adder: energy–delay optimization under fixed input capacitance and output load (Copyright © IEEE 2006).

3. Given the convention adopted on input capacitance related dissipation, the energy of the stage can be improved without worsening delay by simply increasing the input capacitance as shown in Fig. 1.13 [26]. Indeed, a larger input capacitance allows to reach the same delay with a smaller sizing (and hence a smaller dissipation) of the other gates within the stage. Obviously, such an increase negatively affects the energy of the stage preceding the considered one given that its load increases.

According to (2) and (3), for a fixed output load and a variable input capacitance an energy-efficient design region comes out and, as shown in Fig. 1.13, it is located between the minimum energy and minimum delay points [26]. Given a delay constraint, the maximum and minimum values for the input capacitance are found and correspond to the minimum energy and minimum delay point in Fig. 1.13.

The key for overall energy optimization is the analysis for each stage of the sensitivities of the optimized energy to the input capacitance, C_{IN}, under a fixed output load, C_L, and to the output load under a fixed input capacitance

$$\sigma_{E,\,C_{\mathrm{IN}}} = -\frac{\partial E}{\partial C_{\mathrm{IN}}}\Bigg]_{C_{\mathrm{IN\ variable/C_Z\ fixed}}} \tag{1.80}$$

$$\sigma_{E,\,C_{\mathrm{L}}} = -\frac{\partial E}{\partial C_L}\Bigg]_{C_{L\ \mathrm{variable/C_{IN}\ fixed}}}. \tag{1.81}$$

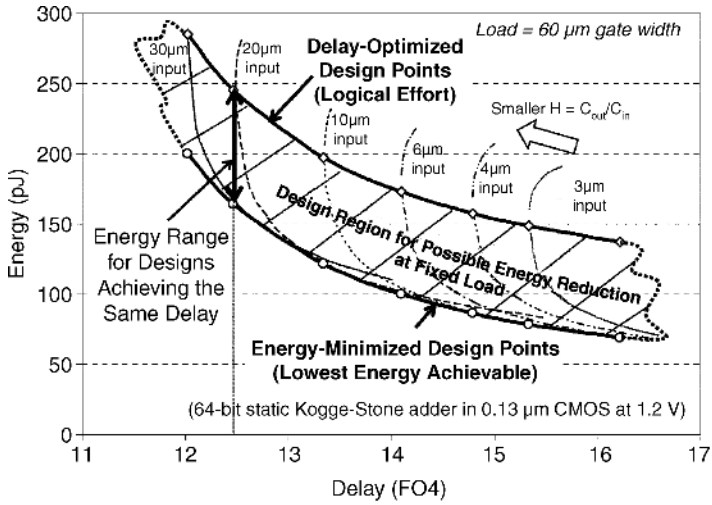

Figure 1.13. 64-bit Kogge–Stone adder: design region for possible energy–delay reduction under varying input capacitance and fixed output load (Copyright © IEEE 2006).

Indeed, in this way one can deal with the improvement of the performance of a stage when increasing its input capacitance and decreasing its output load, and the corresponding decrease in the performance of the preceding and succeeding stages.

The general trends are shown in Fig. 1.14, where it is shown that [26]:

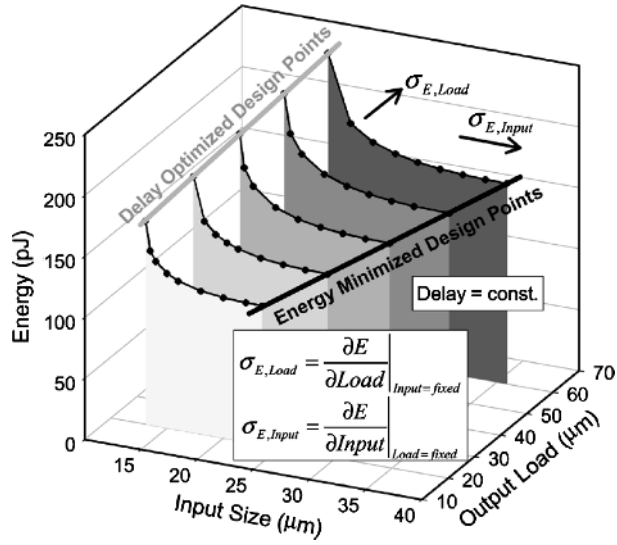

Figure 1.14. Optimized energy of a pipeline stage versus input capacitance under fixed load and versus load under fixed input capacitance (Copyright © IEEE 2006).

- The sensitivity of the optimized energy of the stage to C_{IN} under a fixed C_L asymptotically decreases for larger values of C_{IN} itself. The maximum value of $C_{IN}C_{IN}$ leading to the lowest stage energy corresponds to the minimum energy point in Fig. 1.13 and increases for larger C_L. On the contrary, the sensitivity increases when moving towards the minimum delay point (Logical Effort design) by decreasing the C_{IN} value. Again, the minimum delay point is achieved with a larger $C_{IN}C_{IN}$ when C_L increases.
- The optimized energy of the stage under a fixed $C_{IN}C_{IN}$ is a nearly linear increasing function of C_L both when considering the minimum energy and minimum delay points.

It is easy to show that, when considering a multistage pipeline, the overall minimum energy is reached when the sensitivities of the energy of all stages to their input capacitances and output loads are all equal [26].

Basing on the above considerations, an iterative procedure to determine the energy-efficient sizing of a multistage pipeline comes out [26]:

(a) A set of initial values for the capacitances at the boundaries between the various stages is chosen.
(b) The various stages are optimized for minimum energy given the delay constraint under a fixed input capacitance and output load (the capacitances at the boundary are fixed). This optimization can be performed with any of the methods discussed in this chapter.
(c) The sensitivities in (1.80) and (1.81) are computed for each stage.
(d) If the sensitivities are not equal for all the stages, the values of the capacitances at the boundaries between the various stages are properly updated and the procedure comes back to (b). Otherwise the energy-efficient design for the multistage pipeline is found and procedure ends.

1.6 CONCLUSION

Scaling trends have driven CMOS technology into a so-called power limited regime, where power/energy dissipation has become a prominent aspect and it is no longer possible to focus solely on the optimization of circuit speed. This chapter dealt with the design of energy-efficient digital circuits, that is, with the achievement of the desired speed performances under the minimum energy consumption. Energy-delay models of logic gates and the theoretical background relative to the analysis of circuits in the energy-delay space were discussed, in order to identify the energy-efficient design criteria. Practical guidelines concerning the simulations based optimization of small-size circuits were provided, as well as remarks on the convex optimization of large size circuits. Finally, considerations on the energy-efficient design of pipelined systems were reported.

REFERENCES

1. R. Dennard, F. Gaensslen, V. Rideout, Bassous, and A. LeBlanc, "Design of ion-implanted MOSFETs with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, Vol. 9, No. 5, pp. 256–268, 1974.
2. S. Borkar, "Design challenges of technology scaling," *IEEE Micro 1999*, Vol. 19, No. 4, pp. 23–29, 1999.
3. International Technology Roadmap for Semiconductors, [Online]: http://public.itrs.net/.
4. Y. Taur, "CMOS design near the limit of scaling," *IBM Journal of Research and Development*, Vol. 46, No. 2–3, pp. 213–222, 2002.
5. B. Nikolic, "Design in the power-limited scaling regime," *EEE Transactions on Electron Devices*, Vol. 55, No. 1, pp. 71–83, 2008.
6. A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 4, pp. 473–484, 1992.
7. V. Oklobdzija and R. Krishnamurthy, *"High-Performance Energy-Efficient Microprocessor Design,"* Springer, Berlin, 2006.
8. M. Horowitz,"Timing Models for MOS Circuits," Ph.D. Dissertation, Stanford University, Palo Alto, CA, 1984.
9. J. Rabaey, A. Chandrakasan, and B. Nikolic, "*Digital Integrated Circuits: A Design Perspective* (2nd Edition)," Prentice-Hall, Upper saddle River, NJ, 2003.
10. I. Sutherland, B. Sproull, and D. Harris, *"Logical Effort: Designing Fast CMOS Circuits,"* Morgan Kaufmann Publishers, San Francisco, CA, 1998.
11. Y. Taur and T. Ning, *"Fundamentals of Modern VLSI Devices – 2nd edition,"* Cambridge University Press, Cambridge, 2009.
12. N. Weste and D. Harris, *"CMOS VLSI Design: A Circuits and System Perspective (3rd edition),"* Addison Wesley, Reading, MA, 2004.
13. R. Zlatanovici, Power-Performance Optimization for Digital Circuits, Ph.D. Dissertation, *UC Berkeley*, 2006.
14. J. Rabaey, *Low Power Design Essentials*, Springer, Berlin, 2009.
15. M. Alioto, E. Consoli, and G. Palumbo, "Flip–flop energy/performance versus clock slope and impact on the clock network design," *IEEE Transactions on Circuits and Systems – Part I*, Vol. 57, No. 6, pp. 1273–1286, 2010.
16. M. Alioto, E. Consoli, G. Palumbo, "From energy-delay metrics to constraints on the design of digital circuits," in print on *International Journal of Circuit Theory and Applications*.
17. J. Monteiro, S. Devadas, A. Ghosh, K. Keutzer, and J. White, "Estimation of average switching activity in combinational logic circuits using symbolic simulation," *IEEE Transactions of Computer-Aided Design of Integrated Circuits and Systems*, Vol. 16, No. 1, pp. 121–127, 1997.
18. M. Alioto, E. Consoli, and G. Palumbo, "General strategies to design nanometer flip–flops in the energy-delay space," *IEEE Transactions on Circuits and Systems* – Part I, Vol. 57, No. 7, pp. 1583–1596, 2010.
19. P. Penzes and A. Martin, "Energy-delay efficiency of VLSI computations," *Proceedings of ACM Great Lake Symposium on VLSI*, pp. 104–111 2002.
20. V. Zyuban and P. Strenski, "Unified methodology for resolving power-performance tradeoffs at the microarchitectural and circuit levels," *Proceedings of IEEE International Symposium on Low Power Electronics and Design*, pp. 166–171, 2002.

21. V. Zyuban and P. Strenski, "Balancing hardware intensity in microprocessor pipelines," *IBM Journal of Research and Development*, Vol. 47, No. 5–6, pp. 585–598, 2003.
22. D. Markovic, V. Stojanovic, B. Nikolic, M. Horowitz, and R. Brodersen, "Methods for true energy-performance optimization," *EEE Journal of Solid-State Circuits*, Vol. 39, No. 8, pp. 1282–1293, 2004.
23. A. Martin, "Towards an energy complexity of computation," *Information Processing Letters*, pp. 181–187, 2001.
24. V. Zyuban, "Optimization of scannable latches for low energy," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 778–788, 2003.
25. M. Alioto and G. Palumbo, "Impact of supply voltage variations on full adder delay: analysis and comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 14, No. 12, pp. 1322–1335, 2006.
26. H. Dao, B. Zeydel, and 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, 2006.
27. D. Liu and C. Svensson, "Trading speed for low power by choice of supply and threshold voltage," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 1, pp. 10–17, 1993.
28. R. Gonzalez, B. Gordon, and M. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 8, pp. 1210–1216, 1997.
29. V. Stojanovic and V. Oklobdzija, "Comparative analysis of master–slave latches and flip–flops for high-performance and low-power systems," *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 4, pp. 536–548, 1999.
30. C. Giacomotto, N. Nedovic, and V. Oklobdzija, "The effect of the system specification on the optimal selection of clocked storage elements," *IEEE Journal of Solid-State Circuits*, Vol. 42, No. 6, pp. 1392–1404, 2007.
31. M. Alioto, E. Consoli, G. Palumbo, "Analysis and comparison in the energy-delay-area domain of nanometer CMOS flip–flops: Part I—methodology and design strategies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 19, No. 5, pp. 725–736, May 2011.
32. M. Alioto, E. Consoli, G. Palumbo, "Analysis and comparison in the energy-delay-area domain of nanometer CMOS flip–flops: Part II—results and figures of Merit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 19, No. 5, pp. 737–750, May 2011.
33. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2003.
34. V. Oklobdzija, B. Zeydel, H. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 754–758, 2005.
35. S. Joshi and S. Boyd, "An efficient method for large-scale sizing," *IEEE Transactions on Circuits and Systems* – Part I, pp. 2760–2773, 2008.
36. S. Boyd, S. Kim, D. Patil, and M. Horowitz, "Digital circuit optimization via geometric programming," *Operations Research*, pp. 899–932, 2005.
37. S. Boyd, S. Kim, and S. Mohan, "Geometric programming and its applications to EDA problems," *Proceedings of Design, Automation & Test in Europe Conference*, Tutorial paper, 2005.

38. J. Fishburn and A. Dunlop, "TILOS: A polynomial programming approach to transistor sizing," *Proceedings of IEEE International Conference on CAD*, pp. 326–328 1985.

39. A. Conn, I. Elfadel, W. Molzen, P. O'Brien, P. Strenski, C. Visweswariah, and C. Whan, "Gradient-based optimization of custom circuits using a static timing formulation," *Proceedings of Design Automation Conference*, pp. 452–459 1999.

40. D. Patil and S. Kim,"Stanford Circuit Optimization Tool (SCOT) User Guide," [Online]: www.stanford.edu/class/ee371/tools/SCOT_UserGuide.pdf