Section I

CMOS circuits and technology limits

1 Energy efficiency limits of digital circuits based on CMOS transistors

Elad Alon

1.1 Overview

Over the past several decades, CMOS (complementary metal–oxide–semiconductor) scaling has come to be associated with dramatic and simultaneous improvements in functionality, performance, and energy efficiency. In particular, although the actual historical trends did not uniformly follow a single type of scaling, there was a relatively long period of "Dennard scaling" [1] during which the quadratic (with scale factor) improvements in transistor density were accompanied by a quadratic reduction in power per gate despite a linear increase in switching frequency. All of this was achieved by scaling the operating (i.e., supply) voltage of the circuitry linearly along with the lithographic dimensions of the transistor. Ideally, this would result in constant power consumption per unit chip area, making it relatively easy for chip architects and designers to exploit the increased transistor density with a fixed chip area (and hence power) to cram more functionality into a single die.

Unfortunately, however, as Dennard himself predicted, because of the fact that some intrinsic parameters associated with transistor operation – in particular, the thermal voltage kT/q – do not scale along with the lithographic dimensions, this type of scaling came to an end in the early 2000s. Up until that point, because leakage currents (and hence leakage energy) were essentially negligible, the transistor's threshold voltage had been treated as a scaling parameter that could be reduced with no significant consequence. However, since leakage current depends exponentially on the threshold voltage, this type of scaling indeed eventually came to a halt.

As will be described in detail in Section 1.2, for today's designs (and ever since roughly the 90 nm process technology node), both the threshold and supply voltages must be chosen to balance out the leakage and dynamic energy components at a given desired performance. The implication of this is that simple scaling no longer provides obvious benefits in all three dimensions (density, power, and performance); instead, one is forced to make direct trade-offs between energy and performance – even if given a more lithographically advanced process technology. This section will highlight that at the device level, transistors must achieve an on/off current ratio of ~10^4–10^6 in order to achieve optimal energy efficiency. Section 1.3 next discusses selected techniques – in particular, power gating and parallelism – utilized by architects and circuit designers to achieve the energy-efficiency potential of scaled CMOS technologies. Finally, in Section 1.4 we will highlight the fact that CMOS transistors have a well-defined

minimum energy per operation, and thus even parallelism will eventually cease to be an effective means of keeping chip power consumption in check.

1.2 Energy–performance trade-offs in digital circuits

In order to explain why both the supply and threshold voltages must be balanced to achieve energy-efficient digital circuits, we must first briefly examine the composition of typical digital chips. As highlighted in Fig. 1.1, the largest contributor to the power consumed by a processor (which is a good representative for digital chip designs as a whole) is typically the control/datapath, and in fact, the overall performance and power of the chip generally track with those of the control/datapath as well. As also highlighted by the figure, the clock frequency (performance) of the design is set by the delay of the combinational logic between the clocked registers.

Although there are obviously extremely wide variations in the actual composition of the combinational logic within a digital chip, the behavior (in terms of energy and performance) of all such logic tracks very closely with the behavior of a cascade of inverters. To begin analyzing the underlying trade-offs, we can therefore utilize the simplified model shown in Fig. 1.2 as a proxy for the energy and performance of a generic digital circuit. As highlighted in the figure, the most relevant circuit-level parameters are the activity factor α – which is defined as the average probability of a given node in the circuit transitioning (i.e., changing its state) on any given clock cycle, the capacitive fanout[1] f, the capacitance per inverter (gate) C, and the logic depth (i.e., the number of stages of combinational logic between flip-flops) L_d.

With this model in hand, it is easy to show that the delay t_{delay} of the circuit is simply set by:

$$t_{delay} = \frac{1}{2}\frac{L_d \cdot f \cdot C \cdot V_{dd}}{I_{on}(V_{dd} - V_{th})}, \tag{1.1}$$

where V_{dd} is the power supply voltage of the circuit, and $I_{on}(V_{dd} - V_{th})$ is the effective[2] drain current of the transistors within the inverter when they are in the on-state, driven by a given supply voltage V_{dd}, and with a given threshold voltage V_{th}. One can use a variety of different models to expand the functional relationship between I_{on}, V_{dd}, and V_{th} (e.g., alpha-power law [2], velocity saturation [3], etc.), but as we will see shortly, it is not necessary to do so to understand the underlying causes for the key trade-offs at hand; one must simply realize that the on-current increases if $(V_{dd} - V_{th})$ is increased.

Let us next consider the energy consumed by the chain of inverters during the completion of a single operation. For well-designed digital circuits, the energy will consist essentially of only two components: dynamic energy due to charge/discharging

[1] In this model the fanout may appear logical in that every inverter is driving f copies of itself, but for general digital circuits the fanout should be treated as capacitive – i.e., the ratio of the input capacitance of a given gate to the input capacitance of the succeeding gates in the chain.

[2] The drain current of the devices isn't actually constant during the output transition, but can be well approximated by a single number in most cases of interest.

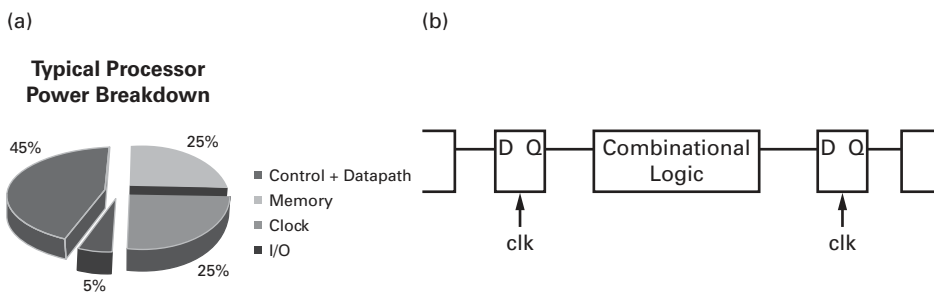

Fig. 1.1 (a) Power breakdown for a typical embedded processor. (b) Conceptual model for synchronous digital circuits.

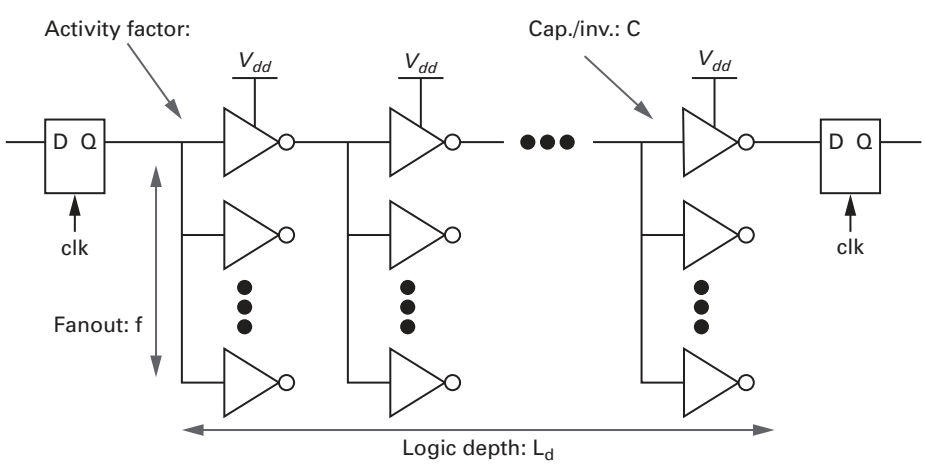

Fig. 1.2 Inverter-based model for combinational logic energy and performance.

the parasitic capacitance within the circuit, and leakage energy due to the fact that even the off switches within the logic gates still conduct current during the entire duration of the operation. Once again referring to the model in Fig. 1.2, the dynamic (E_{dyn}) and leakage (E_{leak}) energy components are:

$$E_{dyn} = \alpha \cdot L_d \cdot f \cdot C \cdot {V_{dd}}^2, \tag{1.2a}$$

$$E_{leak} = L_d \cdot f \cdot I_{off}(V_{th}) \cdot V_{dd} \cdot t_{delay}, \tag{1.2b}$$

where $I_{off}(V_{th})$ is the effective off-state leakage of the transistors within the inverter for a given device threshold voltage V_{th}.[3]

To highlight why one must now choose V_{dd} and V_{th} such that they balance out these two components of energy consumption at a given performance, it is instructive

[3] The supply voltage V_{dd} also affects the leakage current I_{off}, but for the purposes of this discussion this effect does not alter the underlying trade-offs/conclusions.

to combine Eqs. (1.1) and (1.2) as follows into a single expression for the total energy per operation:

$$E_{total} = \alpha \cdot L_d \cdot f \cdot C \cdot V_{dd}{}^2 + L_d \cdot f \cdot I_{off}(V_{th}) \cdot V_{dd} \cdot \frac{1}{2} \frac{L_d \cdot f \cdot C \cdot V_{dd}}{I_{on}(V_{dd} - V_{th})}$$
$$= \alpha \cdot L_d \cdot f \cdot C \cdot V_{dd}{}^2 \cdot \left[1 + \frac{L_d \cdot f}{2\alpha} \cdot \frac{I_{off}(V_{th})}{I_{on}(V_{dd} - V_{th})}\right]. \quad (1.3)$$

The most important point to notice about the expression in Eq. (1.3) is that, although one would like to use a low V_{dd} to reduce energy, one cannot do so without also lowering V_{th} if the same performance (i.e., $t_{delay} \propto CV_{dd}/I_{on}$) is to be maintained, thus increasing the leakage energy. The critical implication of this is that there are optimal V_{dd} and V_{th} values which balance out the two energy components such that the lowest total energy is achieved for a given delay target (or equivalently, the lowest delay for a given energy).

Notice also that the quantity I_{on}/I_{off} when scaled by $L_d f/\alpha$ (which is set purely by circuit-level parameters) is directly indicative of the ratio between dynamic and leakage energy for the whole circuit. In fact, as shown by Nose and Sakurai in [4] for super-threshold CMOS circuits, the optimal I_{on}/I_{off} (and therefore both the resulting optimal V_{dd} and V_{th} as well as the ratio of dynamic to leakage energy) is directly set by $L_d f/\alpha$, and remains relatively fixed regardless of the exact delay target. Furthermore, an analysis by Kam and his co-authors in [5] shows that this result essentially holds true for any CMOS-like device technology in essentially any operating region (i.e., sub- vs. super-threshold), even those with significantly steeper drain current vs. gate voltage than CMOS transistors.

Given the above observations, and in order to provide a numerical guideline for the optimal I_{on}/I_{off}, it is worthwhile to examine representative values for the circuit-level parameters L_d, f, and α, as well as the reasons underlying the selection of those values. Let's begin with the logic depth L_d, which is typically set to ~15–40. Much like the optimal V_{dd} and V_{th}, this selection is driven by balancing out the improved timing slack gained by further pipelining (i.e., reducing L_d) with the increased overhead from additional timing elements (i.e., flip-flops/registers) [6]. Similarly, the fanout f is typically set to greater than 2 to reduce the delay overhead associated with each gate stage and up to ~8 to ensure robust operation (gates with large fanout tend to be much more susceptible to noise/crosstalk). Finally, the overall activity factor α for most practical designs is ~10% down to 0.1%; these relatively low percentages can be understood by the fact that in most complex logic chains (and even more so in memory structures), the large majority of the states of the gates are not changing on any one clock cycle.

Taken together and with the appropriate scale factors, the optimal I_{on}/I_{off} for a wide variety of designs lies within the range 10^4–10^6. Since for reasonable performance levels CMOS transistors achieve ~100 mV/dec effective inverse slope (i.e., $V_{dd}/\log_{10}(I_{on}/I_{off})$, as defined in [5]), the supply voltage necessary to achieve this on/off current ratio is typically 500–600 mV. Note that the farther into the high-performance regime one wants to operate, the worse the effective overall slope will be, and hence many designs operate at closer to 1 V to achieve the desired (peak) performance.

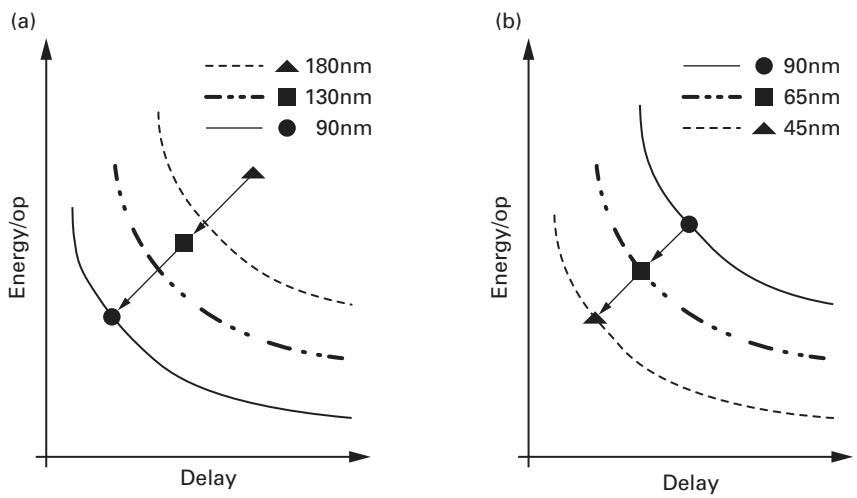

Fig. 1.3 Scaling of designs in the energy per operation vs. delay space using the nominal supply and threshold voltages under (a) traditional (Dennard) scaling and (b) modern (~sub-90 nm) scaling.

Before moving on to the next section, it is worth examining the implications of the above analysis on historical as well as future CMOS scaling. During the traditional (Dennard) scaling regime, simultaneously lowering V_{dd} and V_{th} caused a substantial and dramatic decrease in the I_{on}/I_{off} ratio from one process technology to the next. It turns out, however, that reducing the I_{on}/I_{off} ratio in this way was actually very desirable, because at that point the thresholds had been set so high that the leakage energy component was negligible. It was therefore beneficial to reduce the supply voltage and save on dynamic energy. In other words, the reason that scaling was able to proceed in this manner was that at that point, typical designs were actually not operating on optimal points in the energy vs. delay trade-off space.

To make this perspective clear, Fig. 1.3 uses markers to show where designs operating under the nominal supply and threshold voltages for a given process technology would lie relative to the optimal energy vs. delay curves. As shown in Fig. 1.3(a), typical designs were operating substantially above and to the right of the optimal curves, but as V_{dd} and V_{th} were reduced, scaling brought these designs closer to the actual optimal curves. In other words, a significant portion of the energy-efficiency benefits that came to be associated with scaling were not actually inherently due to the dimensional scaling itself – rather, they were the result of reducing the degree of sub-optimality.

This is of course not to say that dimensional scaling brings no benefits at all in energy and delay – it is simply that once designs were essentially operating on the optimal part of the curve, as highlighted in Fig. 1.3(b), purely dimensional scaling (with V_{dd} and V_{th} fixed) brings at best linear reductions in energy/operation and delay, both due to decreased capacitance/gate [7]. In practice, the poor scaling of interconnect parasitics and variation issues tend to make the capacitance/gate scale relatively poorly (i.e., the minimum total capacitance per gate does not reduce substantially from one process to the next).

Even in the best case, however, simple dimensional scaling does not provide sufficient benefit to enable scaled designs to achieve increased performance and functionality within a given power budget. Specifically, if one leaves the supply and threshold voltages fixed, the power per gate (which is proportional to E_{total}/t_{delay}) is also fixed. Nevertheless, if one actually exploited the increased density to integrate twice as many gates in each process generation, the power of the chip would double as well. In the vast majority of applications chip power must be kept constant from one generation to the next (due either to thermal or battery-life limitations), and thus designers have been forced to utilize other approaches to translate dimensional scaling into usable advances. The most prominent of these approaches – namely, parallelism – will be discussed further in the next section.

1.3 Design techniques for energy efficiency

Since many of the trade-offs between energy and performance discussed in the previous section can be traced backed to the fact that CMOS transistors leak when they are supposed to be off, it is natural to wonder whether a circuit- or system-level technique can be used to eliminate or at least mitigate the leakage energy. The most natural candidate for this is referred to as "power gating" or "sleep transistors" [8]. Figure 1.4 depicts the concept as applied to a chain of inverters, where the key idea is to disconnect an entire block from its power supply during periods of time where one knows that the block is not performing any useful work. The power switch itself must of course also be implemented by some kind of transistor (or more generally, whatever switch is available in the process technology), but if this switch is implemented with a higher I_{on}/I_{off} device (i.e., a device with higher V_{th} and/or larger gate voltage swing), turning this switch off can indeed reduce the leakage of the overall circuit vs. the original circuit in the off-state.

Continuing down the original line of thinking, one may then wonder if power gating could be utilized even more aggressively to cut off the power supply of each gate as soon as it has finished doing useful work, and hence break or at least improve upon the trade-offs described previously. In particular, if the gate was only "awake" whenever its output needs to transition, the activity factor α would effectively be much larger than the numbers quoted earlier. The issue with this idea, however, is that one must know when to turn the power gating switch on or off, and in the limit of power gating every single logic gate separately, one would need to replicate the functionality of the entire gate to

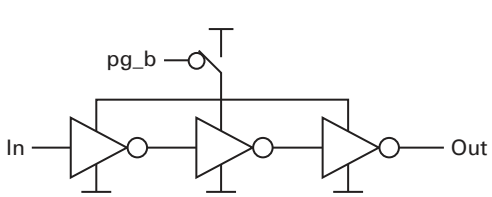

Fig. 1.4 Power gating applied to a chain of inverters.

compute this power gating signal. However, this replicated gate would then suffer from the exact same energy–performance trade-offs described earlier.

Clearly, attempting to power gate every single logic gate does not provide any benefit, but even for more moderate approaches (i.e., power gating individual sub-blocks), the key issue to keep in mind is that not only will the power gate itself introduce energy/performance overheads (due to voltage drops across the power gating device when it is active, and due to the energy consumed by driving the parasitic capacitance of the power gating device), the circuits to compute whether or not the power gate should be active will themselves introduce both static and dynamic energy overheads. Thus, power gating is usually only applied at relatively coarse levels of granularity where it is very straightforward to know (or be told by, e.g., the operating system) whether or not the underlying blocks are performing active work.

Even though power gating does not improve upon the fundamental energy–performance trade-offs described earlier, it is effective in dealing with the practical reality that in most applications, the required computations are bursty. For example, when a mobile phone is in standby mode, the applications processor is typically idle and/or only activated on regular intervals to perform some maintenance tasks. Only once the phone is turned on/being actively used would it be likely for the applications processor to have significant computational tasks to complete.

Continuing with the above example, let's assume that the applications processor as a whole is active only 10% of the time. Without power gating and in comparison to the case where the processor is being used continuously, the activity factor α is now effectively 10× lower, forcing a nearly identical 10× increase in the I_{on}/I_{off} ratio. With CMOS transistors and an 80 mV/dec sub-threshold slope, this would force one to increase the threshold voltage by approximately 80 mV, and hence the supply voltage by a similar percentage (to maintain the same performance). As shown in Fig. 1.5, the achievable energy/operation of this bursty processor would therefore be degraded relative to the case where the processor was used continuously. With an ideal (i.e., zero on-resistance, zero parasitic capacitance, and zero leakage) power gating device and "free" system-level cues to indicate when the processor is active or not, one could

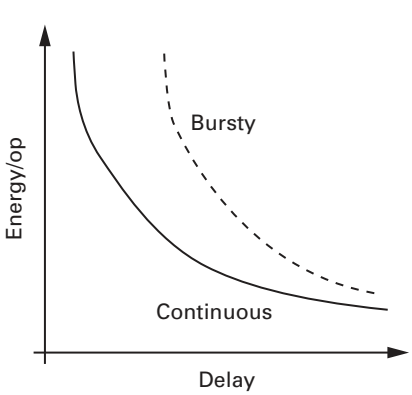

Fig. 1.5 Energy vs. delay implications of bursty vs. continuous usage of a digital circuit.

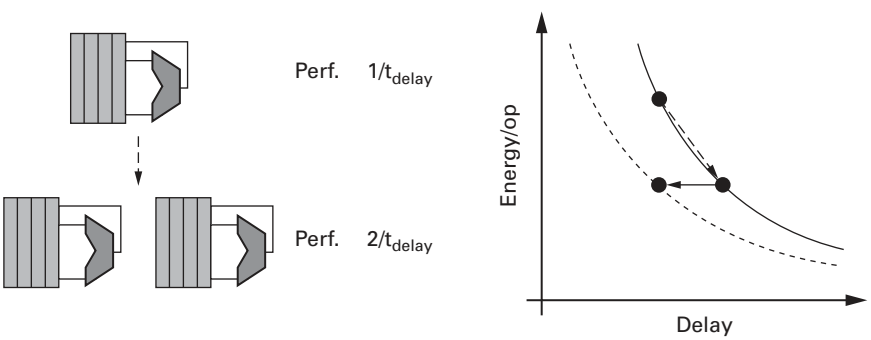

Fig. 1.6 Illustration of parallelism and how it improves the energy vs. performance trade-off on an example with two functional units compared to a single functional unit.

return the processor to the continuous-use energy-delay trade-off curve. In other words, the main benefit of power gating is that it reduces the penalty of the system-level variability in usage patterns.

Having examined the difficulties associated with eliminating or mitigating leakage within the logic gates themselves, we are still left with the fact that designers would like to utilize the dimensional scaling of transistors to simultaneously improve energy, performance, and functionality, but that scaling alone in the most straightforward manner (while leaving chip size fixed) would cause power consumption to increase substantially. Fortunately, there is a technique that designers can and have applied to exploit the availability of additional transistors to improve energy efficiency: parallelism [9].

The basic idea behind parallelism is quite straightforward, and is depicted in Fig. 1.6. In essence, if at the application level one has multiple pieces of data that can be operated on in parallel, replicating the digital hardware units and feeding them with the independent data inputs allows you to complete proportionally more operations within the same time period. Since our goal, however, is to improve energy efficiency, rather than simply increasing the throughput in this manner (but spending proportionally more power), we can instead run each unit more slowly – and therefore at lower energy/operation. As also highlighted in Fig. 1.6, in comparison to a design where we tried to achieve the same performance by running a single unit at a higher frequency (i.e., lower delay), because each of its functional units can operate at a lower energy point of the curve, the parallel implementation can be significantly more energy efficient.

In practice, parallelism does not work quite as ideally as depicted in Fig. 1.6 – there are always some overheads involved in distributing/collecting the data to/from the various units, and not all applications (or even sections of code within a given application) naturally offer parallelism. These overheads can fortunately be made relatively minimal, and so for approximately the last decade, parallelism has indeed been the primary workhorse of the semiconductor industry to convert the availability of additional transistors in a scaled process technology into improved performance without breaking the power budget. In fact, it is very difficult to purchase a laptop PC without at least four cores integrated onto the central processing unit, and even within smartphones

the vast majority of the applications processors utilize at least two cores. However, as we will describe next, even parallelism will soon cease (or perhaps even already has ceased) to be an effective tool for improving energy efficiency.

1.4 Energy limits and conclusions

As first described by Calhoun and Chandrakasan in [10], once a CMOS circuit is operated in the sub-threshold regime, for essentially any combination of L_d, f, and α of practical interest, there is a well-defined minimum energy/operation that the circuit must dissipate. To understand the reasons behind this, we can simply re-examine Eq. (1.3) from Section 1.2, and recall that in the sub-threshold region of operation, I_{off} is exponentially dependent upon $-V_{th}$, while I_{on} is exponentially dependent upon $(V_{dd} - V_{th})$. In this case, the I_{on}/I_{off} ratio depends purely on $V_{dd} - V_{th}$ specifically:

$$E_{total} = \alpha \cdot L_d \cdot f \cdot C \cdot {V_{dd}}^2 \cdot \left[1 + \frac{L_d \cdot f}{2\alpha} \cdot \frac{e^{\left(\frac{-V_{th}}{nkT/q}\right)}}{e^{\left(\frac{V_{dd}-V_{th}}{nkT/q}\right)}}\right] = \alpha \cdot L_d \cdot f \cdot C \cdot {V_{dd}}^2 \cdot \left[1 + \frac{L_d \cdot f}{2\alpha} \cdot e^{\left(\frac{-V_{dd}}{nkT/q}\right)}\right]. \quad (1.4)$$

By plotting Eq. (1.4) above, it is easy to see that there is a specific value of V_{dd} that optimally balances the leakage and dynamic energy contributions. Reducing V_{dd} any further below this point actually increases the total energy because the exponential increase in the delay of the circuit causes the leakage energy actually to increase (despite the reduced supply voltage). The threshold voltage has no effect on the total energy because, even though increasing the threshold exponentially decreases the leakage current, it also exponentially increases the delay. So, increasing V_{th} simply allows the circuit to operate slower, but at no lower energy than before.

Parallelism relies on the principle that running a circuit slower will allow it to achieve lower energy/operation. As depicted in Fig. 1.7 and pointed out in [10], once each sub-unit operates at its minimum energy, slowing down each sub-unit further brings no further improvement in energy/operation – the supply voltage V_{dd} for each unit should remain fixed regardless of the degree of parallelization.

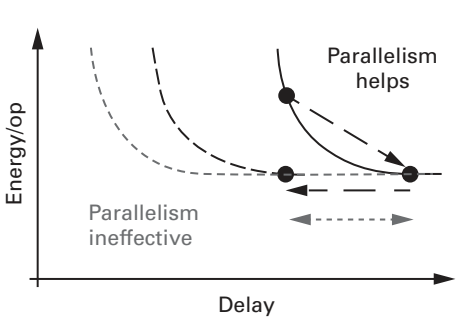

Fig. 1.7 Illustration of the limits of parallelism due to minimum energy/operation.