Chapter 2
Basic Low Power Digital Design

Moore's law [12], which states that the "number of transistors that can be placed inexpensively on an integrated circuit will double approximately every two years," has often been subject to the following criticism: while it boldly states the blessing of technology scaling, it fails to expose its bane. A direct consequence of Moore's law is that the "power density of the integrated circuit increases exponentially with every technology generation". History is witness to the fact that this was not a benign outcome. This implicit trend has arguably brought about some of the most important changes in electronic and computer designs. Since the 1970s, most popular electronics manufacturing technologies used bipolar and nMOS transistors. However, bipolar and nMOS transistors consume energy even in a stable combinatorial state, and consequently, by 1980s, the power density of bipolar designs was considered too high to be sustainable. IBM and Cray started developing liquid, and nitrogen cooling solutions for high-performance computing systems [5,11,16,19,21,23–25]. The 1990s saw an inevitable switch to a slower, but lower-power CMOS technology (Fig. 2.1). CMOS transistors consume lower power largely because, to a first order of approximation, power is dissipated only when they switch states, and not when the state is steady. Now, in the late 2000s, we are witnessing a paradigm shift in computing: the shift to multi-core computing. The power density has once again increased so much that there is little option but to keep the hardware simple, and transfer complexity to higher layers of the system design abstraction, including software layers.

This chapter explains the basics of power consumption and dissipation in the operation of CMOS transistors, and also discusses some of the fundamental mechanisms employed for power reduction.

2.1 CMOS Transistor Power Consumption

The power consumption of a CMOS transistor can be divided into three different components: dynamic, static (or leakage) and short circuit power consumption. Figure 2.2 illustrates the three components of power consumption. Switching power,

P.R. Panda et al., *Power-efficient System Design*, DOI 10.1007/978-1-4419-6388-8_2,

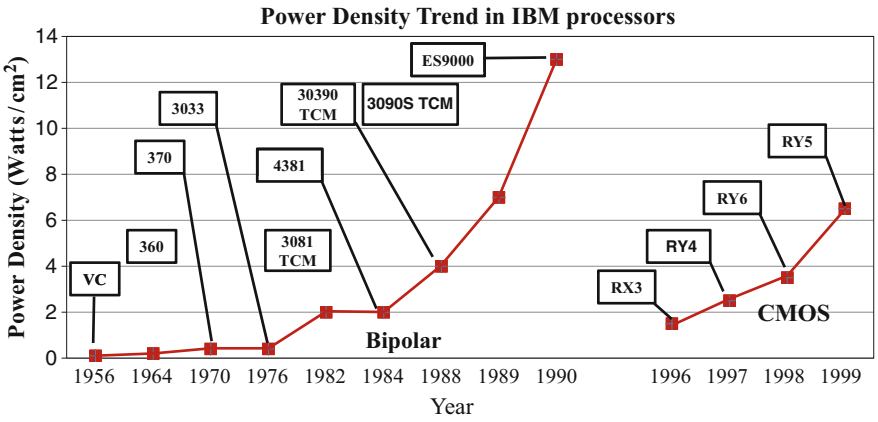


Fig. 2.1 Unsustainable increase in the power density caused a switch from fast, but high-power bipolar transistors to slow, but low-power CMOS transistor technology. We are at the doorsteps of the same problem again (Data courtesy IBM Corp)

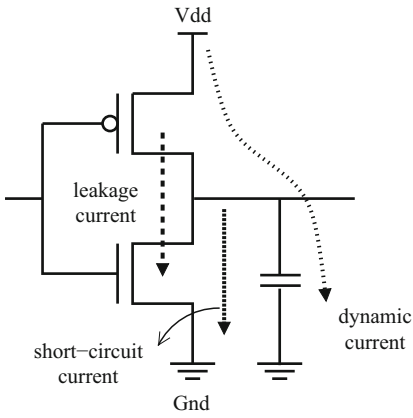


Fig. 2.2 The dynamic, short circuit and leakage power components of transistor power consumption. Dynamic and short circuit power are also collectively known as switching power, and are consumed when transistors change their logic state, but leakage power is consumed merely because the circuit is "powered-on"

which includes both dynamic power and short-circuit power, is consumed when signals through CMOS circuits change their logic state, resulting in the charging and discharging of load capacitors. Leakage power is primarily due to the sub-threshold currents and reverse biased diodes in a CMOS transistor. Thus,

$$P_{total} = P_{dynamic} + P_{short-circuit} + P_{leakage}. \tag{2.1}$$

2.1.1 Switching Power

When signals change their logic state in a CMOS transistor, energy is drawn from the power supply to charge up the load capacitance from 0 to V_{dd}. For the inverter example in Fig. 2.2, the power drawn from the power supply is dissipated as heat in pMOS transistor during the charging process. Energy is needed whenever charge is moved against some potential. Thus, $dE = d(QV)$. When the output of the invertor transitions from logical 0 to 1, the load capacitance is charged. The energy drawn from the power supply during the charging process is given by,

$$dE_P = d(VQ) = V_{dd}.dQ_L$$

since the power supply provides power at a constant voltage V_{dd}. Now, since $Q_L = C_L.V_L$, we have:

$$dQ_L = C_L.dV_L$$

Therefore,

$$dE_P = V_{dd}.C_L.dV_L$$

Integrating for full charging of the load capacitance,

$$\begin{aligned} E_P &= \int_0^{V_{dd}} V_{dd}.C_L.dV_L \\ &= V_{dd}.C_L.\int_0^{V_{dd}} dV_L \\ &= C_L.V_{dd}^2 \end{aligned} \tag{2.2}$$

Thus a total of $C_L.V_{dd}^2$ energy is drawn from the power source. The energy E_L stored in the capacitor at the end of transition can be computed as follows:

$$dE_L = d(VQ) = V_L.dQ_L$$

where V_L is the instantaneous voltage across the load capacitance, and Q_L is the instantaneous charge of the load capacitance during the loading process. Therefore,

$$dE_L = V_L.C_L dV_L$$

Integrating for full charging of the load capacitance,

$$\begin{aligned} E_L &= \int_0^{V_{dd}} C_L.V_L.dV_L \\ &= \frac{(C_L.V_{dd}^2)}{2} \end{aligned} \tag{2.3}$$

Comparing Equations 2.2 and 2.3, we notice that only half of the energy drawn from the power supply is stored in the load capacitance; the rest is dissipated as heat. This $\frac{1}{2}\ C_L.V_{dd}^2$ energy stored in the output capacitance is released during the discharging of the load capacitance, which occurs when the output of the invertor transitions from logical 1 to 0. The load capacitance of the CMOS logic gate consists of the output node capacitance of the logic gate, the effective capacitance of the interconnects, and the input node capacitance of the driven gate.

2.1.2 Short Circuit Power

Another component of power, *short-circuit power* (also known as *crowbar power*, or —em rush-through power) becomes important because of finite non-zero rise and fall times of transistors, which causes a direct current path between the supply and ground. This power component is usually not significant in logic design, but it appears in transistors that are used to drive large capacitances, such as bus wires and especially off-chip circuitry. As wires on chip became narrower, long wires became more resistive. CMOS gates at the end of those resistive wires see slow input transitions. Consider the inverter in Fig. 2.2. Figure 2.3 shows the variation of the short circuit current I_{SC} as the inverter is driven by a rising ramp input from time 0 to T_R. As the input voltage rises, at time t_0, we have $V_{in} > VT_n$, i.e., the input voltage become higher than the threshold voltage of the nMOS transistor. At this time a short-circuit current path is established. This short circuit current increases as the nMOS transistor turns "on". Thereafter, the short circuit current first increases and then decreases until, after t_1, we have $V_{in} > V_{dd} - VT_p$, i.e., the pMOS transistor turns off, signalling the end of short-circuit current. Therefore, in the duration when

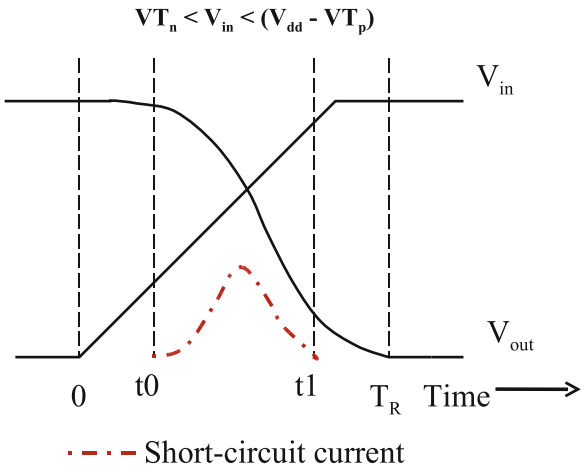


Fig. 2.3 Short circuit power is consumed in a circuit when both the nMOS and pMOS transistors are "on"

$(VT_n < V_{in} < (V_{dd} - VT_p))$ holds, there will be a conductive path open between V_{dd} and GND because both the nMOS and pMOS devices will be simultaneously on. Short-circuit power is typically estimated as:

$$P_{short-circuit} = 1/12.k.\tau.F_{clk}.(V_{dd} - 2V_t)^3 \tag{2.4}$$

where τ is the rise time and fall time (assumed equal) and k is the gain factor of the transistor [22].

2.1.3 Leakage Power

The third component of power dissipation in CMOS circuits, as shown in Equation 2.1, is the static or leakage power. Even though a transistor is in a stable logic state, just because it is powered-on, it continues to leak small amounts of power at almost all junctions due to various effects. Next we discuss about the significant components of leakage power.

2.1.3.1 Reverse Biased Diode Leakage

The reverse biased diode leakage is due to the reverse bias current in the parasitic diodes that are formed between the diffusion region of the transistor and substrate. It results from minority carrier diffusion and drift near the edge of depletion regions, and also from the generation of electron hole pairs in the depletion regions of reverse-bias junctions. As shown in Fig. 2.4, when the input of inverter in Fig. 2.2 is high, a reverse potential difference of V_{dd} is established between the drain and the n-well, which causes diode leakage through the drain junction. In addition, the n-well region of the pMOS transistor is also reverse biased with respect to the p-type substrate. This also leads to reverse bias leakage at the n-well junction. The reverse bias leakage current is typically expressed as,

$$I_{rbdl} = A.Js.(e^{q.V_{bias}/kT} - 1) \tag{2.5}$$

where A is the area of the junction, J_s is the reverse saturation current density, and V_{bias} is the reverse bias voltage across the junction, and $V_{th} = kT/q$ is the thermal voltage. Reverse biased diode leakage will further become important as we continue to heavily dope the n- and p-regions. As a result, zener and band-to-band tunneling [14] will also become contributing factors to the reverse bias current.

2.1.3.2 Gate Induced Drain Leakage

Gate-induced drain leakage (GIDL) is caused by high field effect in the drain junction of MOS transistors [2]. In an nMOS transistor, when the gate is biased to form

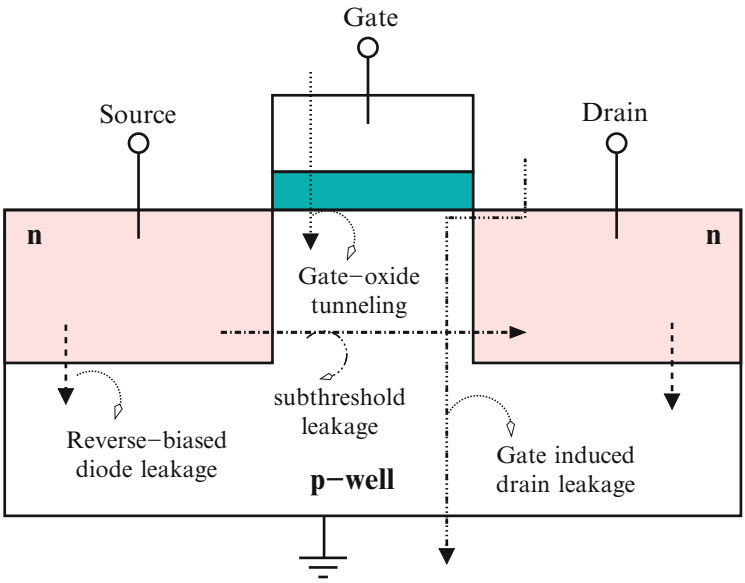


Fig. 2.4 Components of Leakage Power: (i) Subthreshold current flows between source and drain; (ii) Reverse-biased diode leakage flows across the parasitic diodes; (iii) Gate induced drain leakage flows between the drain and substrate

accumulation layer in the silicon surface under the gate, the silicon surface has almost the same potential as the p-type substrate, and the surface acts like a p-region more heavily doped than the substrate. However, when the gate is at zero or negative voltage and the drain is at the supply voltage level, there can be a dramatic increase of effects like avalanche multiplication and band-to-band tunneling. Minority carriers underneath the gate are swept to the substrate, creating GIDL current. GIDL current or I_{GIDL} is typically estimated as:

$$I_{GIDL} = AE_s.e^{-\frac{B}{E_s}} \tag{2.6}$$

where E_s is the transverse electric field at the surface [3]. Thinner oxide T_{ox} and higher supply voltage V_{dd} increase GIDL. GIDL is also referred to as surface band-to-band tunneling leakage.

2.1.3.3 Gate Oxide Tunneling

Gate oxide tunneling current I_{ox}, flows from the gate through the oxide insulation to the substrate. In oxide layers thicker than 3-4 nm, this type of current results from the Fowler-Nordheim tunneling of electrons into the conduction band of the oxide layer under a high applied electric field across the oxide layer [17]. As oxides

get thinner, this current could surpass many other smaller leakages, e.g., weak inversion and DIBL as a dominant leakage mechanism in the future. I_{ox} is typically estimated as:

$$I_{ox} = A.E_{ox}^2.e^{-\frac{B}{E_{ox}}} \tag{2.7}$$

where E_{ox} is the electric field across the oxide.

In oxide layers less than 3-4 nm thick, there can also be direct tunneling through the silicon oxide layer. Mechanisms for direct tunneling include electron tunneling in the conduction band (ECB), electron tunneling in the valence band (EVB), and hole tunneling in the valence band (HVB).

2.1.3.4 Subthreshold Leakage

Subthreshold current flows from the source to drain even if the gate to source voltage is below the threshold voltage of the device. This happens due to several reasons. First is the weak inversion effect: when the gate voltage is below V_T, carriers move by diffusion along the surface similar to charge transport across the base of bipolar transistors. Weak inversion current becomes significant when the gate to source voltage is smaller than but very close to the threshold voltage of the device. The second prominent effect is the Drain-Induced Barrier Lowering (DIBL). DIBL is essentially the reduction of threshold voltage of the transistor at higher drain voltages. As the drain voltage is increased, the depletion region of the p-n junction between the drain and body increases in size and extends under the gate, so the drain assumes a greater portion of the burden of balancing depletion region charge, leaving a smaller burden for the gate. As a result, the charge present on the gate retains the charge balance by attracting more carriers into the channel, an effect equivalent to lowering the threshold voltage of the device. DIBL is enhanced at higher drain voltage and shorter effective channel length (L_{eff}) [14]. The third effect is the direct punch-through of the electrons between drain and source. It occurs when when the drain and source depletion regions approach each other and electrically "touch" deep in the channel. In a sense, punch-through current is a subsurface version of DIBL.

As a combination of all these sub-currents, I_{sub} is typically modeled as:

$$I_{sub} = I_0 e^{\frac{V_G - V_S - V_{T0} - \gamma V_S + \eta V_{DS}}{n V_{th}}} \left(1 - e^{\frac{-V_{DS}}{V_{th}}}\right), \tag{2.8}$$

where $V_{th} = kT/q$ is the thermal voltage, n is the subthreshold swing coefficient constant, γ is the linearized body effect coefficient, η is the DIBL coefficient, and I_0 is the technology dependent subthreshold leakage which can be represented as,

$$I_0 = \mu_0 C_{ox} \frac{W}{L} V_{th}^2 e^{1.8}. \tag{2.9}$$

2.2 Trends in Power Consumption

At the macro-level, the most significant trend is the increasing contribution of leakage power in the total power dissipation of an electronic system designed in CMOS technology. For a long time, the switching component of the dynamic power was the major component of the total power dissipated by a circuit. However, in order to keep power dissipation and power delivery costs under control, the operational voltage V_{dd} was scaled down at the historical rate of 30% per technology generation. In conjunction, to improve transistor and circuit performance the threshold voltage V_t was also reduced at the same rate so that a sufficiently large gate overdrive $(V_{dd} - V_t)$ is maintained. However, as seen from Equation 2.8, reduction in V_t causes transistor subthreshold leakage current (I_{sub}) to increase exponentially. Furthermore, other components of leakage current, e.g., the gate leakage and reverse-biased junction Band To Band Tunneling (BTBT) become important as we scale fabrication technology to 45 nm and downwards. Other factors such as gate-induced drain leakage (GIDL) and drain-induced barrier lowering (DIBL) will also become increasingly significant.

In addition to the increasing dominance of leakage power, the subthreshold leakage and the gate-oxide tunneling increase extremely rapidly (exponentially) with technology scaling, and dwarf dynamic power. We are already at a point where $V_{dd} - V_t$ is low, and leakage power is comparable to dynamic switching power, and in some cases, may actually even dominate the overall power dissipation. Large leakage comes with several associated problems such as lower noise immunity of dynamic logic circuits, instability of SRAM cells, and eventually, lower yield.

Another dimension of worry is added by the fact that unlike dynamic power, leakage power increases exponentially with temperature. In order to improve performance we have been continuously scaling the supply and threshold voltages. While this results in high frequency of operation, temperatures rise due to large active power consumption. The high temperature increases the sub-threshold leakage (which is a strong function of temperature), further increasing temperature. This circular situation is depicted in Fig. 2.5. If heat cannot be dissipated effectively, a positive feedback between leakage power and temperature can result in thermal runaway. Such a situation can have disastrous consequences, including permanent physical damage of the circuit. Most processors are now equipped with thermal sensors and hardware circuitry that will stop the processor if the temperature increases beyond safe limits.

2.3 Techniques for Reducing Dynamic Power

The dynamic power of a circuit in which all the transistors switch exactly once per clock cycle will be $\frac{1}{2}CV^2F$, if C is the switched capacitance, V is the supply voltage, and F is the clock frequency. However, most of the transistors in a circuit rarely switch from most input changes. Hence, a constant called the activity

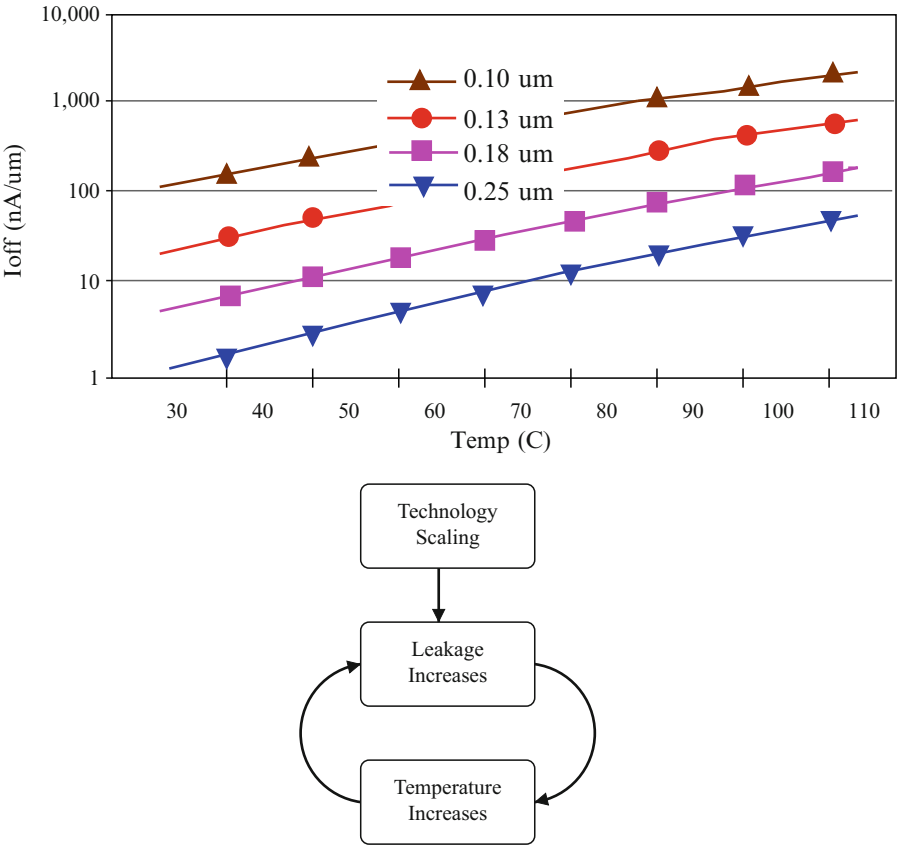


Fig. 2.5 Leakage increases exponentially with temperature. This can create a positive feedback loop, where high power-density increases temperature, which in turn further increases the power-density, causing a thermal runaway

factor ($0 \leq A \leq 1$) is used to model the average switching activity in the circuit. Using A, the dynamic power of a circuit composed of CMOS transistors can be estimated as:

$$P = ACV^2F \tag{2.10}$$

The importance of this equation lies in pointing us towards the fundamental mechanisms of reducing switching power. Figure 2.6 shows that one scheme is by reducing the activity factor A. The question here is: "how to achieve the same functionality by switching only a minimal number of transistors?" Techniques to do this span several design hierarchy levels, right from the synthesis level, where, for example, we can encode states so that the most frequent transitions occur with minimal bit switches, to the algorithmic level, where, for example, changing the sorting algorithm from insertion sort to quick sort, will asymptotically reduce the resulting

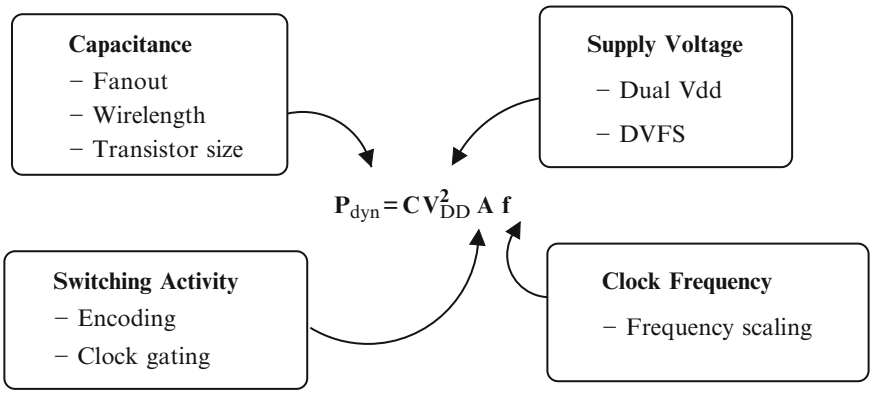


Fig. 2.6 Fundamental techniques to reduce dynamic power

switching activity. The second fundamental scheme is to reduce the load capacitance, C_L. This can be done by using small transistors with low capacitances in non-critical parts of the circuit. Reducing the frequency of operation F will cause a linear reduction in dynamic power, but reducing the supply voltage V_{DD} will cause a quadratic reduction. In the following sections we discuss some of the established and effective mechanisms for dynamic power reduction.

2.3.1 Gate Sizing

The power dissipated by a gate is directly proportional to its capacitive load C_L, whose main components are: i) output capacitance of the gate itself (due to parasitics), ii) the wire capacitance, and iii) input capacitance of the gates in its fanout. The output and input capacitances of gates are proportional to the gate size. Reducing the gate size reduces its capacitance, but increases its delay. Therefore, in order to preserve the timing behavior of the circuit, not all gates can be made smaller; only the ones that do not belong to a critical path can be slowed down.

Any gate re-sizing method to reduce the power dissipated by a circuit will heavily depend on the accuracy of the timing analysis tool in calculating the true delay of the circuit paths, and also discovering false paths. Delay calculation is relatively easier. A circuit is modeled as a directed acyclic graph. The vertices and edges of the graph represent the components and the connection between the components in the design respectively. The weight associated with a vertex (an edge) is the delay of the corresponding component (connection). The delay of a path is represented by the sum of the weights of all vertices and edges in the path. The arrival time at the output of a gate is computed by the length of the longest path from the primary inputs to this gate. For a given delay constraint on the primary outputs, the required time is the time at which the output of the gate is required to be stable. The time slack

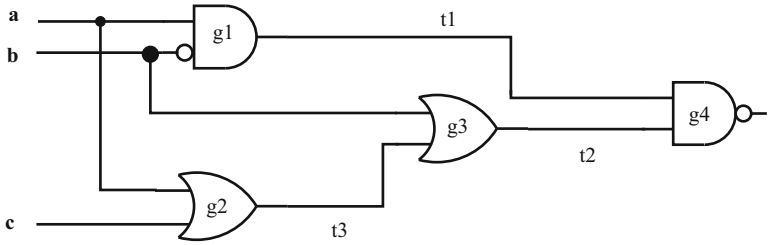


Fig. 2.7 False paths, which do not affect the timing, must be discovered and excluded during timing analysis before performing gate sizing

is defined as the difference of the required time and the arrival time of a gate. If the time slack is greater than zero, the gate can be down-sized. Consider the example circuit in Fig. 2.7. Assuming the delay of AND and OR gates to be 4 units, delay of NOT gate to be 1 unit, and wire delays to be 0, the top 3 timing-critical paths in the circuit are:

$$
\begin{aligned}
p_1 &: a \rightarrow g_2 \rightarrow g_3 \rightarrow g_4 \rightarrow d \\
p_2 &: b \rightarrow g_1 \rightarrow g_4 \rightarrow d \\
p_3 &: c \rightarrow g_2 \rightarrow g_3 \rightarrow g_4 \rightarrow d
\end{aligned}
$$

The delay of the path p_1 is 10 units, p_2 is 13 units, and p_3 is 13 units. Therefore static timing analysis will conclude that the gates in both the paths p_2 and p_3 cannot be down-sized. However, by logic analysis, it is easy to figure out that both these seemingly timing critical paths are actually false paths, and can never affect the final stable value of output d. Logically, we have:

$$
\begin{aligned}
d &= \overline{t1.t2} \\
&= \overline{(a.\overline{b}).(b + t3)} \\
&= \overline{(a.\overline{b}).(b + (a + c))} \\
&= \overline{(a.\overline{b}).(a + b + c)} \\
&= \overline{(a.\overline{b}.a) + (a.\overline{b}.b) + (a.\overline{b}.c)} \\
&= \overline{(a.\overline{b}) + (a.\overline{b}.c)} \\
&= \overline{(a.\overline{b}).(1 + c)} \\
&= \overline{(a.\overline{b})}
\end{aligned}
$$

Thus, the actual critical path of output d is p_1, which has a delay of only 10 units. Therefore, from the perspective of output d, gates that constitute paths p_2 and p_3, e.g., g_2 and g_3 can be down-sized. The question now is: how to find out if a path is

false? A false path is a path that is not *sensitizable*. A path $p = (i, g_0, g_1, ...g_n, o)$ from i to o is sensitizable if a $0 \rightarrow 1$ or $1 \rightarrow 0$ transition at input i can propagate along the entire path p to output o. Whether a change in the inputs will affect the final output at the end of a path however, depends on the values of the other inputs. Since for a given circuit with n primary inputs, there are 2^n possible input vectors, finding out whether a path is sensitizable by enumerating all the input values is clearly infeasible.

To work around this, typically a path-based approach is taken [6, 10, 26]. In order to allow a signal to go through path p from primary input i to the primary output o, we need to set all the other signals feeding to gates along p to be non-control values. The non-control value for an AND (OR) gate is 1 (0). For the path $p_1 = (b, g_1, g_4, d)$, we must set t_2 to be 1, and a to be 1. t_2 can be set to 1 by setting $t3$ to 1, and b to 0. Finally, $t3$ can be set to 1 by setting a to 1, and c can be anything. In this case, there is an assignment of primary inputs that will allow this. On the other hand, if there is no consistent assignment of primary inputs resulting in non-control values of the other inputs of the gates along a path, then the path is non-sensitizable, or a false path. For example, sensitizing path p_3 requires t_1 to be 1, b to be 0, and a to be 0. But this is not possible, because t_1 cannot be 1 with a set to 0. Therefore, path p_3 is a false path.

Taking path sensitizability into account, the calculation of slack time can now be formulated. For an input vector v, let $AT(g_i, v)$ be the arrival time of gate g_i and $RT(g_i, v)$ be the required time of gate g_i under a given delay constraint. The time slack of gate g_i with respect to the input vector v is given by:

$$slack(g_i, v) = RT(g_i, v) - AT(g_j, v)$$

For all input vectors, the slack of gate g_i is defined as:

$$slack(g_i) = min_{\forall v} slack(g_i, v)$$

All gates with slack time greater than zero are candidates for down-sizing. However, to choose a specific one, we must consider several factors, such as the power reduction achievable, and the slack consumption by resizing it. To achieve maximum power reduction, the gate with least delay increment should have higher priority, but to save the total consumed time slack of multiple paths by down-sizing a gate, the gate belonging to a smaller number of paths is preferable for resizing. A combined metric could be:

$$gain(g_i) = \frac{\delta power(g_i)}{\delta delay(g_i) \times |P_{g_i}|}$$

where where $\delta power(g_i)$ is the reduction in power, $\delta delay(g_i)$ is the increase in delay by down-sizing gate g_i, and $|P_{g_i}|$ is the number of noncritical paths passing through gate g_i. The gate with the maximum gain value is selected for resizing [10].

After a gate on a non-critical path is down-sized, a critical path may become non-critical since the load capacitances of some gates on the critical path may decrease. Consequently, the gates on the original critical path may be down-sized. Thus, simultaneous down-sizing and up-sizing multiple gates may reduce power consumption. The satisfaction of delay constraint can be retained if up-sizing gates can compensate the loss in delay caused by downsizing gates on the critical path.

2.3.2 Control Synthesis

Most control circuits are conceived as Finite State Machines (FSM), formally defined as graphs where the nodes represent states, and directed edges, labeled with inputs and outputs, describe the transition relation between states. The state machine is eventually implemented using a state register and combinational logic, that takes in the current state, the current inputs and computes the outputs and the new state, which is then written into the state register at the end of the cycle. The binary values of the inputs and outputs of the FSM are usually determined by external requirements, while the state encoding is left to the designer. Depending on the complexity of the circuit, a large fraction of the power is consumed due to the switching of the state register; this power is very dependent on the selected state encoding. The objective of low power approaches is therefore to choose a state encoding that minimizes the switching power of the state register.

Given a state encoding, the power consumption can be modeled as:

$$P = \frac{1}{2} V_{dd}^2 f \times C_{sr} \times E_{sr}$$

where f is the clock frequency of the state machine, C_{sr} is the effective capacitance of the state register, and E_{sr} is the expected state register switching activity. If S is the set of all states, we can estimate E_{sr} as:

$$E_{sr} = \sum_{i,j \in S} p_{ij} \times h_{ij}$$

where p_{ij} is the probability of a transition between states i and j, and h_{ij} is the Hamming Distance between the codes of states i and j. The best way to estimate p_{ij} is to apply a sufficiently long series of input patterns until the state occurrence and transition probabilities converge towards discrete values [13]. Otherwise, equal probabilities of all input patterns can be assumed, and p_{ij} can be determined stochastically by solving Chapman Kolmogorov equations [4].

Once we have the state transition probabilities, the state encoding problem can be formulated as an embedding of the state transition graph into a Boolean hypercube Fig. 2.8. A Boolean hypercube of dimension n is a graph with 2^n nodes, where every node is labeled with a unique binary value from 0 to $2^n - 1$. Every node v

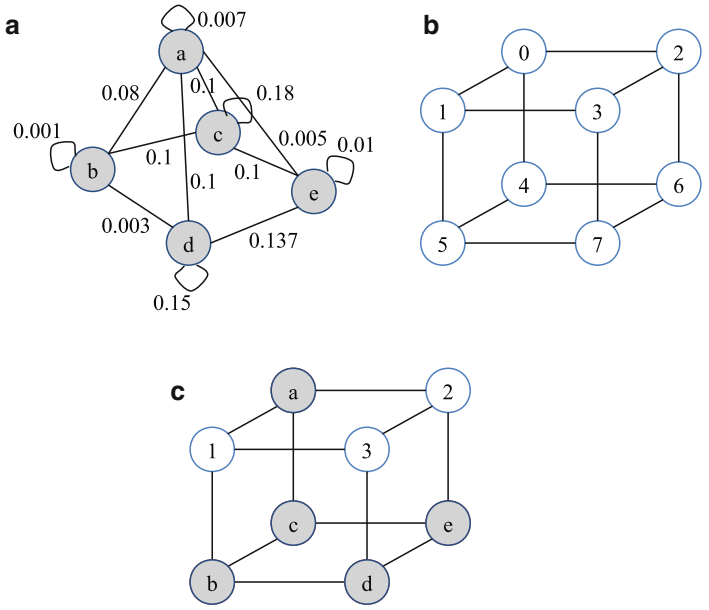


Fig. 2.8 (**a**) State transition graph (STG) contains nodes as states, and switching probabilities as edge weights. The problem of state encoding for low-power is of embedding the STG onto a k-dimensional hypercube. (**b**) A n-dimensional hypercube has 2^n nodes and edges connect nodes that have hamming distance of 1. (**c**) The problem is then to find an injective mapping of nodes from STG to nodes of the hybercube so that the sum of product of distances between nodes and switching frequency is minimized

is connected to n edges labeled $1, ...n$ leading to all nodes whose encodings have a Hamming Distance of 1 from v. Consequently, the hamming distance between any two nodes in the hypercube is equal to the length of the shortest path between the nodes. An embedding of a graph G into a host graph H is an injective mapping of the nodes of G to the nodes of H, so that every edge in G corresponds to the shortest path between the mappings of its terminal nodes in H. The *dilation* of an edge of G is defined as the length of the corresponding path in H.

The dimensionality of a hypercube refers to the number of bits in the state register. To optimize the switching power, a graph embedding with small dilation in a small hypercube must be found. We can always find a low dilation embedding in a hypercube of high dimensionality, but this increases the area and also the power cost of the implementation.

Given the dimensionality of the hypercube, the best solution would be an embedding with dilation 1 for all edges. While such an embedding is possible for cubical graphs, for many graphs it is not possible. The problem of finding an embedding with minimal dilation is NP-complete [7]. The problem of finding the minimum dimension in which a given graph can be embedded, is also NP-complete.

One of the effective solutions for this problem was suggested by Noth et al. [15], in which they create a cubical subgraph of G, which contains the edges with the highest weights, and find a dilation-1 embedding of this subgraph. Since all trees are cubical, a good choice for a subgraph is the maximum spanning tree (MST), which can be easily extracted using greedy algorithms, e.g., Prim's algorithm [9]. Since the dimension of the embedding is strongly connected to the degree of nodes in the tree, tighter embeddings can be found by limiting the degree of any node in the resulting spanning tree. Once the MST is obtained, they embed the MST onto the hypercube using a divide and conquer approach. For this, they first find the center of the tree V_c, E_c with respect to the longest paths. If $p = (v_0, ...v_k)$ is the longest path of the MST, then $V_C^p = \{v_{\lfloor k/2 \rfloor}, v_{\lceil k/2 \rceil}\}$ is the set of nodes in the center of p, where k is the length of longest path p. The center of the tree can then be defined as the set of center nodes of each path. Thus, $V_c = \cup V_c^p$. After picking V_c, we need E_c. It can be shown that every tree has a unique center, essentially proving that there will be either one two nodes in the center of the tree. If there is one node, we can pick an edge of V_c along any of the longest paths; if there are two nodes in V_c, then the two nodes must be connected by an edge, and that edge is our E_c. Removing the edge E_c breaks up the longest path at or near its center, leaving two subtrees of unknown size and structure. Both subtrees are then embedded recursively. Clearly it would be best to balance the subtree embeddings with respect to dimension in order to minimize the dimension of the overall embedding. One approach is to select an edge $(v, w) \in E_c$, whose removal from E leads to the most evenly sized subtrees with respect to the number of edges of the subtrees.

The low power state encoding problem is similar to the classical state encoding problem that targets low area and high performance, but sufficiently different that the resulting encodings will have different properties. Relatively low state register power consumption can be expected.

2.3.3 Clock Gating

Clock signals are omnipresent in synchronous circuits. The clock signal is used in a majority of the circuit blocks, and since it switches every cycle, it has an activity factor of 1. Consequently, the clock network ends up consuming a huge fraction of the on-chip dynamic power. Clock gating has been heavily used in reducing the power consumption of the clock network by limiting its activity factor. Fundamentally, clock gating reduces the dynamic power dissipation by disconnecting the clock from an unused circuit block.

Traditionally, the system clock is connected to the clock pin on every flip-flop in the design. This results in three major components of power consumption:

1. power consumed by combinatorial logic whose values are changing on each clock edge;
2. power consumed by flip-flops – this has a non-zero value even if the inputs to the flip-flops are steady, and the internal state of the flip-flops is constant; and

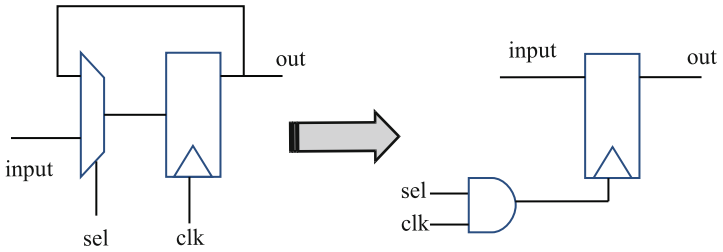


Fig. 2.9 In its simplest form, clock gating can be implemented by finding out the signal that determines whether the latch will have a new data at the end of the cycle. If not, the clock is disabled using the signal

3. power consumed by the clock buffer tree in the design. Clock gating has the potential of reducing both the power consumed by flip-flops and the power consumed by the clock distribution network.

Clock gating works by identifying groups of flip-flops sharing a common *enable* signal (which indicates that a new value should be clocked into the flip-flops). This enable signal is ANDed with the clock to generate the *gated clock*, which is fed to the clock ports of all of the flip-flops that had the common enable signal. In Fig. 2.9, the sel signal encodes whether the latch retains its earlier value, or takes a new input. This sel signal is ANDed with the clk signal to generate the gated clock for the latch. This transformation preserves the functional correctness of the circuit, and therefore does not increase the burden of verification. This simple transformation can reduce the dynamic power of a synchronous circuit by 5–10%.

There are several considerations in implementing clock gating. First, the enable signal should remain stable when clock is high and can only switch when clock is in low phase. Second, in order to guarantee correct functioning of the logic implementation after the gated-clock, it should be turned on in time and glitches on the gated clock should be avoided. Third, the AND gate may result in additional clock skew. For high-performance design with short-clock cycle time, the clock skew could be significant and needs to be taken into careful consideration.

An important consideration in the implementation of clock gating for ASIC designers is the granularity of clock gating. Clock gating in its simplest form is shown in Fig. 2.9. At this level, it is relatively easy to identify the enable logic. In a pipelined design, the effect of clock gating can be multiplied. If the inputs to one pipeline stage remain the same, then all the later pipeline stages can also be frozen. Figure 2.10 shows the same clock gating logic being used for gating multiple pipeline stages. This is a multi-cycle optimization with multiple implementation tradeoffs, and can can save significant power, typically reducing switching activity by 15–25%.

Apart from pipeline latches, clock gating is also used for reducing power consumption in *dynamic* logic. Dynamic CMOS logic is sometimes preferred over

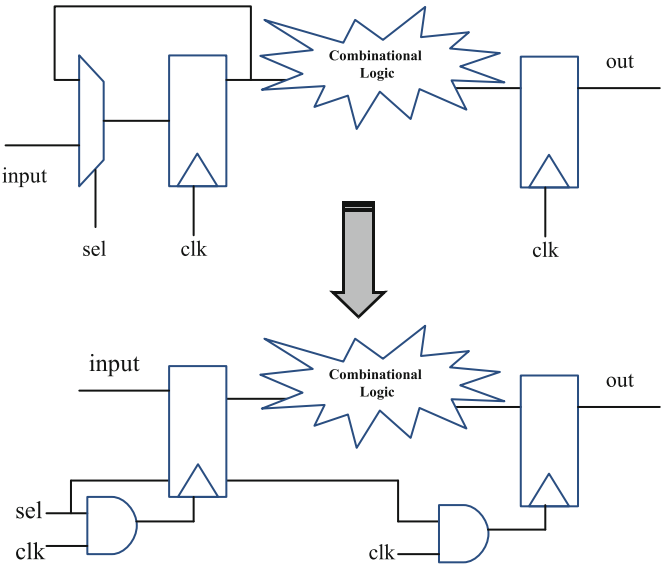


Fig. 2.10 In pipelined designs, the effectiveness of clock gating can be multiplied. If the inputs to a pipeline stage remain the same, then the clock to the later stages can also be frozen

static CMOS for building high speed circuitry such as execution units and address decoders. Unlike static logic, dynamic logic uses a clock to implement the combinational circuits. Dynamic logic works in two phases, *precharge* and *evaluate*. During precharge (when the clock signal is low) the load capacitance is charged. During evaluate phase (clock is high) depending on the inputs to the pull-down logic, the capacitance is discharged.

Figure 2.11 shows the gating technique applied to a dynamic logic block. In Fig. 2.11(a), when the clock signal is applied, the dynamic logic undergoes precharge and evaluate phases (charging the capacitances C_G and C_L) to evaluate the input *In*, so even if the input does not change, the power is dissipated to re-evaluate the same. To avoid such redundant computation, the clock port is gated as shown in Fig. 2.11(b). In this case, when the input does not change or when the output is not used, the gating signal is enabled, which prevents the logic from evaluating the inputs and thereby reduces dynamic power dissipation. An additional *AND* gate is introduced to facilitate clock gating. This additional logic presents its own capacitance and hence dissipates power, but compared to the power saved by preventing the charging of capacitances C_G and C_L (usually large for complex execution units), the *AND* gate power is negligible.

Clock gating at coarse granularity or system level is much more difficult to automate, and designers have to implement it in the functionality themselves. For example, sleep modes in a cell phone may strategically disable the display,

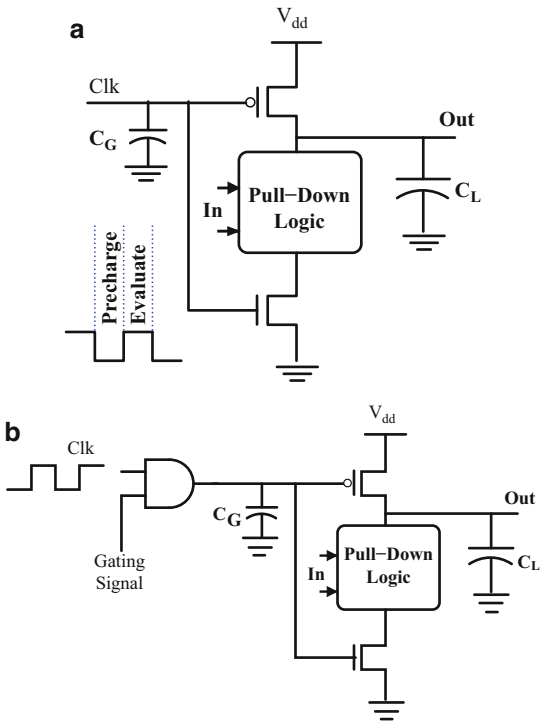


Fig. 2.11 (**a**) Dynamic CMOS Logic (**b**) Clock-gated Dynamic CMOS Logic

keyboard, or radio depending on the phone's current operational mode. System-level clock-gating shuts off entire RTL blocks. Because large sections of logic are not switching for many cycles it has the most potential to save power. However, it may result in inductive power issues due to higher di/dt, since large groups of circuits are turned on/off simultaneously. In contrast, local clock gating is more effective in reducing the worst-case switching power, and also suffers less from di/dt issues. However, local clock gating may lead to frequent toggling of the clock-gated circuit between enable and disable states, as well as higher area, power, and routing overhead, especially when the clock-gating control circuitry is comparable with the clock-gated logic itself.

2.3.4 Voltage and Frequency Scaling

Dynamic power is proportional to the square of the operating voltage. Therefore, reducing the voltage significantly improves the power consumption. Furthermore, since frequency is directly proportional to supply voltage, the frequency of the circuit can also be lowered, and thereby a cubic power reduction is possible. However, the delay of a circuit also depends on the supply voltage as follows.

$$\tau = k.C_L.\frac{V_{dd}}{(V_{dd} - V_t)^2} \tag{2.11}$$

where τ is the circuit delay, k is the gain factor, C_L is the load capacitance, V_{dd} is the supply voltage, and V_t is the threshold voltage. Thus, by reducing the voltage, although we can achieve cubic power reduction, the execution time increases. The main challenge in achieving power reduction through voltage and frequency scaling is therefore to obtain power reduction while meeting all the timing constraints.

Simple analysis shows that if there is slack in execution time, executing as slow as possible, while just meeting the timing constraints is more dynamic-power-efficient than executing as fast as possible and then idling for the remaining time. This is the main idea that is used in exploiting the power reduction that arises due to the cubic relationship with power, and inverse relationship with delay, of the supply voltage.

One approach to recover the lost performance is by scaling down the threshold voltage to the same extent as the supply voltage. This allows the circuit to deliver the same performance at a lower V_{dd}. However, smaller threshold voltages lead to smaller noise margins and increased leakage current. Furthermore, this cubic relationship holds only for a limited range of V_t scaling. The quadratic relationship between energy and V_{dd} deviates as V_{dd} is scaled down into the sub-threshold voltage level. In the sub-threshold region, while the dynamic power still reduces quadratically with voltage, the sub-threshold leakage current increases exponentially with the supply voltage. Hence dynamic and leakage power become comparable in the sub-threshold voltage region, and therefore, "just in time completion" is not energy inefficient. In practice, extending the voltage range below half V_{dd} is effective, but extending this range to sub-threshold operations may not be beneficial.

2.3.4.1 Design-Time Voltage and Frequency Setting

One of the most common ways to reduce power consumption by voltage scaling is that during design time, circuits are designed to exceed the performance requirements. Then, the supply voltage is reduced so as to just meet the performance constraints of the system. This is also called design-time voltage and frequency scaling. Design-time schemes scale and set the voltage and frequency, which remains constant (and therefore inefficient) for all applications at all times.

2.3.4.2 Static Voltage and Frequency Scaling

Systems can be designed for several (typically a few) voltage and frequency levels, and these levels can be switched at run time. In static voltage and frequency scaling, the change to a different voltage and frequency is pre-determined; this is quite popular in embedded systems. However, there are significant design challenges in supporting multiple voltages in CMOS design.

Timing analysis for multiple voltage design is complicated as the analysis has to be carried out for different voltages. This methodology requires libraries characterized for the different voltages used. Constraints are specified for each supply voltage level or operating point. There can be different operating modes for different voltages. Constraints need not be same for all modes and voltages. The performance target for each mode can vary. Timing analysis should be carried out for all these situations simultaneously. Different constraints at different modes and voltages have to be satisfied.

While local on-chip voltage regulation is good way to provide multiple voltages, unfortunately most of the digital CMOS technologies are not suitable for the implementation of either switched mode of operation or linear voltage regulation. A separate power rail structure is required for each power domain. These additional power rails introduce different levels of IR drop, imposing limits on the achievable power efficiency.

2.3.4.3 Dynamic Voltage and Frequency Scaling

The application and system characteristics can be dynamically analyzed to determine the voltage and frequency settings during execution. For example, adaptive scaling techniques make the decision on the next setting based on the recent history of execution.

Voltage and frequency scaling can also be applied to parts of a circuit. Thus, higher voltage can be applied to the timing critical path and modules in other paths may run on lower voltage. This maintains the overall system performance, while significantly reducing power consumption.

Implementing multiple on-chip voltage islands is also a challenge. Signals crossing from one voltage domain to another voltage domain have to be interfaced through *level shifter* buffers which appropriately shift the signal levels. Design of suitable level shifters is a challenging job. The speed at which different power domains switch on or off is also important – a low voltage power domain may activate early compared to the high voltage domain.

Every power domain requires independent local power supply and grid structure and some designs may even have a separate power pad. A separate power pad is possible in flip-chip designs where the power pad can be taken out near from the power domain. For other packaging technologies, the power pads have to be taken out from the periphery, which may impose a limit on the number of power domains.

2.4 Techniques for Reducing Short Circuit Power

Short circuit power is directly proportional to rise time and fall time on gates. Therefore, reducing the input transition times will decrease the short circuit current component. However, propagation delay requirements have to be considered

while doing so. Short circuit currents are significant when the rise/fall time at the input of a gate is much larger than the output rise/fall time. This is because the short circuit path will be active for a longer period of time. To minimize the total average short circuit current, it is desirable to have equal input and output edge times. In this case, the power consumed by the short circuit current is typically less than 10% of the total dynamic power. An important point to note is that if the supply is lowered to below the sum of the thresholds of the transistors, $V_{dd} < VT_n + |VT_p|$, the short-circuit currents can be eliminated because both devices will never be on at the same time for any input voltage value.

2.5 Techniques for Reducing Leakage Power

In order to contain the increase in the dynamic power, the supply V_{dd} has undergone a continous reduction in successive technology generations. Along with V_{dd}, V_t must also be scaled down, which results in an exponential increase in leakage power. Consequently, leakage power has become a significant contributor in the total chip power dissipation. Leakage power reduction techniques are especially important for handheld devices such as cell phones, which are "on", but not active most of the time. Consequently, even though such devices dissipate minimal dynamic energy, leakage power becomes a significant contributor in their power equation.

Some of the fundamental techniques to reduce leakage power are discussed in the following sections.

2.5.1 Multiple Supply Voltage

The multiple supply system provides a high-voltage supply for high-performance circuits and a low-voltage supply for low-performance circuits. In a dual V_{dd} circuit, the reduced voltage (low-V_{dd}) is applied to the circuit on non-critical paths, while the original voltage (high-V_{dd}) is applied to the circuit on critical paths. Since the critical path of the circuit is unchanged, this transformation preserves the circuit performance. If a gate supplied with low-V_{dd} drives a gate supplied with high-V_{dd}, the pMOS may never turn off. Therefore a level converter is required whenever a module at the lower supply drives a gate at the higher supply (step-up). Level converters are not needed for a step-down change in voltage. The overhead of level converters can be mitigated by doing conversions at register boundaries and embedding the level conversion inside the latch. Figure 2.12(a) shows a pipeline stage in which some of the paths have low-V_{dd} gates. These are shown in a darker shade in the figure. Notice that some high-V_{dd} gates drive low-V_{dd}, but not vice versa. The transition from low to high V_{dd} is condensed into the level converter latchesshown in the figure. A simple design of level converter latches is shown in Fig. 2.12(b).

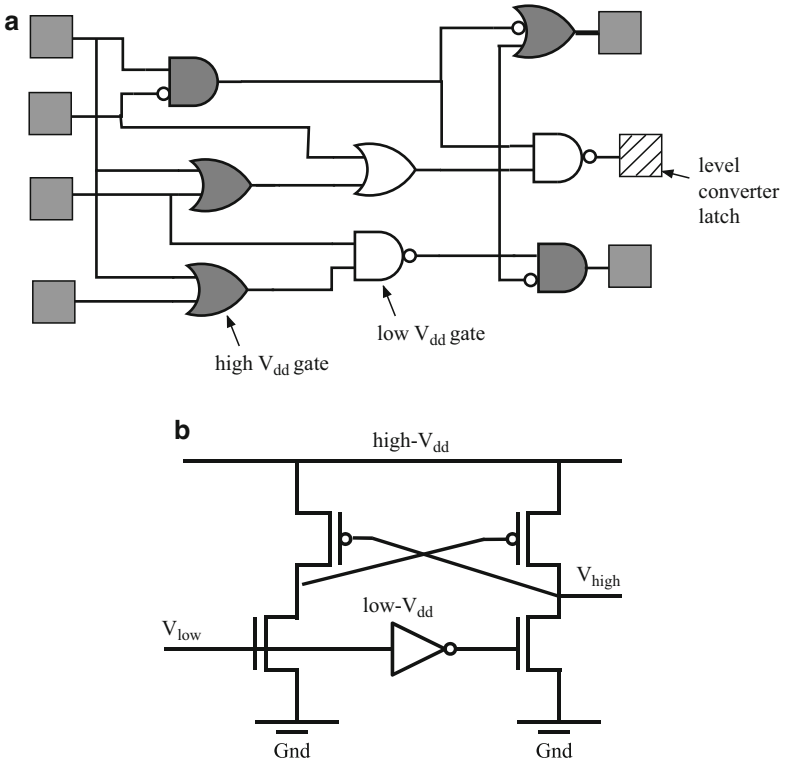


Fig. 2.12 Using multiple V_{dd}s essentially reduces the power consumption by exploiting the slack in the circuit. However, it requires a level converter. (**a**) Multiple supply-voltage pipeline stage. (**b**) Level converter latch

Essentially, the multiple V_{dd} approach reduces power by utilizing excessive slack in a circuit. Clearly, there is an optimum voltage difference between the two V_{dd}s. If the difference is small, the effect of power reduction is small, while if the difference is large, there are few logic circuits that can use low-V_{dd}. Compared to circuits that operate at only high V_{dd}, the power is reduced. The latch circuit includes a level-transition (DC-DC converter) if there is a path where a signal propagates from low V_{dd} logic to high V_{dd} logic.

To apply this technique, the circuit is typically designed using high-V_{dd} gates at first. If the propagation delay of a circuit path is less than the required clock period, the gates in the path are given low-V_{dd}. In an experimental setting [8], the dual V_{dd} system was applied on a media processor chip providing MPEG2 decoding and real-time MPEG1 encoding. By setting high-V_{dd} at 3.3 V and low-V_{dd} at 1.9 V, system power reduction of 47% in one of the modules and 69% in the clock distribution was obtained.

2.5.2 Multiple Threshold Voltage

Multiple V_t MOS devices are used to reduce power while maintaining speed. High speed circuit paths are designed using low-Vt devices, while the high-V_t devices are applied to gates in other paths in order to reduce subthreshold leakage current. Unlike the multiple-V_{dd} transformation, no level converter is required here as shown in Fig. 2.13. In addition, multi-V_t optimization does not change the placement of the cells. The footprint and area of low-V_t and high-V_t cells are similar. This enables timing-critical paths to be swapped by low-V_t cells easily. However, some additional fabrication steps are needed to support multiple V_t cells, which eventually lengthens the design time, increases fabrication complexity, and may reduce yield [1]. Furthermore, improper optimization of the design may utilize more low-V_t cells and hence could end up with increased power!

Several design approaches have been proposed for dual-V_t circuit design. One approach builds the entire device using low-V_t transistors at first. If the delay of a circuit path is less than the required clock period, the transistors in the path are replaced by high-V_t transistors. The second approach allows all the gates to be built with high-V_t transistors initially. If a circuit path cannot operate at a required clock speed, gates in the path are replaced by low-V_t versions. Finally, a third set of approaches target the replacement of groups of cells by high-V_t or low-V_t versions at one go.

In one interesting incremental scheme [18], the design is initially optimized using the higher threshold voltage library only. Then, the multi-V_t optimization computes the power-performance tradeoff curve up to the maximum allowable leakage power limit for the next lower threshold voltage library. Subsequently, the optimization starts from the most critical slack end of this power-performance curve and switches the most critical gate to next equivalent low-V_t version. This may increase the leakage in the design beyond the maximum permissible leakage power. To compensate for this, the algorithm picks the least critical gate from the other end of the power-performance curve and substitutes it with its high-V_t version. If this does not bring the leakage power below the allowed limit, it traverses further from the curve (from

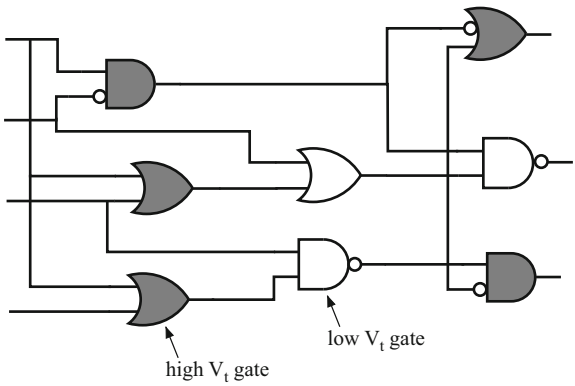


Fig. 2.13 Multiple V_t technology is very effective in power reduction without the overhead of level converters. The white gates are implemented using low-V_t transistors

least critical towards most critical) substituting gates with high-V_t gates, until the leakage limit is satisfied. Then the algorithm continues with the second most critical cell and switches it to the low-V_t version. The iterations continue until we can no longer replace any gate with the low-V_t version without violating the leakage power limit. The multi-V_t approach is very effective. In a 16-bit ripple-carry adder, the active-leakage current was reduced to one-third that of the all low-V_t adder [1].

2.5.3 Adaptive Body Biasing

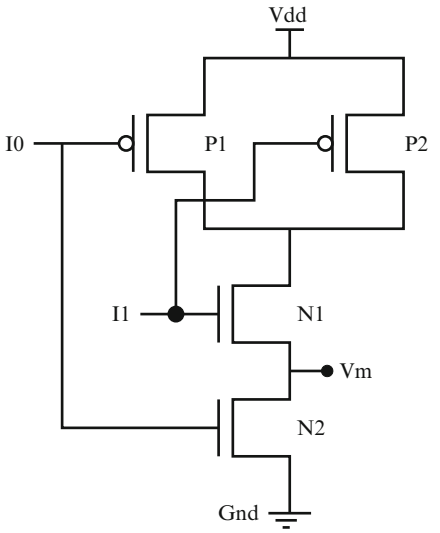
One efficient method for reducing power consumption is to use low supply voltage and low threshold voltage without losing performance. But increase in the lower threshold voltage devices leads to increased sub threshold leakage and hence more standby power consumption. One solution to this problem is *adaptive body biasing (ABB)*. The substrate bias to the n-type well of a pMOS transistor is termed V_{bp} and the bias to the p-type well of an nMOS transistor is termed V_{bn}. The voltage between V_{dd} and V_{bp}, or between *GND* and V_{bn} is termed V_{bb}. In the active mode, the transistors are made to operate at low-V_{dd} and low-V_t for high performance. The fluctuations in V_t are reduced by an adaptive system that constantly monitors the leakage current, and modulates V_{bb} to force the leakage current to be constant. In the idle state, leakage current is blocked by raising the effective threshold voltage V_t by applying substrate bias V_{bb}.

The ABB technique is very effective in reducing power consumption in the idle state, with the flexibility of even increasing the performance in the active state. While the area and power overhead of the sensing and control circuitry are shown to be negligible, there are some manufacturing-related drawbacks of these devices [20]. ABB requires either twin well or triple well technology to achieve different substrate bias voltage levels in different parts of the IC. Experiments applying ABB to a discrete cosine transform processor reported a small 5% area overhead. The substrate-bias current of V_{bb} control is less than 0.1% of the total current, a small power penalty.

2.5.4 Transistor Stacking

The subthreshold leakage current flowing through a stack of transistors connected in series, is reduced if atleast one of them is switched *off*. For example, consider the NAND gate schematic in Fig. 2.14. When both N1 and N2 are turned off, the voltage V_m at the intermediate node between N1 and N2 is positive due to the small drain current. This has the following three effects on subthreshold leakage current of the circuit:

- Due to the positive potential V_m, the gate to source volatage of transistor N1 $(0 - V_m = -V_m)$ becomes negative, resulting in reduced subthreshold current of N1.



Inputs (I0 I1)	Leakage
00	$2.I_{leak}$
01	I_{leak}
10	I_{leak}
11	~0

Fig. 2.14 Several gates in a typical CMOS circuit are automatically stacked, and in low-leakage state. The problem of finding low-leakage input vector is to find the inputs that will put most of the transistors in a low-leakage state

- The positive volatage V_m acts as a body bias to increase the threshold voltage of N1, again resulting in reducing the subthreshold leakage.
- The drain to source potential of N1 increases due to positive potential V_m, resulting in further increase in the threshold voltage, thereby decreasing the leakage current.

Due to this *stacking effect*, leakage power of a CMOS gate depends heavily on the input vector. For example, consider the NAND gate schematic in Fig. 2.14. When the inputs *I0 I1* are "00", both P1 and P2 are on, and N1 and N2 are off. Therefore, $I_{leak}^{00} = I_{N1} + I_{N2} \approx 2 * I_{leak}$. When the inputs are "01", N1 is off, but N2 is on. N1 can be treated as shorted, and its leakage current is ignored. Also, since P1 is on and P2 is off, $I_{leak}^{01} = I_{P2} = I_{leak}$. Similarly, when the inputs are "10", $I_{leak}^{10} = I_{P1} = I_{leak}$. Finally, when inputs are "11", N1 and N2 are on, but P1 and P2 are off. Due to the stacking effect, $I_{leak}^{11} \ll I_{leak}$.

Typically, several gates in a large circuit block may already have low leakage due to the stacking effect. Input vector selection seeks to further reduce the leakage of an idle circuit by applying an input vector that gives as small a leakage current as possible.

The problem of finding the low leakage input vector is motivated by demonstrating that the leakage power of circuits can vary by an order of magnitude over a range of randomly chosen input vectors. The idea is that by using minimal additional circuitry, the logic can be modified so that whenever a circuit is in standby mode, its internal state is set to low leakage. When the circuit is reactivated, it is returned to its last valid state. If the idle periods are long enough, this leads to significant power reductions.

2.5.5 Power Gating

Power Gating is an extremely effective scheme for reducing the leakage power of idle circuit blocks. The power (V_{dd}) to circuit blocks that are not in use is temporarily turned off to reduce the leakage power. When the circuit block is required for operation, power is supplied once again. During the temporary shutdown time, the circuit block is not operational – it is in *low power* or *inactive* mode. Thus, the goal of power gating is to minimize leakage power by temporarily cutting-off power to selective blocks that are not active.

As shown in Fig. 2.15, power gating is implemented by a pMOS transistor as a header switch to shut off power supply to parts of a design in standby or sleep mode. nMOS footer switches can also be used as sleep transistors. Inserting the sleep transistors splits the chip's power network into two parts: a *permanent power*

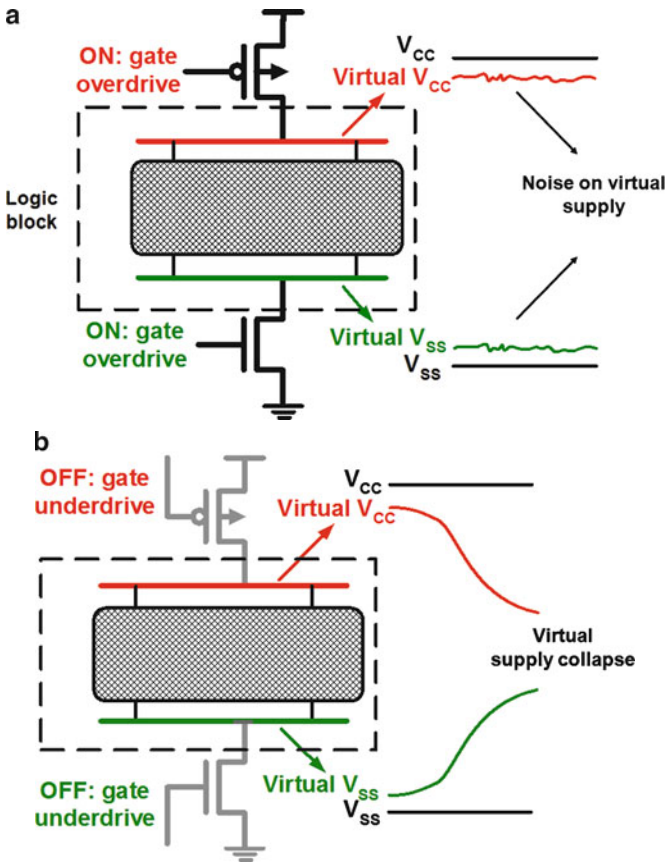


Fig. 2.15 (**a**) Active mode: in the "on" state, the circuit sees a virtual V_{cc} and virtual V_{ss}, which are very close to the actual V_{cc}, and V_{ss} respectively. (**b**) Idle mode: in the "off" state, both the virtual V_{cc} and virtual V_{ss} go to a floating state.

network connected to the power supply and a *virtual power network* that drives the cells and can be turned off.

The biggest challenge in power gating is the size of the power gate transistor. The power gate size must be selected to handle the required amount of switching current at any given time. The gate must be big enough such that there is no measurable voltage (IR) drop due to it. Generally, we use 3X the switching capacitance for the gate size as a rule of thumb.

Since the power gating transistors are rather large, the slew rate is also large, and it takes more time to switch the circuit on and off. This has a direct implication on the effectiveness of power gating. Since it takes a long time for the power-gated circuit to transition in and out of the low power mode, it is not profitable to power gate large circuits for short idle durations. This implies that either we implement power gating at fine granularity, which increases the overhead of gating, or find large idle durations for coarse-grain power gating, which are fewer and more difficult to discover. In addition, coarse-grain power gating results in a large switched capacitance, and the resulting rush current can compromise the power network integrity. The circuit needs to be switched in stages in order to prevent this. Finally, since power gates are made of active transistors, the leakage of the power gating transistor is an important consideration in maximizing power savings.

For fine-grain power-gating, adding a sleep transistor to every cell that is to be turned off imposes a large area penalty. Fine-grain power gating encapsulates the switching transistor as a part of the standard cell logic. Since switching transistors are integrated into the standard cell design, they can be easily be handled by EDA tools for implementation. Fine-grain power gating is an elegant methodology resulting in up to 10X leakage reduction.

In contrast, the coarse-grained approach implements the grid style sleep transistors which drive cells locally through shared virtual power networks. This approach is less sensitive to process variations, introduces less IR-drop variation, and imposes a smaller area overhead than the fine-grain implementations. In coarse-grain power gating, the power-gating transistor is a part of the power distribution network rather than the standard cell.

2.6 Summary

Power considerations led to the replacement of bipolar logic by CMOS in the 1980s in spite of the former resulting in smaller and faster devices. Continuous advacements in CMOS technology enabled an exponential the scaling down of transistor area and scaling up of switching speed over the decades. Unfortunately, the power density increased prominently as a consequence. Leakage currents are expected to result in further increases in power density for technology nodes below 45 nm, which is becoming unsustainable with currently available solutions. This sets up the stage for investigation into aggressive power reduction techniques at every level of design abstraction.

Through this chapter we have introduced various fundamental device and circuit level techniques as well as power management techniques for low power CMOS technology. We discussed various sources of power consumption in a CMOS transistor. We introduced the major components of power: dynamic, short-circuit, and leakage, followed by the basics of device and circuit level techniques to reduce these power components. Techniques for power optimizations at higher levels of design abstraction (microarchitectural, compiler, operating systems, etc.), discussed in later chapters, build upon the foundations laid here.

References

1. Agarwal, A., Kang, K., Bhunia, S.K., Gallagher, J.D., Roy, K.: Effectiveness of low power dual-Vt designs in nano-scale technologies under process parameter variations. In: ISLPED '05: Proceedings of the 2005 international symposium on Low power electronics and design, pp. 14–19. ACM, New York, NY, USA (2005). DOI http://doi.acm.org/10.1145/1077603.1077609
2. Brews, J.R.: The Submicron MOSFET, Chapter 3 in S.M. Sze, editor, High Speed Semiconductor Devices. John Wiley & Sons, New York (1990)
3. Choi, Y.K., Ha, D., King, T.J., Bokor, J.: Investigation of gate-induced drain leakage (gidl) current in thin body devices: single-gate ultra-thin body, symmetrical double-gate, and asymmetrical double-gate mosfets. Japan Journal of Applied Physics part I pp. 2073–2076 (2003)
4. Cox, D., Miller, H.: The Theory of Stochastic Processes. Chapman Hall (1965)
5. Davidson, E.: Packaging technology for the IBM 3090 series systems. In: IEEE Computer Society Spring Workshop (1985)
6. Du, D.H., Yen, S.H., Ghanta, S.: On the general false path problem in timing analysis. In: DAC '89: Proceedings of the 26th ACM/IEEE Design Automation Conference, pp. 555–560. ACM, New York, NY, USA (1989). DOI http://doi.acm.org/10.1145/74382.74475
7. Garey, M.R., Johnson, D.S.: Computers and Intractibility – A Guide to the Theory of NP-Completeness. W.H. Freeman (1979)
8. Ichiba, F., Suzuki, K., Mita, S., Kuroda, T., Furuyama, T.: Variable supply-voltage scheme with 95 In: ISLPED '99: Proceedings of the 1999 international symposium on Low power electronics and design, pp. 54–59. ACM, New York, NY, USA (1999). DOI http://doi.acm.org/10.1145/313817.313849
9. Lengauer, T.: Combinatorial Algorithms for Integrated Circuit Layout. Verlag (1990)
10. Lin, H.R., Hwang, T.T.: Power reduction by gate sizing with path-oriented slack calculation. In: Design Automation Conference, 1995. Proceedings of the ASP-DAC '95/CHDL '95/VLSI '95., IFIP International Conference on Hardware Description Languages; IFIP International Conference on Very Large Scale Integration., Asian and South Pacific, pp. 7–12 (1995). DOI 10.1109/ASPDAC.1995.486194
11. Lyman, J.: Supercomputers demand innovation in packaging and cooling. Electronics Maganize pp. 136–143 (1982)
12. Moore, G.E.: Cramming more components onto integrated circuits. Electronics Maganize **38**(8) (1965)
13. Najm, F.N., Goel, S., Hajj, I.N.: Power estimation in sequential circuits. In: DAC '95: Proceedings of the 32nd annual ACM/IEEE Design Automation Conference, pp. 635–640. ACM, New York, NY, USA (1995). DOI http://doi.acm.org/10.1145/217474.217602
14. Neamen, D.A.: Semiconductor Physics and Devices: Basic Principles. Tata McGraw Hill Publishing Company (1992)
15. Nöth, W., Kolla, R.: Spanning tree based state encoding for low power dissipation. In: DATE '99: Proceedings of the conference on Design, automation and test in Europe, p. 37. ACM, New York, NY, USA (1999). DOI http://doi.acm.org/10.1145/307418.307482

16. Oktay, S.., Kammerer, H.C.: A conduction-cooled module for high performance LSI devices. IBM Journal of Research and Development **26**(1), 55–56 (1982)
17. Pierret, R.F.: Semiconductor Device Fundamentals. Addison-Wesley, Reading, MA (1996)
18. Puri, R.: Minimizing power under performance constraint. In: International Conference on Integrated Circuit Design and Technology, 2004. ICICDT '04., pp. 159–163 (2004)
19. Ramadhyani, S.., Incropera, F.P.: Forced convection cooling of discrete heat sources with or without surface enhancement. In: International Symposium on Cooling Technology for Electronic Equipment, pp. 249–264 (1987)
20. Tsividis, Y.P.: Operation and Modeling of the MOS Transistor. McGraw-Hill, New York (1987)
21. Tuckerman, D., Pease, F.: High-performance heat sinking for VLSI. IEEE Electron Device Letters **EDL-2**(5), 126–129 (1981)
22. Veendrick, H.: Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits. Solid-State Circuits, IEEE Journal of **19**(4), 468–473 (1984)
23. Watari, T., Murano, H.: Packaging technology for the NEX SX supercomputer. In: Electronic Components Conference, pp. 192–198 (1985)
24. Wu, P., Little, W.A.: Measurement of the heat transfer characteristics of gas flow in fine channel heat exchangers used for microminiature refrigerators (1984)
25. Yamamoto, H., Udagawa, Y., Okada, T.: Cooling and packaging technology for the FACOM M-780. Fujistsu Journal **37**(2), 124–134 (1986)
26. Yen, H.C., Ghanta, S., Du, H.C.: A path selection algorithm for timing analysis. In: DAC '88: Proceedings of the 25th ACM/IEEE Design Automation Conference, pp. 720–723. IEEE Computer Society Press, Los Alamitos, CA, USA (1988)