Unit 3

Sources of Power Dissipation

6.1 Introduction

In order to develop techniques for minimizing power dissipation, it is essential to identify various sources of power dissipation and different parameters involved in each of them. Power dissipation may be specified in two ways. One is maximum power dissipation, which is represented by "peak instantaneous power dissipation." Peak instantaneous power dissipation occurs when a circuit draws maximum power, which leads to a supply voltage spike due to resistances on the power line. Glitches may be generated due to this heavy flow of current and the circuit may malfunction, if proper care is not taken to suppress power-line glitches. The second one is the "average power dissipation," which is important in the context of battery-operatedportable devices. The average power dissipation will decide the battery lifetime. Here, we will be concerned mainly with the average power dissipation, although the techniques used for reducing the average power dissipation will also lead to the reduction of peak power dissipation and improve reliability by reducing the possibility of power-related failures.

In CMOS circuits, power dissipation can be divided into two broad categories: *dynamic* and *static*. Dynamic power dissipation in CMOS circuits occur when the circuits are in working condition or active mode, that is, there are changes in input and output conditions with time. In this section, we introduce the following three basic mechanisms involved in dynamic power dissipation:

- *Short-circuit power:* Short-circuit power dissipation occurs when both the nMOS and pMOS networks are ON. This can arise due to slow rise and fall times of the inputs as discussed in Sect. 6.2.

- *Switching power dissipation:* As the input and output values keep on changing, capacitive loads at different circuit points are charged and discharged, leading to power dissipation. This is known as *switching power* dissipation. Until recently, this was the most dominant source of power dissipation. The switching power dissipation is discussed in Sect. 6.3.

- *Glitching power dissipation:* Due to a finite delay of the logic gates, there are spurious transitions at different nodes in the circuit. Apart from the abnormal behavior of the circuits, these transitions also result in power dissipation known as glitching power dissipation. This is discussed in Sect. 6.4.

6.2 Short-Circuit Power Dissipation

When there are finite rise and fall times at the input of CMOS logic gates, both pMOS and nMOS transistors are simultaneously ON for a certain duration, shorting the power supply line to ground. This leads to current flow from supply to ground. Short-circuit power dissipation takes place for input voltage in the range $V_{tn} < V_{in} < V_{dd} - |V_{tp}|$, when both pMOS and nMOS transistors turn ON creating a conducting path between V_{dd} and ground (GND). It is analyzed in the case of a CMOS inverter as shown in Fig. 6.2. To estimate the average short-circuit current, we have used simple

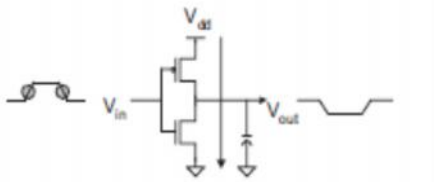


Fig. 6.2 Short-circuit power dissipation during input transition

model shown in Fig. 6.3. It is assumed that τ is both rise and fall times of the input ($\tau_r = \tau_f = \tau$) and the inverter is symmetric, i.e., $\beta_n = \beta_p = \beta$ and $V_{tn} = -V_{tp} = V_t$.

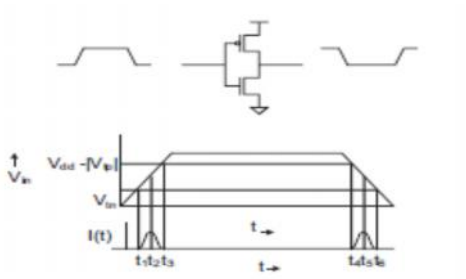


Fig. 6.3 Model for short-circuit power dissipation

The short-circuit power is given by

$$P_{sc} = V_{dd} I_{mean} = \frac{\beta}{12}(V_{dd} - 2V_t)^3 \tau f = \frac{\beta}{12} V_{dd}^3 \left(1 - 2\frac{V_t}{V_{dd}}\right)^3 \tau f. \qquad (6.7)$$

As the clock frequency decides how many times the output changes per second, the short-circuit power is proportional to the frequency. The short-circuit current is also proportional to the rise and fall times. Short-circuit currents for different input slopes are shown in Fig. 6.4. The power supply scaling affects the short-circuit power considerably because of cubic dependence on the supply voltage.

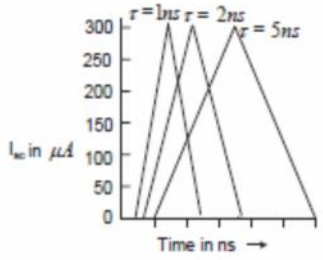


Fig. 6.4 Short-circuit current as a function of input rise/fall time

such a situation, the short-circuit current will be very small. It is maximum when there is no load capacitance. The variation of short-circuit current for different out-put capacitances is shown in Fig. 6.5. From this analysis, it is evident that the short-circuit power dissipation can be minimized by making the output rise/fall times smaller. The short-circuit power dissipation is also reduced by increasing the load capacitance. However, this makes the circuit slower. One good compromise is to have equal input and output slopes. Because of the cubic dependence of the short-circuit power on supply voltage, the supply voltage may be scaled to reduce short-circuit power dissipation.

We may conclude this subsection by stating that the short-circuit power dissipation depends on the input rise/fall time, the clock frequency, the load capacitance, gate sizes, and above all the supply voltage.

6.3 Switching Power Dissipation

There exists capacitive load at the output of each gate. The exact value of capacitance depends on the fan-out of the gate, output capacitance, and wiring capacitances and all these parameters depend on the technology generation in use. As the output changes from a low to high level and high to low level, the load capacitor charges and discharges causing power dissipation. This component of power dis-sipation is known as switching power dissipation.

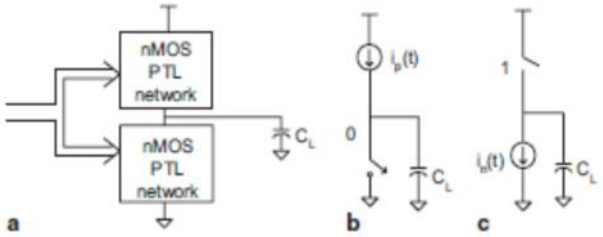


Fig. 6.8 Dynamic power dissipation model

Switching power dissipation can be estimated based on the model shown in Fig. 6.8. Figure 6.8a shows a typical CMOS gate driving a total output load capacitance $C_{L.}$ For some input combinations, the pMOS network is ON and nMOS network is OFF as modeled in Fig. 6.8b. In this state, the capacitor is charged to V_{dd} by drawing power from the supply. For some other input combinations, the nMOS network is ON and pMOS network is OFF, which is modeled in Fig. 6.8c. In this state, the capacitor discharges through the nMOS network. For simplicity, let us assume that the CMOS gate is an inverter. This implies that half of the energy is stored in the capacitor, and the remaining half $(1 / 2)C_L V_{dd}^2$ is dissipated in the pMOS transistor network. During the V_{dd} to 0 transition at the output, no energy is drawn from the power supply and the charge stored in the capacitor is discharged in the nMOS transistor network.

If a square wave of repetition frequency f (I/T) is applied at the input, average power dissipated per unit time is given by

$$P_d = \frac{1}{T} C_L V_{dd}^2 = C_L V_{dd}^2 f. \qquad (6.12)$$

The switching power is proportional to the switching frequency and independent of device parameters. As the switching power is proportional to the square of the sup-ply voltage, there is a strong dependence of switching power on the supply voltage. Switching power reduces by 56 %, if the supply voltage is reduced from 5 to 3.3 V, and if the supply voltage is lowered to 1 V, the switching power is reduced by 96 % compared to that of 5 V. This is the reason why voltage scaling is considered to be the most dominant approach to reduce switching power.

6.3.1 Dynamic Power for a Complex Gate

For an inverter having a load capacitance C_L, the dynamic power expression is $C_L V_{dd}^2 f$. Here, it is assumed that the output switches from rail to rail and input switching occurs for every clock. This simple assumption does not hold good for complex gates because of several reasons. First, apart from the output load capacitance, there exist capacitances at other nodes of the gate. As these internal nodes also charge and discharge, dynamic power dissipation will take place on the internal nodes. This leads to two components of dynamic power dissipation-load power and internal node power. Second, at different nodes of a gate, the voltage swing may not be from rail to rail. Finally, to take into account the condition when the capacitive node of a gate might not switch when the clock is switching, a concept known as *switching activity* is introduced.

Switching activity determines how often switching occurs on a capacitive node. These three issues are considered in the following subsections.

6.3.2 Reduced Voltage Swing

There are situations where a rail-to-rail swing does not take place on a capacitive node. This situation arises in pass transistor logic and when the pull-up device is an enhancement-type nMOS transistor in nMOS logic gates as shown in Fig. 6.9. In such cases, the output can only rise to $V_{dd} - V_t$. This situation also happens in interval nodes of CMOS gates. Instead of $C_L V_{dd}^2$ for full-rail charging, the energy drawn from power supply for charging the capacitance to $(V_{dd} - V_t)$ is given by

$$E_{0\to1} = C_L V_{dd}(V_{dd} - V_t). \tag{6.13}$$

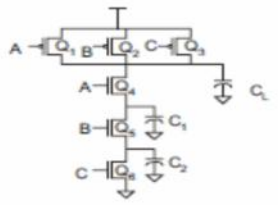
6.3.3 Internal Node Power

A three-input NAND gate is shown in Fig. 6.10. Apart from the output capacitance $C_{L,}$ two capacitances C_1 and C_2 are shown in two interval nodes of the gate. For input combination 110, the output is "1" and transistors Q_3, Q_4, and Q_5 are ON. All the capacitors will draw energy from the supply. Capacitor C_L will charge to V_{dd} through Q_3, capacitor C_1 will charge to $(V_{dd} - V_t)$ through Q_3 and Q_4. Capacitor C_2 will also charge to $(V_{dd} - V_t)$ through Q_3, Q_4, and Q_5. For each 0-to-V_{dd} transition at an internal node, the energy drawn is given by

$$E_{0\to1} = C_i V_i V_{dd}, \tag{6.14}$$

where C_i is the internal node capacitance and V_i is internal voltage swing at node i.

Fig. 6.10 Switching nodes of a three-input NAND gate



6.3.4 Switching Activity

For a complex logic gate, the switching activity depends on two factors—the topol-ogy of the gate and the statistical timing behavior of the circuit. To handle the transi-tion rate variation statistically, let $n(N)$ be the number of 0-to-V_{dd} output transitions in the time interval $[0,N]$. Total energy E_N drawn from the power supply for this interval is given by

$$E_N = C_L V_{dd}^2 \times n(N). \tag{6.15}$$

The average power dissipation during an extended interval is $P_{avg} = \lim_{N\to\infty} \frac{E_N}{N} \times f$, where f is the clock frequency.

$$P_{avg} = \left(\lim_{N\to\infty} \frac{n(N)}{N} \right) C_L V_{dd}^2 f. \tag{6.16}$$

6.4 Glitching Power Dissipation

In the power calculations so far, we have assumed that the gates have zero delay. In practice, the gates will have finite delay and this delay will lead to spurious undesirable transitions at the output. These spurious signals are known as *glitches.* In the case of a static CMOS circuit, the output node or internal nodes can make undesirable transitions before attaining a stable value. Consider the circuit shown in Fig. 6.15. If the inputs ABC change value from 101 to 000, ideally for zero gate delay the output should remain at the 0 logic level. However, considering unit gate delay of the first gate stage, output O_1 is delayed compared to the C input. As a consequence, the output switches to 1 logic level for one gate delay duration. This transition increases the dynamic power dissipation and this component of dynamic power is known as *glitching power*. Glitching power may constitute a significant portion of dynamic power, if circuits are not properly designed.

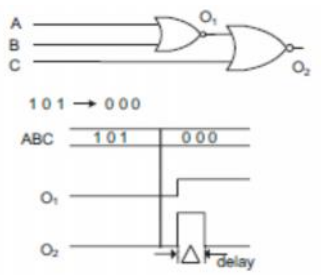


Fig. 6.15 Output waveform showing glitch at output O_2

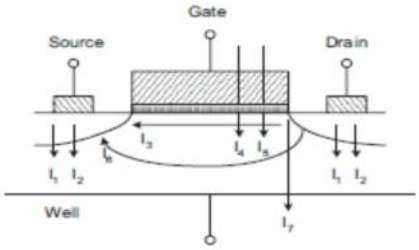
Usually, cascaded circuits as shown in Fig. 6.16a exhibit high glitching power. The glitching power can be minimized by realizing a circuit by balancing delays, as shown in Fig. 6.16b. On highly loaded nodes, buffers can be inserted to balance delays and cascaded implementation can be avoided, if possible, to minimize glitching power.

Fig. 6.16 Realization of A, B, C, and D, **a** in cascaded form, **b** balanced realization

6.5 Leakage Power Dissipation

When the circuit is not in an active mode of operation, there is static power dissipation due to various leakage mechanisms. In deep-submicron devices, these leak-age currents are becoming a significant contributor to power dissipation of CMOS circuits. Figure 6.17 illustrates the seven leakage mechanisms. Here, I_1 is the *re-verse-bias* p–n *junction diode leakage current*; I_2 is the reverse-biased p–n junction current due to *tunneling* of electrons from the valence bond of the *p* region to the conduction bond of the n region; I_3 is the *subthreshold leakage current* between the source and the drain when the gate voltage is less than the threshold voltage V_t; I_4 is the *oxide-tunneling current* due to a reduction in the oxide thickness; I_5 is gate current due to *hot-carrier injection* of elections; I_6 is the *GIDL current* due to a high field effect in the drain junction; and I_7 is the *channel punch-through current* due to the close proximity of the drain and the source in short-channel devices. These leakage components are discussed in the following subsections.

Fig. 6.17 Summary of leak-age current mechanisms of deep-submicron transistors



6.5.1 p–n Junction Reverse-Biased Current

Let us consider the physical structure of a CMOS inverter shown in Fig. 6.18. As shown in the figure, source–drain diffusions and n-well diffusions form parasitic diodes in the bulk of silicon substrate. As parasitic diodes are reverse-biased, their leakage currents contribute to static power dissipation. The current for one diode is given by

$$I_{ndiode} = AJ_s\left[e^{-\frac{qV_d}{nKT}} - 1\right], \qquad (6.29)$$

where J_s is the reverse saturation current density (this increases with temperature), I_s is the AJ_s, V_d is the diode voltage, n is the emission coefficient of the diode (some-times equal to 1), q is the charge of an electron (1.602×10^{-19}), K is Boltzmann constant

At room temperature, $V_T \approx 26mV$.

The leakage current approaches I_s, the reverse saturation current even for a small reverse-biased voltage across the diode. The reverse saturation current per unit area is defined as the current density J_s, and the resulting current is approximately $I_L = A_D \cdot J_s$, where A_D is area of drain diffusion. For a typical CMOS process, $J_s \approx 1-5 \text{pA}/\mu\text{m}^2$ at room temperature, and the A_D is about 6 μm^2 for a 1.0-μm minimum feature size. It leads to a leakage current of about 1 fA per device at room temperature. The reverse diode leakage current I_{sdlc} increases significantly with temperature. The total static diode leakage current for n devices is given by $I_{rdlc} = \sum_{i=1}^{n} I_i$ and the total static power dissipation for n devices is equal to

$$P = V_{dd} \cdot \sum_{i=1}^{n} I_i. \tag{6.30}$$

Then, the total static power dissipation due to diode leakage current for one million transistors is given by

$$P = V_{dd} \sum_{i=1}^{10^6} I_{di} \approx 0.01\mu\text{W}. \tag{6.31}$$

6.5.2 Band-to-Band Tunneling Current

When both n regions and p regions are heavily doped, a high electric field across a reverse biased p–n junction causes a significant current to flow through the junction due to tunneling of electrons from the valence band of the p region to the conduction band of n region. This is illustrated in Fig. 6.19. It is evident from this diagram that for the voltage drop across that junction should be more than the band gap.

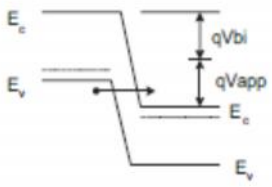


Fig. 6.19 BTBT in reverse-biased p–n junction

6.5.3 Subthreshold Leakage Current

The subthreshold leakage current in CMOS circuits is due to carrier diffusion between the source and the drain regions of the transistor in weak inversion, when the gate voltage is below V_t. The behavior of an MOS transistor in the subthreshold operating region is similar to a bipolar device, and the subthreshold current exhibits an exponential dependence on the gate voltage. The amount of the subthreshold current may become

significant when the gate-to-source voltage is smaller than, but very close to, the threshold voltage of the device.

Various mechanisms which affect the subthreshold leakage current are:

- Drain-induced barrier lowering (DIBL)
- Body effect
- Narrow-width effect
- Effect of channel length and V_{th} roll-off
- Effect of temperature

6.5.3.1 Drain-Induced Barrier Lowering

For long-channel devices, the sources and drain region are separated far apart and the depletion regions around the drain and source have little effect on the potential distribution in the channel region. So, the threshold voltage is independent of the channel length and drain bias for such devices. However, for short-channel devices, the source and drain depletion width in the vertical direction and the source drain potential have a strong effect on a significant portion of the device leading to variation of the subthreshold leakage current with the drain bias. This is known as the DIBL effect. Because of the DIBL effect, the barrier height of a short-channel device reduces with an increase in the subthreshold current due to a lower threshold voltage.

DIBL occurs when the depletion regions of the drain and the source interact with each other near the channel surface to lower the source potential barrier. Figure 6.19 shows the lateral energy band diagram at the surface versus distance from the source to the drain. It is evident from the figure that DIBL occurs for short-channel lengths and it is further enhanced at high drain voltages. Ideally, the DIBL effect does not change the value of S_t, but does lower V_{th}.

6.5.3.2 Body Effect

As a negative voltage is applied to the substrate with respect to the source, the well-to-source junction, the device is reverse-biased and bulk depletion region is widened. This leads to an increase in the threshold voltage. This effect is known as the body effect. The threshold voltage equation given below gives the relationship of the threshold voltage with the body bias

$$V_{th} = V_{fb} + 2\psi_B + \frac{\sqrt{2\varepsilon_{si} q N_a (2\psi_B + V_{bs})}}{C_{ox}},$$

where V_{fb} is the flat-band voltage, N_a is the doping density in the substrate, and $\psi_B = \left(\frac{KT}{\varepsilon}\right) \ln\left(N_a / n_i\right)$ is the difference between the Fermi potential and the intrinsic potential in the substrate.

The variation of the threshold voltage with respect to the substrate bias dV_{th} / dV_{bs} is referred to as the substrate sensitivity:

$$\frac{dV_{th}}{dV_{bs}} = \frac{\sqrt{\frac{\varepsilon_{si} q N_a}{2(2\psi_B + V_{bs})}}}{C_{ox}}.$$

Supply Voltage Scaling for Low Power

7.1 Introduction

In the preceding chapter, various sources of power dissipation in complementary metal–oxide–semiconductor (CMOS) circuits have been discussed. The total power dissipation can be represented by the simplified equation:

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{static}} \qquad (7.1)$$

7.2 Device Feature Size Scaling

Continuous improvements in process technology and photolithographic techniques have made the fabrication of metal–oxide–semiconductor (MOS) transistors of smaller and smaller dimensions to provide a higher packaging density. As a reduction in feature size reduces the gate capacitance, this leads to an improvement in performance. This has opened up the possibility of scaling device feature sizes to compensate for the loss in performance due to voltage scaling. The reduction of the size, i.e., the dimensions of metal–oxide–semiconductor field-effect transistors (MOSFETs), is commonly referred to as *scaling*. To characterize the process of scaling, a parameter *S*, known as *scaling factor*, is commonly used. All horizontal and vertical dimensions are divided by this scaling factor, $S > 1$, to get the dimensions of the devices of the new generation technology. Obviously, the extent of scaling, in other words the value of *S*, is decided by the minimum feature size of the prevalent technology. It has been observed that over a period of every 2 to 3 years, a new generation technology is introduced by downsizing the device dimensions by a factor of *S*, lying in the range 1.2–1.5.

Figure 7.3 shows the basic geometry of an MOSFET and the various parameters scaled by a scaling factor *S*. It may be noted that all the three dimensions are proportionally reduced along with a corresponding increase in doping densities. There are two basic approaches of device size scaling–*constant-field scaling* and *constant-voltage scaling*. In constant-field scaling, which is also known as *full scaling*, the supply voltage is also scaled to maintain the electric fields same as the previous generation technology as shown in Fig. 7.2d. In this section, we examine, in detail, both the scaling strategies and their effect on the vital parameters of an MOSFET.

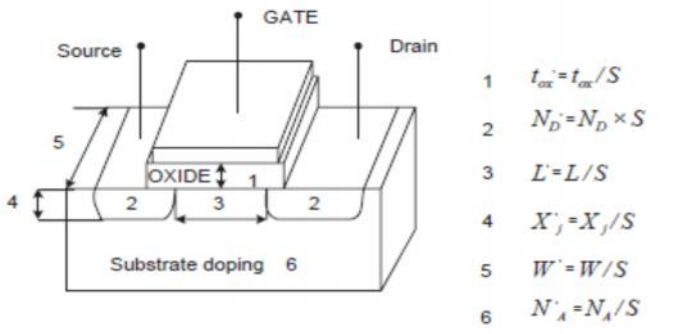


Fig. 7.3 Scaling of a typical metal–oxide–semiconductor field-effect transistors (MOSFET) by a scaling factor *S*

7.2.1 Constant-Field Scaling

In this approach, the magnitudes of all the internal electric fields within the device are preserved, while the dimensions are scaled down by a factor of *S*. This requires that all potentials must be scaled down by the same factor. Accordingly, supply and threshold voltages are scaled down proportionately. This also dictates that the doping densities are to be increased by a factor of *S* to preserve the field conditions. A list of scaling factors for all device dimensions, potentials, and doping densities are given in Table 7.2.

Table 7.2 Constant-field scaling of the device dimensions, voltages, and doping densities

Quantity	Before scaling	After scaling
Channel length	L	$L' = L/S$
Channel width	W	$W' = W/S$
Gate oxide thickness	t_{ox}	$t'_{ox} = t_{ox}/S$
Junction depth	x_j	$x'_j = x_j/S$
Power supply voltage	V_{dd}	$V'_{dd} = V_{dd}/S$
Threshold voltage	V_{T0}	$V'_{T0} = V_{T0}/S$
Doping densities	N_A	$N'_A = N_A \cdot S$
	N_D	$N'_D = N_D \cdot S$

7.2.2 Constant-Voltage Scaling

In constant-voltage scaling, all the device dimensions are scaled down by a factor of *S* just like constant-voltage scaling. However, in many situations, scaling of supply voltage may not be feasible in practice. For example, if the supply voltage of a cen-tral processing unit (CPU) is scaled down to minimize power dissipation, it leads to electrical compatibility with peripheral devices, which usually operate at higher supply voltages. It may be necessary to use multiple supply voltages and compli-cated-level translators to resolve this problem. In such situations, constant-voltage scaling may be preferred. In a constant-voltage scaling approach, power supply voltage and the threshold voltage of the device remain unchanged. To preserve the charge–field relations, however, the doping densities have to be scaled by a factor of S^2. Key device dimensions, voltages, and doping densities for constant-voltage scaling are shown in Table 7.4.

Constant-voltage scaling results in an increase in drain current (both in linear mode and in saturation mode) by a factor of *S*. This, in turn, results in an increase in the power dissipation by a factor of *S* and the power density by a factor of S^3, as shown in Table 7.5. As there is no decrease in delay, there is also no improvement in performance. This increase in power density by a factor of S^3 has possible adverse effects on reliability such as electromigration, hot-carrier degradation, oxide break-down, and electrical overstress.

Table 7.4 Constant-voltage scaling of the device dimensions, voltages, and doping densities

Quantity	Before scaling	After scaling
Channel length	L	$L' = L/S$
Channel width	W	$W' = W/S$
Gate oxide thickness	t_{ox}	$t'_{ox} = t_{ox}/S$
Junction depth	x_j	x'_j/s
Power supply voltage	V_{dd}	$V'_{dd} = V_{dd}$
Threshold voltage	V_{to}	$V'_{t0} = V_{t0}$
Doping densities	N_A	$N'_A = N_A \cdot S^2$
	N_D	$N'_D = N_D \cdot S^2$

Table 7.5 Effects of constant-voltage scaling on the key device parameters

Quality	Before scaling	After scaling
Gate capacitance	C_g	$C'_g = C_g/S$
Drain current	I_D	$I'_D = I_D \cdot S$
Power dissipation	P	$P' = P \cdot S$
Power density	P/area	P' / area′ = $S^3 P$/area
Delay ($t_d \propto C_g \cdot V_{dd}/I_{ds}$)	t_d	$t'_d = t_d/S^2$

$$I'(\text{lin}) = S \cdot I(\text{lin})$$
$$I'_0(\text{sat}) = S \cdot I_D(\text{sat}), \tag{7.9}$$

$$P' = I'_0 \cdot V'_{ds} = (SI_D)V_{ds} = S \cdot P. \tag{7.10}$$

7.3 Architectural-Level Approaches

Architectural-level refers to register-transfer-level (RTL), where a circuit is rep-resented in terms of building blocks such as adders, multipliers, read-only memo-ries (ROMs), register files, etc. High-level synthesis technique transforms a behavioral-level specification to an RTL-level realization. It is envisaged that low-power synthesis technique on the architectural level can have a greater impact than that of gate-level approaches. Possible architectural approaches are: parallelism, pipelining, and power management, as discussed in the following subsections.

7.3.1 Parallelism for Low Power

Parallel processing is traditionally used for the improvement of performance at the expense of a larger chip area and higher power dissipation. Basic idea is to use multiple copies of hardware resources, such as arithmetic logic units (ALUs) and processors, to operate in parallel to provide a higher performance. Instead of using parallel processing for improving performance, it can also be used to reduce power. We know that supply voltage scaling is the most effective way to reduce power con-sumption.

7.3.2 Multi-Core for Low Power

The idea behind the parallelism for low power can be extended for the realization of multi-core architecture. Figure 7.6 shows a four-core multiplier architecture. Table 7.7 shows how the clock frequency can be reduced with commensurate scaling of the supply voltage as the number of

cores is increased from one to four while maintaining the same throughput. This is the basis of the present-day multi-core commercial processors introduced by Intel, AMD, and other processor manufacturers. Thread-level parallelism is exploited in multi-core architectures to increase throughput of the processors.

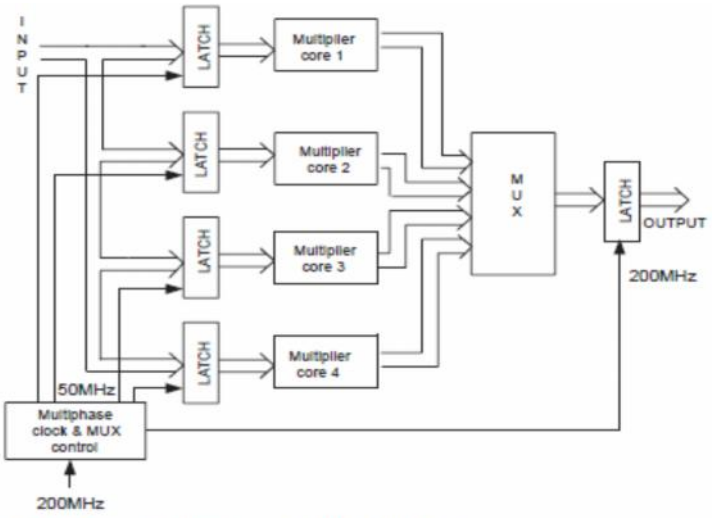


Fig. 7.6 A four-core multiplier architecture. *MUX* multiplexer

Table 7.7 Power in multi-core architecture

Number of cores	Clock in MHz	Core supply voltage	Total power
1	200	5	15.0
2	100	3.6	8.94
4	50	2.7	5.20
8	25	2.1	4.5

7.3.3 Pipelining for Low Power

Instead of reducing the clock frequency, in pipelined approach, the delay through the critical path of the functional unit is reduced such that the supply voltage can be reduced to minimize the power. As an example, consider the pipelined realization of 16-bit adder using two-stage pipeline shown in Fig. 7.7. In this realization, instead of 16-bit addition, 8-bit addition is performed in each stage. The critical path delay through the 8-bit adder stage is about half that of 16-bit adder stage. Therefore, the 8-bit adder will operate at a clock frequency of 100 MHz with a reduced power supply voltage of $V_{ref}/2$. It may be noted that in this realization, the area penalty is much less than the parallel implementation leading to $C_{pipe} = 1.15C_{ref}$. Substituting these values, we get:

$$P_{pipe} = C_{pipe} \cdot V_{pipe}^2 \cdot f_{pipe} = (1.15C_{ref}) \cdot \left(\frac{V_{ref}}{2}\right)^2 \cdot f = 0.28P_{ref}. \qquad (7.16)$$

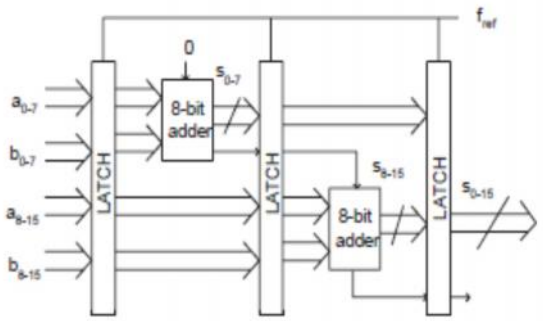


Fig. 7.7 Pipelined realization 16-bit adder

Table 7.8 Impact of pipelining on area, power, and throughput

Parameter	Without V_{dd} scaling	With V_{dd} scaling
Area	1.15X	1.15X
Power	2.30X	0.28X
Throughput	2X	1X

It is evident that the power reduction is very close to that of a parallel implementation with an additional bonus of a reduced area overhead. The impact of pipelining is highlighted in Table 7.8. Here, column 2 shows pipelining for improved performance with larger power dissipation, higher clock frequency, and without voltage scaling, whereas column 3 corresponds to parallelism for low power with voltage scaling and without degradation of performance.

7.3.4 Combining Parallelism with Pipelining

An obvious extension of the previous two approaches is to combine the parallelism with pipelining. Here, more than one parallel structure is used and each structure is pipelined. Figure 7.8 shows the realization of a 16-bit adder by combining both pipelining and parallelism. Two pipelined 16-bit adders have been used in parallel. Both power supply and frequency of operation are reduced to achieve substantial overall reduction in power dissipation:

$$P_{\text{parpipe}} = C_{\text{parpipe}} \cdot V_{\text{parpipe}}^2 \cdot f_{\text{parpipe}} . \qquad (7.17)$$

The effective switching capacitance C_{parpipe} will be more than the previous because of the duplication of functional units and more number of latches. It is assumed to be equal to 2.5 C_{ref}. The supply voltage can be more aggressively reduced to about one quarter of V_{ref} and the frequency of operation is reduced to half the reference frequency $f_{\text{ref}}/2$. Thus,

$$\begin{aligned} P_{\text{parpipe}} &= (2.5C_{\text{ref}})(0.3V_{\text{ref}})^2\left(\frac{f_{\text{ref}}}{2}\right) \\ &= 0.1125P_{\text{ref}} . \end{aligned} \qquad (7.18)$$

7.4 Voltage Scaling Using High-Level Transformations

For automated synthesis of digital systems, high-level transformations such as dead code elimination, common sub-expression elimination, constant folding, in-line expansion, and loop unrolling are typically used to optimize the design parameters such as the area and throughput [4]. These high-level transformations can also be used to reduce the power consumption either by reducing the supply voltage or the switched capacitance. In this section, we discuss how loop unrolling can be used to minimize power by voltage scaling.

7.5 Multilevel Voltage Scaling

As high V_{dd} gates have a less delay, but higher dynamic and static power dissipation, devices on time-critical paths can be assigned higher V_{dd}, while devices on noncritical paths shall be assigned lower V_{dd}, such that the total power consumption can be reduced without degrading the overall circuit performance [5]. Figure 7.14 shows that the delay on the critical path is 10 ns, whereas delay on the noncritical path is 8 ns. The gates on the critical path are assigned higher supply voltage V_{ddH}. The slack of the noncritical path can be traded for lower switching power by assigning lower supply voltage V_{ddL} to the gates on the noncritical path.

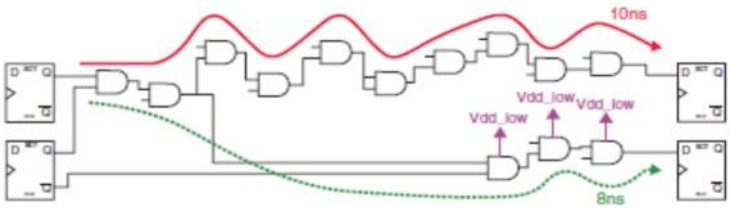


Fig. 7.14 Assignment of multiple supply voltages based on delay on the critical path

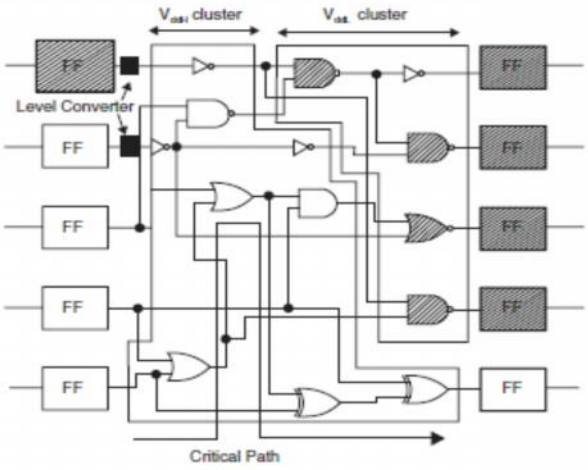


Fig. 7.15 Clustered voltage scaling. *FF* flip-flop

For multiple dual-V_{dd} designs, the voltage islands can be generated at different levels of granularity, such as macro level and standard cell level. In the standard-cell level, gates on the critical path and noncritical paths are clustered into two groups. Gates on the critical path are selected from the higher supply voltage (V_{ddH}) stan-dard cell library, whereas gates on the noncritical path are selected from the lower supply voltage (V_{ddL}) standard cell library, as shown in Fig. 7.15. This approach modifies the normal distribution of path delays using a single supply voltage in a design to a distribution of path delays skewed toward higher delay with multiple supply voltages, as shown in Fig. 7.16.

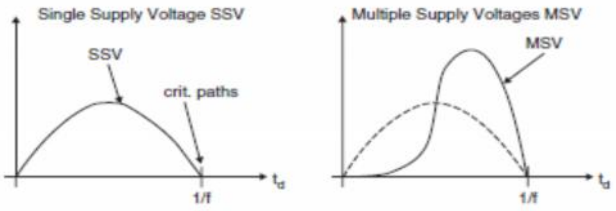


Fig. 7.16 Distribution of path delays under single supply voltage (SSV) and multiple supply voltage (MSV)

7.6 Challenges in MVS

The overhead involved with multiple-V_{dd} systems includes the additional power supply networks, insertion of level converters, complex characterization and static timing analysis, complex floor planning and routing, and power-up–power-down sequencing. As a consequence, even a simple multi-voltage design presents the de-signer with a number of challenges, which are highlighted in this section.

7.6.1 Voltage Scaling Interfaces

When signals go from one voltage domain to another voltage domain, quite often, it is necessary to insert level converters or shifters that convert the signals of one volt-age level to another voltage level. Consider a signal going from a low-V_{dd} domain to a high-V_{dd} domain, as shown in Fig. 7.18. A high-level output from the low-V_{dd} domain has an output V_{ddL}, which may turn on both nMOS and pMOS transistors of the high-V_{dd} domain inverter resulting in a short circuit between V_{ddH} and the GND. A level converter needs to be inserted to avoid this static power consumption. Moreover, to avoid the significant rise and fall time degradations between the voltage-domain boundaries, it is necessary to insert buffers to improve the quality of signals that go from one domain to another domain with proper voltage swing and rise and fall times. So, it may be necessary to insert buffers even when signals go from high- to low-voltage domain. This approach of clean interfacing helps to maintain the timing characteristics and improves ease of reuse.

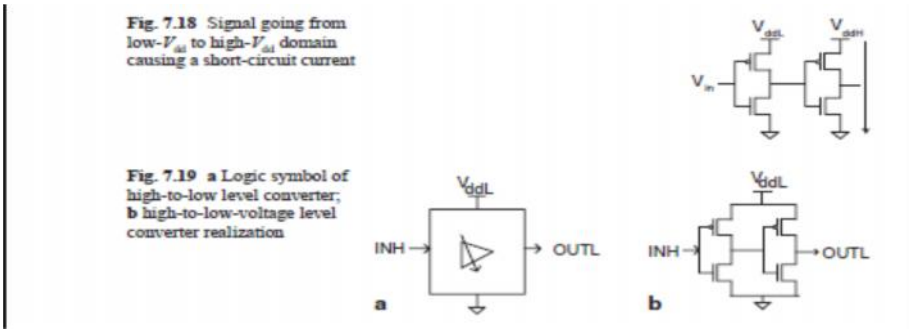


Fig. 7.18 Signal going from low-V_{dd} to high-V_{dd} domain causing a short-circuit current

Fig. 7.19 **a** Logic symbol of high-to-low level converter; **b** high-to-low-voltage level converter realization

High-to-Low-Voltage Level Converters The need for a level converter as a sig-nal passes from high-V_{dd} domain to low-V_{dd} domain arises primarily to provide a clean signal having a desired voltage swing and rise and fall times. Without a level converter, the voltage swing of the signal reaching the low-V_{dd} domain is 0 to V_{ddH}. This causes higher switching power dissipation and high leakage power dissipa-tion due to GIDL effect. Moreover, because of the longer wire length between the voltage domains, the rise and fall time may be long leading to increase in short-circuit power dissipation. To overcome this problem, a level converter as shown in Fig. 7.19b may be inserted. The high-to-low level converter is essentially two inverter stages in cascade. It introduces a buffer delay and its impact on the static timing analysis is small.

Low-to-High-Voltage Level Converters Driving logic signals from a low-voltage domain to high-voltage domain is a more critical problem because it has significant degrading effect on the operation of a circuit. The logic symbol of the low-to-high-voltage level converter is shown in Fig. 7.20a.

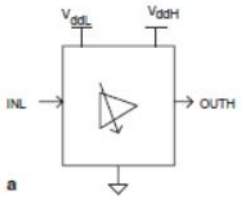


Fig. 7.20 a Logic symbol of low-to-high level converter;

7.6.2 Converter Placement

One important design decision in the voltage scaling interfaces is the placement of converters. As the high-to-low level converters use low-V_{dd} voltage rail, it is appropriate to place them in the receiving or destination domain, that is, in the low-V_{dd} domain. This not only avoids the routing of the of the low-V_{dd} supply rail from the low-V_{dd} domain to the high-V_{dd} domain but also helps in the improvement of the rise and fall time of the signals in low-V_{dd} domain. Placement of high-to-low level converter is shown in Fig. 7.21a. It is also recommended to place the low-to-

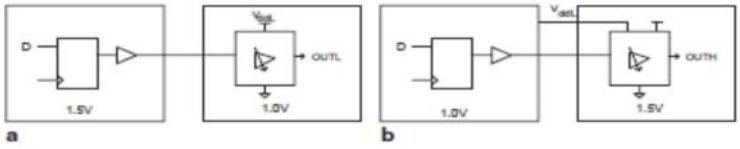


Fig. 7.21 a High-to-low converter placement; b low-to-high converter placement

high level converters in the receiving domain, that is, in the high-V_{dd} domain. This, however, involves routing of the low-V_{dd} supply rail to the high-V_{dd} domain. As the low-to-high level converters require both low- and high-V_{dd} supply rails, at least one of the supply rails needs to be routed from one domain to the other domain. The placement of the low-to-high level converter is shown in Fig. 7.21b.

7.7 Dynamic Voltage and Frequency Scaling

DVFS has emerged as a very effective technique to reduce CPU energy [6]. The technique is based on the observation that for most of the real-life applications, the workload of a processor varies significantly with time and the workload is bursty in nature for most of the applications. The energy drawn for the power supply, which is the integration of power over time, can be significantly reduced. This is particularly important for battery-powered portable systems.

7.7.1 Basic Approach

The energy drawn from the power supply can be reduced by using the following two approaches:

Dynamic Frequency Scaling During periods of reduced activity, there is a scope to lower the operating frequency with varying workload keeping the sup-ply voltage constant. As we know, digital CMOS circuits are used in a majority of microprocessors, and, for present-day digital CMOS.

Dynamic Voltage and Frequency Scaling An alternative approach is to reduce the operating frequency along with the supply voltage without sacrificing the performance required at that instance. It has been established that CMOS circuits can operate over a certain voltage range with reasonable reliability, where frequency increases monotonically with the supply voltage. For a particular process technology, there is a maximum voltage limit beyond which the circuit operation is destructive. Similarly, there is a lower voltage limit below which the circuit operation is unreliable or the delay paths no longer vary monotonically. Within the reliable operating range, the delay increases monotonically with the decrease in supply voltage following Eq. 7.23. Therefore, the propagation delay restricts the clock frequency in a microprocessor:

$$\text{Delay}(D) \propto \frac{V_{dd}}{(V_{dd}-V_t)^2} = \frac{1}{V_{dd}\left(1-\frac{V_t}{V_{dd}}\right)^2}. \qquad (7.23)$$

7.7.2 DVFS with Varying Work Load

So far, we have considered static conditions to establish the efficiency of DVFS ap-proach. In real systems, however, the frequency and voltage are to be dynamically adjusted to match the changing demands for processing power. The implementation of the DVFS system will require the following hardware building blocks:

• Variable voltage processor $\mu(r)$: The need of a processor which can operate over a frequency range with a corre-sponding lower supply voltage range can be manufactured using the present-day process technology and several such processors are commercially available. Table 7.10 provides the relation among frequency, voltage, and power consumption for this processor.

• Variable voltage generator $V(r)$: The variable voltage power supply can be realized with the help of a direct current (DC)-to-DC converter, which receives a fixed voltage V_F and generates a variable voltage V_r based on the input from the workload monitoring system.

• Variable frequency generator $f(r)$: The variable frequency is generated with the help of a phase lock loop (PLL) sys-tem. The heart of the device is the high-performance PLL-core, consisting of a phase frequency

detector (PFD), programmable on-chip filter, and voltage-con-trolled oscillator (VCO). The PLL generates a high-speed clock which drives a fre-quency divider. The divider generates the variable frequency $f(r)$. The PLL and the divider together generate the independent frequencies related to the PLL operating frequency.

In addition to the above hardware building blocks, there is the need of a work-load monitor. The workload monitor can be performed by the operating system (OS). Usually, the OS has no prior knowledge of the workload to be generated by a bursty application. In general, the future workloads are nondeterministic. As a consequence, predicting the future workload from the current situation is very difficult and errors in prediction can seriously reduce the gains of DVFS, which has been observed in several simulation studies. Moreover, the rate at which the DVFS is done has a significant bearing on the performance and energy. So, it is essential to develop suitable strategy for workload prediction and the rate of DVFS, such that processor utilization and energy saving are maximized.

Table 7.10 Relationship between voltage, frequency, and power

Frequency (f) MHz	Voltage V_{dd}	Relative power
700	1.65	100
600	1.60	80.59
500	1.50	59.03
400	1.40	41.14
300	1.25	24.60
200	1.10	12.70

7.7.3 The Model

A generic block diagram of a variable voltage processing system is shown in Fig. 7.29. The tasks generated from various sources are represented by the task queue. Each of the sources produce events at an average rate of λ_k, $(k = 1,2,\ldots,n)$. The task scheduler of the OS manages all these tasks and decides which process should run on the processor. The average rate of arrival of tasks at the processor is $\lambda = \lambda_k$. The processor, in turn, provides time varying processing rate $\mu(r)$. The OS

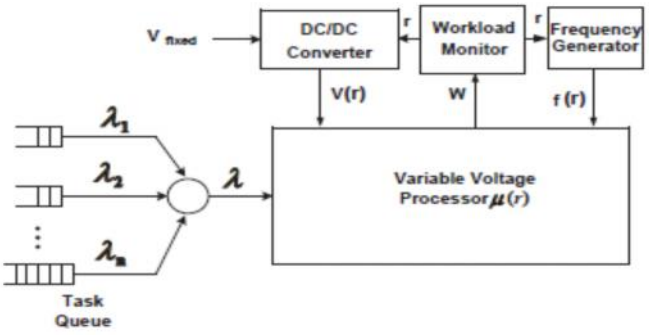


Fig. 7.29 Model for dynamic voltage scaling

7.7.4 Workload Prediction

It is assumed that the workload for the next observation interval can be predicted based on the workload statistics of the previous *N* intervals. The workload prediction for (n + 1) interval can be represented by

$$W_p[n+1] = \sum_{k=0}^{N-1} h_n[k] W(n-k), \qquad (7.24)$$

where $W[n]$ denotes the average normalized workload in the interval $(n-1)T \le t \le nT$ and $h_n[k]$ represents an N-tap, adaptable finite impulse response (FIR) filter, whose coefficients are updated in every observation interval based on the difference between the predicted and actual workloads. Uses of three possible filter types are examined below:

Moving Average Workload (MAW) In this case $h_n[k] = 1/N$, that is the filter pre-dicts the work load in the next time slot as the average of the previous *N* times slots. This simplistic scheme removes high-frequency workload changes and does not provide a satisfactory result for the time-varying workload statistics.

Exponential Weighted Averages (EWA) In this approach, instead of giving equal weightage to all the workload values of the previous slots, higher weightages are given to the most recent workload history. The idea is to give progressively decreas-ing importance to historical data. Mathematically, this can be achieved by providing the filter coefficients $h_n[k] = a^{-k}$, for all *n*, where a positive value of *a* is chosen such that $\sum h_n[k] = 1$.

Least Mean Square In this approach, the filter coefficients are modified based on the prediction error. One of the popular adaptive filter algorithms is the least-mean-square (LMS) algorithm, where $w[n]$ and $w_p[n]$ are the actual workload and predicted workload, respectively. Then the prediction error is given by $W_e[n] = W[n] - W_e[n]$. The filter coefficients are updated based on the following rule:

$h_n + 1[k] = h_n[k] + \mu W_e[n]. W[n-k]$, where μ is the step size.

7.7.5 Discrete Processing Rate

The operating points are determined analytically, first by finding out appropriate clock frequencies for different workloads. As PLLs along with frequency dividers are used to generate different frequencies, it is preferable to select clock periods which are multiples of the PLL frequency. This helps to generate clock frequencies with minimum latency.

Otherwise, it is necessary to change the PLL frequency, which requires a longer time to stabilize. Finally, the voltages required to support each of the frequencies are found out.

7.7.6 Latency Overhead

There is a latency overhead involved in processing rate update. This is due to the finite feedback bandwidth associated with the DC-to-DC converter. Changing the processor clock frequency also involves a latency overhead, during which the PLL circuit locks. To be on the safe side, it is recommended that the voltage and frequen-cy changes should not be done in parallel. In case of switching to higher processing rate, the voltage should be increased first, followed by the increase in frequency, and the following steps are to be followed:

- Set the new voltage.
- Allow the new voltage to settle down.
- Set the new frequency by changing the divider value, if possible. Otherwise, change the PLL clock frequency.

In case of switching to low processing rates, the frequency should be decreased first and then the voltage should be reduced to the appropriate level, and the following steps are to be followed:

- Set the new frequency by changing the divider value, if possible. Otherwise, change the PLL clock frequency.
- Set the new voltage. The CPU continues operating at the new frequency while voltage settles to the new value.

7.8 Adaptive Voltage Scaling

The voltage scaling techniques discussed so far are open loop in nature [7]. Voltage–frequency pairs are determined at design time keeping sufficient margin for guaranteed operation across the entire range of best- and worst-case PVT conditions. As the design needs to be very conservative for successful operation, the actual benefit obtained is lesser than actually possible. A better alternative that can overcome this limitation is the adaptive voltage scaling (AVS) where a close-loop feedback system is implemented between the voltage scaling power supply and delay-sensing performance monitor at execution time. The on-chip monitor not only checks the actual voltage developed but also detects whether the silicon is slow, typical, or fast and the effect of temperature on the surrounding silicon.

The implementation of the AVS system is shown in Fig. 7.32. The dynamic votage control (DVC) emulates the critical path characteristic of the system by using a delay synthesizer and controls the dynamic supply voltage. It consists of three major components: the pulse generator, the delay synthesizer, and the delay detector. By comparing the digitized delay value with the target value, the delay detector determines whether to increase, decrease, or keep the present supply voltage

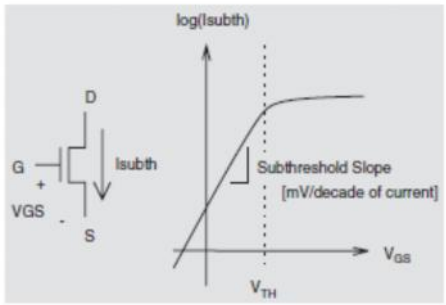


Fig. 7.33 Subthreshold region of operation