3

LOW POWER TECHNIQUES

3.1 General

Low power consumption has emerged as a major challenge in the design of integrated circuits. In this chapter, we discuss the basic principles for power consumption in standard CMOS circuits. Afterwards, a review of low-power techniques for CMOS circuits is given.

3.2 Power Dissipation Sources

In CMOS circuits, the main contributions to the power consumption are from short-circuit, leakage, and switching currents. In the following subsections, we introduce them separately.

3.2.1 Short-Circuit Power

In a static CMOS circuit, there are two complementary networks: p-network (pull-up) and n-network (pull-down). The logic functions for the two networks are complementary. Normally when the input and output state are stable, only one network is turned on and conducts the output either to power supply node or to ground node and the other network is turned off and blocks the current from flowing. Short-circuit current exists during the transitions as one network is turned on and the other network is still active. For example, the input signal to an inverter is switching from 0 to $V_{dd}$. It exists a short time interval where the input voltage is larger than $V_{in}$ but less than $V_{dd} - |V_{tp}|$. During this time interval, both PMOS-transistor (p-network) and NMOS-transistor (n-network) are turned on and the short-circuit current flows through both kinds of transistors from power supply line to the ground.
The exact analysis of the short-circuit current in a simple inverter [56] is complex; it can be studied by simulation using SPICE. It is observed that the short-circuit current is proportional to the slope of input signals, the output loads and the transistor sizes [57]. The short-circuit current consumes typically less than 10% of the total power in a “well-designed” circuit [57].

3.2.2 Leakage Power

There are two contributions to leakage currents: one from the currents that flow through the reverse biased diodes, the other from the currents that flow through transistors that are non-conducting.

![Figure 3.1 Leakage current types: (a) reverse biased diode current, (b) subthreshold leakage current.](image)

The leakage currents are proportional to the leakage area and exponential of the threshold voltage. The leakage currents depend on the technology and cannot be modified by the designers except in some logic styles.

The leakage current is in the order of pico-Ampere with current technology but it will increase as the threshold voltage is reduced. In some cases, like large RAMs, the leakage current is one of the main concerns. The leakage current is currently not a severe problem in most digital designs. However, the power consumed by leakage current can be as large as the power consumed by the switching current for 0.06 mm technology. The usage of multiple threshold voltages can reduce the leakage current in deep-submicron technology.
3.2.3 Switching Power

The switching currents are due to the charging and discharging of node capacitances. The node capacitances mainly include gate, overlapping, and interconnection capacitances.

The power consumed by switching current [58] can be expressed as

\[ P = a C_L f V_{dd}^2 / 2 \]  (3.1)

where \( a \) is the switching activity factor, \( C_L \) is the capacitance load, \( f \) is the clock frequency, and \( V_{dd} \) is the power supply voltage.

The equation shows that the switching power depends on a few quantities that are readily observable and measurable in CMOS circuits. It is applicable to almost every digital circuits and gives the guidance to the low power design.

The power consumed by switching current is the dominant part of the power consumption. Reducing the switching current is the focus of most low power design techniques.

3.3 Low Power Techniques

Low power techniques can be discussed at various levels of abstractions: system level, algorithm and architecture level, logic level, circuit level, and technology level. Fig. 3.2 shows some examples of techniques at the different levels.

- System: Partitioning, Power-down
- Algorithm: Parallelism, Pipelining
- Architecture: Voltage scaling
3.3.1 System Level

A system typically consists of both hardware and software components, which affect the power consumption.

The system design includes the hardware/software partitioning, hardware platform selection (application-specific or general purpose processors), resource sharing (scheduling) strategy, etc. The system design usually has the largest impact on the power consumption and hence the low power techniques applied at this level have the most potential for power reduction.

At the system level, it is hard to find the best solution for low power in the large design space and there is a shortage of accurate power analysis tools at this level. However, if, for example, the instruction-level power models for a given processor are available, software power optimization can be performed [59]. It is observed that faster code and frequently usage of cache are most likely to reduce the power consumption. The order of instructions also have an impact on the internal switching within processors and hence on the power consumption.

The power-down and clock gating are two of the most used low power techniques at system level. The non-active hardware units are shut down to save the power. The
clock driver, which often consumes 30-40% of the total power consumption, can be gated to reduce the switching activities as illustrated in Fig. 3.3.

![Figure 3.3 Clock gating.](image)

The power-down can be extended to the whole system. This is called sleep mode and widely used in low power processors. The Strong ARM SA-1100 processor has three power states and the average power varies for each state [60]. These power states can be utilized by the software through advanced configuration and power management interface (ACPI). In the recent year, the power management has gained a lot attention in operating system design. For example, the Microsoft desktop operating system supports advanced power management (APM).

![Figure 3.4 Power states for Strong ARM SA-1100 processor.](image)

The system is designed for the peak performance. However, the computation requirement is time varying. Adapting clocking frequency and/or dynamic voltage scaling to match the performance constraints is another low power technique. The lower requirement for performance at certain time interval can be used to reduce the power supply voltage. This requires either feedback mechanism (load monitoring and voltage control) or predetermined timing to activate the voltage down-scaling.

Another less explored domain for low power design is using asynchronous design techniques. The asynchronous designs have many attractive features, like non-
global clocking, automatic power down, no spurious transitions, and low peak current, etc. It is easy to reduce the power consumption further by combining the asynchronous design technique with other low power techniques, for instance, dynamic voltage scaling technique [44]. This is illustrated in Fig. 3.5.

![Asynchronous design with dynamic voltage scaling](image)

**Figure 3.5 Asynchronous design with dynamic voltage scaling.**

### 3.3.2 Algorithm Level

The algorithm selection has large impact on the power consumption. For example, using fast Fourier transform instead of direct computation of the DFT reduces the number of operations with a factor of 102.4 for a 1024-point Fourier transform and the power consumption is likely to be reduced with a similar factor.

The task of algorithm design is to select the most energy efficient algorithm that just satisfies the constraints. The cost of an algorithm includes the computation part and the communication/storage part. The complexity measurement for an algorithm includes the number of operations and the cost of communication/storage. Reduction of the number of operations, cost per operation, and long distance communications are key issues to algorithm selection.

One important technique for low power of the algorithmic level is algorithmic transformations [62] [63]. This technique exploits the complexity, concurrency,
regularity, and locality of an algorithm. Reducing the complexity of an algorithm reduces the number of operations and hence the power consumption. The possibility of increasing concurrency in an algorithm allows the use of other techniques, e.g., voltage scaling, to reduce the power consumption. The regularity and locality of an algorithm affects the controls and communications in the hardware.

The loop unrolling technique [64] [65] is a transformation that aims to enhance the speed. This technique can be used for reducing the power consumption. With loop unrolling, the critical path can be reduced and hence voltage scaling can be applied to reduce the power consumption. In Fig. 3.6, the unrolling reduces the critical path and gives a voltage reduction of 26% [65]. This reduces the power consumption with 20% even the capacitance load is increases with 50% [65]. Furthermore, this technique can be combining with other techniques at architectural level, for instance, pipeline and interleaving, to save more power.

![Figure 3.6](image.png)

Figure 3.6 (a) Original signal flow graph. (b) Unrolled signal flow graph.

In some cases, like wave digital filters, the faster algorithms, combined with voltage-scaling, can be chosen for energy-efficient applications [54].

**3.3.3 Architecture Level**

As the algorithm is selected, the architecture can be determined for the given algorithm.
As we can see from Eq. (3.1), an efficient way to reduce the dynamic power consumption is the voltage scaling. When supply voltage is reduced, the power consumption is reduced. However, this increases the gate delay. The delay of a min-size inverter (0.35 mm standard CMOS technology) increases as the supply voltage is reduced, which is shown in Fig. 3.7.

![Graph for Delay vs. supply voltage for an inverter.](image)

**Example 3.1 Parallel [66].**

The use of two parallel datapath is equivalent to interleaving of two computational tasks. A datapath to determine the largest number of $C$ and $(A + B)$ is shown in Fig. 3.8. It requires an adder and a comparator. The original clock frequency is 40 MHz [66].
In order to maintain the throughput while reducing the power supply voltage, we use a parallel architecture. The parallel architecture with twice the amount of resources is shown in Fig. 3.9. The clock frequency can be reduced to half, from 40 MHz to 20 MHz since two tasks are executed concurrently. This allows the supply voltage to be scaled down from 5 V to 2.9 V [66]. Since the extra routing is required to distribute computations to two parallel units, the capacitance load is increased by a factor of 2.15 [66]. Still, this gives a significant power saving [66]:

\[
P_{par} = C_{par}V_{par}^2f_{par} = (2.15C_{orig})(0.58V_{orig})^2\left(\frac{f_{orig}}{2}\right)
\]

\[
\approx 0.36P_{orig}
\]
Example 3.2 Pipelining [66].

Pipelining is another method for increasing the throughput. By adding a pipelining register after the adder in Fig. 3.8, the throughput can be increased from $1/(T_{add} + T_{comp})$ to $1/\max(T_{add}, T_{comp})$. If $T_{add}$ is equal to $T_{comp}$, this increases the throughput by a factor of 2. With this enhancement, the supply voltage can also be scaled down to 2.9 V (the gate delay doubles) [66]. The effective capacitance increases to a factor of 1.15 because of the insertions of latches [66]. The power consumption for pipelining [66] is

$$P_{pipe} = C_{pipe}V_{pipe}^2f_{pipe}$$

$$= (1.15C_{orig})(0.58V_{orig})^2f_{orig} = 0.39P_{orig}$$
One benefit of pipelining is the low area overhead in comparison with using parallel datapaths. The area overhead equals the area of the inserted latches. Another benefit is that the amount of glitches can be reduced.

Further power saving can be obtained by parallelism and/or pipelining. However, since the delay increases significantly as the voltage approaches the threshold voltage and the capacitance load for routing and/or pipeline registers increases, there exists an optimal power supply voltage. Reduction of supply voltage lower than the optimal voltage increases the power consumption.

Locality is also an important issue for architecture trade-off. The on-chip communication through long buses requires significant amount of power. To reduce such communications is important.

### 3.3.4 Logic Level

The power consumption depends on the switching activity factor, which in turn depends on the statistical characteristics of data. However, most low power techniques do not concentrate on this issue from the system level to the architecture level. The low power techniques at the logic level, however, focus mainly on the reduction of switching activity factor by using the signal correlation and, off course, the node capacitances.
As we know from the gated clocking, the clock input to nonactive functional block does not change by gating, and, hence, reduces the switching of clock network. Precomputation [67] uses the same concept to reduce the switching activity factor: a selective precomputing of the output of a circuit is done before the outputs are required, and this reduces the switching activity by gating those inputs to the circuit. This is illustrated in Fig. 3.11. The input data is partitioned into two parts, corresponding to registers $R_1$ and $R_2$. One part, $R_1$, is computed in precomputation block $g$ one clock cycle before the main computation $A$ is performed. The result from $g$ decides gating of $R_2$. The power can then be saved by reducing the switching activity factor in $A$.

An example of precomputation for low-power is the comparator. The comparator takes the MSB of the two numbers to register $R_1$ and the others to $R_2$. The comparison of MSB is performed in $g$. If two MSBs are not equal, the output from $g$ gated the remaining inputs. In this way, only a small portion of inputs to the comparator’s main block $A$ (subtractor) is changed. Therefore the switching activity is reduced.

![Figure 3.11 A precomputation structure for low power.](image)

Gate reorganization [68] [39] [70] is a technique to restructure the circuit. This can be decomposition a complex gate to simple gates, or composition simple gates to a complex gate, duplication of a gate, deleting/addition of wires. The decomposition of a complex gate and duplication of a gate help to separate the critical and
noncritical path and reduce the size of gates in the non-critical path, and, hence, the power consumption. In some cases, the decomposition of a complex gate increases the circuit speed and gives more space for power supply voltage scaling. The composition of simple gates can reduce the power consumption if the complex gate can reduce the charge/discharge of high-frequently switching node. The deleting of wires reduces the capacitance load and circuit size. The addition of wires helps to provide an intermediate circuit that may eventually lead to a better one.

Encoding defines the way data bits are represented on the circuits. The encoding is usually optimized for reduction of delay or area. In low power design, the encoding is optimized for reduction of switching activities since various encoding schemes have different switching properties.

In a counter design, counters with binary and Gray code have the same functionality. For $N$-bit counter with binary code, a full counting cycle requires $2(2^n - 1)$ transitions [58]. A full counting cycle for a Gray coded $N$-bit counter requires only $2^n$ transitions. For instance, the full counting cycle for a 2-bit binary coded counter is from 00, 01, 10, 11, and back to 00, which requires 6 transitions. The full counting cycle for 2-bit Gray coded counter is from 00, 01, 11, 10, and back to 00, which requires 4 transitions. The binary coded counter has twice transitions as the Gray coded counter when the $n$ is large. Using binary coded counter therefore requires more power consumption than using Gray coded counter under the same conditions.

As we can see from the previous example, the logic coding style has large impact on the number of transitions. Traditionally, the logic coding style is used for enhancement of speed performance. Careful choice of coding style is important to meet the speed requirement and minimize the power consumption. This can be applied to the finite state machine, where states can be coded with different schemes.
A bus is an on-chip communication channel that has large capacitance. As the on-chip transfer rate increases, the use of buses contributes with a significant portion of the total power. Bus encoding is a technique to exploit the property of transmitted signal to reduce the power consumption. For instance, adding an extra bit to select one of the inverse or the non-inverse bits at the receiver end can save power [71]. Low swing techniques can be applied for the bus also [72].

### 3.3.5 Circuit Level

At the circuit level, the potentials power saving are often less than that of higher abstract levels. However, this cannot be ignored. The power savings can be significant as the basic cells are frequently used. A few percents improvement for D flip-flop can significantly reduce the power consumption in deep pipelined systems. In CMOS circuits, the dynamic power consumption is caused by the transitions. Spurious transitions typically consume between 10% and 40% of the switching activity power in the typical combinational logic [41]. In some cases, like array multipliers, the amount of spurious transitions is large. To reduce the spurious transitions, the delays of signals from registers that converge at a gate should be roughly equal. This can be done by insertions of buffers and device sizing [73]. The insertions of buffer increase the total load capacitance but can still reduce the spurious transitions. This technique is called path balancing.

Many logic gates have inputs that are logically equivalent, i.e., the swapping of inputs does not modify the logic function of the gate. Example gates are NANDs, NORs, XORs, etc. However, from the power consumption point of view, the order of inputs does effect the power consumption. For instance, the A-input, which is near the output in a two-input NAND gate, consumes less power than the B-input closed to the ground with the same switching activity factor. Pin ordering is to assign more frequently switching to input pin that consumes less power. In this way, the power consumption will be reduced without cost. However, the statistics
of switching activity factors for different pins must be known in advanced and this limits the use of pin ordering [58].

Different logic styles have different electrical characteristics. The selection of logic style affects the speed and power consumption. In most cases, the standard CMOS logic is a good starting point for speed and power trade-off. In some cases, for instance, the XOR/NXOR implementation, other logic styles, like complementary pass-transistor logic (CPL) is efficient. CPL implements a full-adder with fewer transistors than the standard CMOS. The evaluation of full-adder is done only with NMOS transistors network. This gives a small layout as well.
Transistor sizing affects both delay and power consumption. Generally, a gate with smaller size has smaller capacitance and consumes less power. This is paid for by larger delay. To minimize the transistor sizes and meet the speed requirement is a trade-off. Typically, the transistor sizing uses static timing analysis to find out those gates (whose slack time is larger than 0) to be reduced. The transistor sizing is generally applicable for different technologies.

3.4 Low Power Guidelines

Several approaches to reduce the power consumption have been briefly discussed. Below we summarize some of the most commonly used low power techniques.

- Reduce the number of operations. The selection of algorithm and/or architecture has significant impact on the power consumption.
- Power supply voltage scaling. The voltage scaling is an efficient way to reduce the power consumption. Since the throughput is reduced as the voltage is reduced, this may need to be compensated for with parallel and/or pipelining techniques.
- I/Os between chips can consume large power due to the large capacitive loads. Reducing the number of chips is a promising approach to reduce the power consumption.
- Power management. In many systems, the most power consuming parts are often idle. For example, in a lap-top computer, the portion of display and hard disk could consume more than 50% of the total power consumption. Using power management strategies to shut down these components when they are idle for a long time can achieve good power saving.
- Reducing the effective capacitance. The effective capacitance can be reduced by several approaches, for example, compact layout and efficient logic style.
- Reduce the number of transitions. To minimize the number of transitions, especially the glitches, is important.
3.5 Summary

In this chapter we discussed some low power techniques that are applicable at different abstraction levels.