x9

EXTRA TOPICS RELATED TO

MOS Digital Integrated Circuits

This supplement contains material removed from previous editions of the textbook. These topics continue to be relevant and for this reason will be of great value to many instructors and students.

The topics presented here relate to advanced topics in MOS digital integrated circuits, and can be selected to augment the materials in Chapter 17 (Sections x9.1 to x9.5) and Chapter 18 (Sections x9.6 to x9.8).

x9.1 Velocity Saturation

The short channels of MOSFETs fabricated in deep-submicron processes give rise to physical phenomena not present in long-channel devices, and thus to changes in the MOSFET i–v characteristics. The most important of these **short-channel** effects is **velocity saturation**. Here we refer to the drift velocity of electrons in the channel of an NMOS transistor (holes in PMOS) under the influence of the longitudinal electric field established by v_{DS}. In our derivation of the MOSFET i–v characteristics in Chapter 5 of the eighth edition, we assumed that the velocity v_n of the electrons in an n-channel device is given by

$$v_n = \mu_n E \tag{x9.1}$$

where E is the electric field given by

$$E = \frac{v_{DS}}{L} \tag{x9.2}$$

The relationship in Eq. (x9.1) applies as long as E is below a critical value E_{cr} that falls in the range 1 V/μm to 5 V/μm. For $E > E_{cr}$, the drift velocity saturates at a value v_{sat} of approximately 10^7 cm/s. Figure x9.1 shows a sketch of v_n versus E. Although the change from a linear to a constant v is gradual, we shall assume for simplicity that v saturates abruptly at $E = E_{cr}$.

The electric field E in a short-channel MOSFET can easily exceed E_{cr} even though V_{DD} is low. If we denote the value of v_{DS} at which velocity saturation occurs by $V_{DS\,\text{sat}}$, then from Eq. (x9.2),

$$E_{cr} = \frac{V_{DS\,\text{sat}}}{L} \tag{x9.3}$$

which when substituted in Eq. (x9.1) provides

$$v_{\text{sat}} = \mu_n \left(\frac{V_{DS\,\text{sat}}}{L}\right) \tag{x9.4}$$

or alternatively,

$$V_{DS\,\text{sat}} = \left(\frac{L}{\mu_n}\right) v_{\text{sat}} \tag{x9.5}$$

Thus, $V_{DS\,\text{sat}}$ is a device parameter.

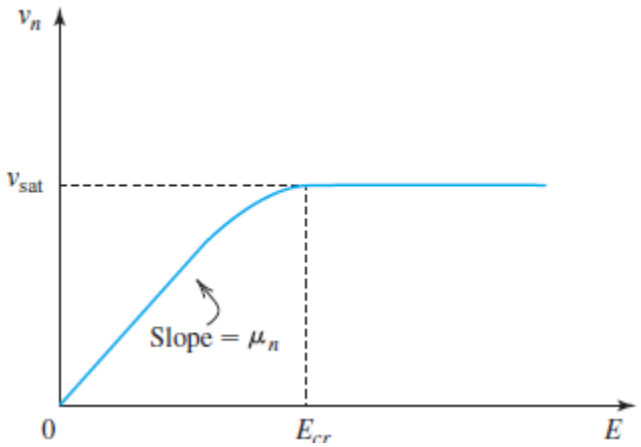

Figure x9.1 The velocity of electrons in the channel of an NMOS transistor reaches a constant value $v_{\text{sat}} \simeq 10^7$ cm/s when the electric field E reaches a critical value E_{cr}. A similar situation occurs for p-channel devices.

EXERCISE

x9.1 Find $V_{DS\ sat}$ for an NMOS transistor fabricated in a 0.25-μm CMOS process with μ_n = 400 cm2/V · s. Let L = 0.25 μm and assume v_{sat} = 107 cm/s.

Ans. 0.63 V

x9.1.1 The i_D–v_{DS} Characteristics

The i_D–v_{DS} equations of the MOSFET can be modified to include velocity saturation as follows. Consider a long-channel NMOS transistor operating in the triode region with v_{GS} set to a constant value V_{GS}. The drain current will be

$$i_D = \mu_n C_{ox}\left(\frac{W}{L}\right) v_{DS}\left[(V_{GS} - V_t) - \frac{1}{2}v_{DS}\right] \tag{x9.6}$$

where we have for the time being neglected channel-length modulation. We know from our study in Chapter 5 of the eighth edition that i_D will saturate at

$$v_{DS} = V_{OV} = V_{GS} - V_t \tag{x9.7}$$

and the saturation current will be

$$i_D = \frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)(V_{GS} - V_t)^2 \tag{x9.8}$$

This will also be the case in a short-channel device as long as the value of v_{DS} in Eq. (x9.7) is lower than $V_{DS\ sat}$. That is, as long as

$$V_{OV} < V_{DS\,sat}$$

the current i_D will be given by Eqs. (x9.6) and (x9.8). If, on the other hand,

$$V_{OV} > V_{DS\,sat}$$

then velocity saturation kicks in at $v_{DS} = V_{DS\ sat}$ and i_D saturates at a value $I_{D\ sat}$, as shown in Fig. x9.2. The value of $I_{D\ sat}$ can be obtained by substituting $v_{DS} = V_{DS\ sat}$ in Eq. (x9.6),

$$I_{D\ sat} = \mu_n C_{ox}\left(\frac{W}{L}\right) V_{DS\ sat}\left(V_{GS} - V_t - \frac{1}{2}V_{DS\ sat}\right) \tag{x9.9}$$

This expression can be simplified by utilizing Eq. (x9.5) to obtain

$$I_{D\ sat} = WC_{ox}v_{sat}\left(V_{GS} - V_t - \frac{1}{2}V_{DS\ sat}\right) \tag{x9.10}$$

Replacing V_{GS} in Eq. (x9.9) with v_{GS}, and incorporating the channel-length modulation factor $(1 + \lambda v_{DS})$, we obtain a general expression for the drain current of an NMOS transistor operating in velocity saturation,

$$i_D = \mu_n C_{ox} \left(\frac{W}{L}\right) V_{DS\,\text{sat}} \left(v_{GS} - V_t - \frac{1}{2} V_{DS\,\text{sat}}\right) (1 + \lambda v_{DS}) \tag{x9.11}$$

which applies for

$$v_{GS} - V_t \geq V_{DS\,\text{sat}} \qquad \text{and} \qquad v_{DS} \geq V_{DS\,\text{sat}} \tag{x9.12}$$

Figure x9.3 shows a set of i_D–v_{DS} characteristic curves and clearly delineates the three regions of operation: triode, saturation, and velocity saturation.

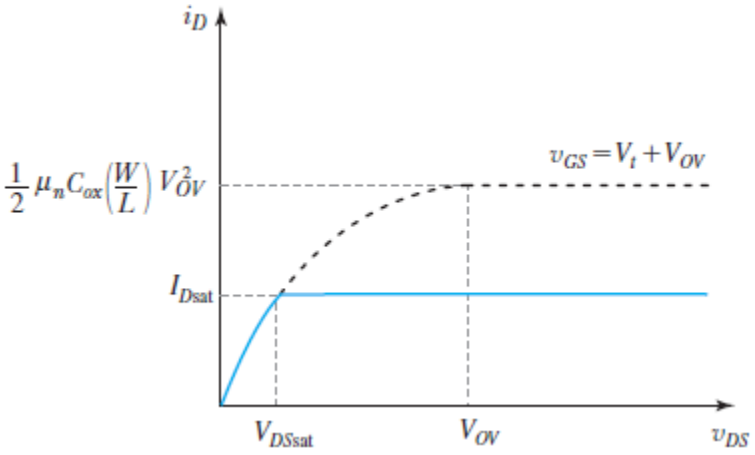

Figure x9.2 Velocity saturation causes the i_D–v_{DS} characteristic to saturate at V_{DSsat}. This early saturation results in a current I_{Dsat} that is lower than the value for a long-channel device.

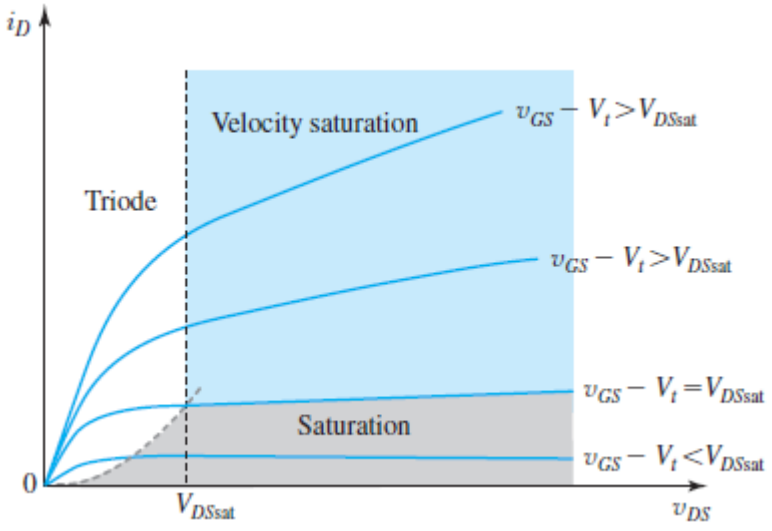

Figure x9.3 The i_D–v_{DS} characteristics of a short-channel MOSFET. Note the three different regions of operation: triode, saturation, and velocity saturation.

Equation (x9.11) indicates that in the velocity-saturation region, i_D is linearly related to v_{GS}. This is a major change from the quadratic relationship that characterizes operation in the saturation region. Figure x9.4 makes this point clearer by presenting a graph for i_D versus v_{GS} of a short-channel device operating at $v_{DS} > V_{DS\,sat}$. Observe that for $0 < v_{GS}–V_t \leq V_{DS\,sat}$, the MOSFET operates in the saturation region and i_D is related to v_{GS} by the familiar quadratic equation (Eq. x9.8). For $v_{GS}–V_t \geq V_{DS\,sat}$, the transistor enters the velocity-saturation region and i_D varies linearly with v_{GS} (Eq. x9.11).

Short-channel PMOS transistors undergo velocity saturation at the same value of v_{sat} (approximately 107cm/s), but the effects on the device characteristics are less pronounced than in the NMOS case. This is due to the lower values of μ_p and the correspondingly higher values of E_{cr} and $V_{DS\,sat}$.

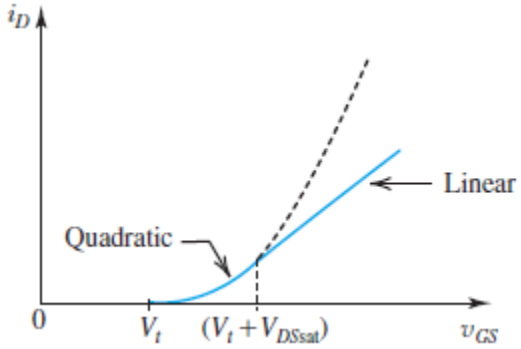

Figure x9.4 The i_D–v_{GS} characteristic of a short-channel NMOS transistor operating at $v_{DS} > V_{DS\,sat}$. Observe the quadratic and the linear portions of the characteristic. Also note that in the absence of velocity saturation, the quadratic curve would continue as shown with the broken line.

Example x9.1

Consider MOS transistors fabricated in a 0.25-μm CMOS process for which $V_{DD} = 2.5$ V, $V_{tn} = -V_{tp} = 0.5$ V, $\mu_n C_{ox} = 115\ \mu A/V^2$, $\mu_p C_{ox} = 30\ \mu A/V^2$, $\lambda_n = 0.06\ V^{-1}$, and $|\lambda_p| = 0.1\ V^{-1}$. Let $L = 0.25\ \mu m$ and $(W/L)_n = (W/L)_p = 1.5$. Measurements indicate that for the NMOS transistor, $V_{DS\,sat} = 0.63$ V, and for the PMOS device, $|V_{DS\,sat}| = 1$ V. Calculate the drain current obtained in each of the NMOS and PMOS transistors for $|V_{GS}| = |V_{DS}| = V_{DD}$. Compare with the values that would have been obtained in the absence of velocity saturation. Also give the range of v_{DS} for which i_D is saturated, with and without velocity saturation.

Solution

For the NMOS transistor, $V_{GS} = 2.5$ V results in $V_{GS} - V_{tn} = 2.5 - 0.5 = 2$ V, which is greater than $V_{Ds\,sat}$. Also, $V_{DS} = 2.5$ V is greater than $V_{DS\,sat}$; thus both conditions in Eq. (x9.12) are satisfied, and the NMOS transistor will be operating in the velocity-saturation region, and thus i_D is given by Eq. (x9.11):

$$i_D = 115 \times 10^{-6} \times 1.5 \times 0.63 \times \left(2.5 - 0.5 - \frac{1}{2} \times 0.63\right) \times (1 + 0.06 \times 2.5)$$
$$= 210.6\ \mu\text{A}$$

If velocity saturation were absent, the current would be

$$i_D = \frac{1}{2}(\mu_n C_{ox}) \left(\frac{W}{L}\right)_n (v_{GS} - V_{tn})^2 (1 + \lambda v_{DS})$$
$$= \frac{1}{2} \times 115 \times 10^{-6} \times 1.5 \times (2.5 - 0.5)^2 \times (1 + 0.06 \times 2.5)$$
$$= 396.8\ \mu\text{A}$$

Thus, velocity saturation reduces the current level by nearly 50%! The saturation current, however, is obtained over a larger range of v_{DS}; specifically, for v_{DS} = 0.63 V to 2.5 V. (Of course, the current does not remain constant over this range because of channel-length modulation.) In the absence of velocity saturation, the current saturates at $V_{OV} = V_{GS} - V_t$ = 2 V, and thus the saturation current is obtained over the range v_{DS} = 2 V to 2.5 V.

For the PMOS transistor, since $|V_{GS}| - |V_t|$ = 2 V and $|V_{DS}|$ = 2.5 V are both larger than $|V_{DS\ \text{sat}}|$ = 1 V, the device will be operating in velocity saturation, and i_D can be obtained by adapting Eq. (x9.11) as follows:

$$i_D = (\mu_p C_{ox}) \left(\frac{W}{L}\right)_p |V_{DS\ \text{sat}}| \left(|V_{GS}| - |V_{tp}| - \frac{1}{2}|V_{DS\ \text{sat}}|\right) (1 + |\lambda_p||V_{DS}|)$$
$$= 30 \times 10^{-6} \times 1.5 \times 1 \times \left(2.5 - 0.5 - \frac{1}{2} \times 1\right) (1 + 0.1 \times 2.5)$$
$$= 84.4\ \mu\text{A}$$

Without velocity saturation, we have

$$i_D = \frac{1}{2}(\mu_p C_{ox}) \left(\frac{W}{L}\right)_p (|V_{GS}| - |V_{tp}|)^2 (1 + |\lambda_p||V_{DS}|)$$
$$= \frac{1}{2} \times 30 \times 10^{-6} \times 1.5 \times (2.5 - 0.5)^2 (1 + 0.1 \times 2.5)$$
$$= 112.5\ \mu\text{A}$$

Thus velocity saturation reduces the current by 25% (which is less than in the cse of the NMOS transistor), and the saturated current is obtained over the range $|V_{DS}|$ = 1 V to 2.5 V. In the absence of velocity saturation, the saturated i_D would have been obtained for $|V_{DS}|$ = 2 V to 2.5 V.

EXERCISE

x9.2 Repeat the problem in Example x9.1 for transistors fabricated in a 0.13-μm CMOS process for which V_{DD} = 1.2 V, $V_{tn} = -V_{tp}$ = 0.4 V, $\mu_n C_{ox}$ = 110 μA/V^2, $\lambda_n = |\lambda_p|$ = 0.1 V^{-1}. Let L = 0.13 μm, $(W/L)_p$ = 1.5, $V_{DS\,sat}$ (NMOS) = 0.34 V, and $V_{DS\,sat}$ (PMOS) = 0.6 V.

Ans. NMOS: I_D = 154.7 μA, compared to 231.2 μA without velocity saturation; saturation is obtained over the range v_{DS} = 0.34 V to 1.2 V, compared to v_{DS} = 0.8 V to 1.2 V in the absence of velocity saturation. PMOS: ID = 55.4 μA compared to 59.1 μA, and $|v_{DS}|$ = 0.6 V to 1.2 V compared to 0.8 V to 1.2 V.

x9.1.2 Effect on the Inverter Characteristics

The VTC of the CMOS inverter can be derived using the modified i_D–v_{DS} characteristics of the MOSFETs. The results, however, indicate relatively small changes from the VTC derived in Chapter 16 of the eighth edition using the long-channel equations (see Rabaey et al., 2003, and Hodges et al., 2004), and we shall not pursue this subject here. The dynamic characteristics of the inverter, however, are significantly impacted by velocity saturation. This is because the current available to charge and discharge the equivalent load capacitance C is substantially reduced.

x9.1.3 A Remark on the MOSFET Model

The model derived above for short-channel MOSFETs is an approximate one, intended to enable the circuit designer to perform hand analysis to gain insight into circuit operation. Also, the model parameter values are usually obtained from measured data by means of a numerical curve-fitting process. As a result, the model applies only over a restricted range of terminal voltages.

Modeling short-channel MOSFETs is an advanced topic that is beyond the scope of this book. Suffice it to say that sophisticated models have been developed and are utilized by circuit simulation programs such as SPICE (see Appendix B). Circuit simulation is an essential step in the design of integrated circuits. However, it is not a substitute for initial hand analysis and design.

x9.2 Subthreshold Conduction

x9.1.4 Subthreshold Conduction

In our study of the NMOS transistor in Chapter 5 of the eighth edition, we assumed that current conduction between drain and source occurs only when v_{GS} exceeds V_t. That is, we assumed that for $v_{GS} < V_t$ no current flows between drain and source. This, however, turns out not to be the case, especially for deep-submicron devices. Specifically, for $v_{GS} < V_t$ a small current i_D flows. To be able to see this **subthreshold conduction**, we have redrawn the i_D–v_{GS} graph, utilizing a logarithmic scale for i_D, as shown in Fig. x9.5. Observe that at low values of v_{GS}, the relationship between log i_D and v_{GS} is linear, indicating that i_D varies exponentially with v_{GS},

$$i_D = I_S e^{v_{GS}/nV_T} \qquad \text{(x9.13)}$$

where I_S is a constant, $V_T = kT/q$ is the thermal voltage $\simeq$ 25 mV at room temperature, and n is a constant whose value falls in the range 1–2, depending on the material and structure of the device. Subthreshold conduction has been put to good use in the design of very-low-power circuits such as those needed for electronic watches. Generally speaking, however, subthreshold conduction is a problem in digital IC design, for two reasons:

1. The nonzero current that flows for $v_{GS} = 0$ (see Fig. x9.5) causes the CMOS inverter to dissipate static power. To keep this **off current** as low as possible, V_t of the MOSFET is kept relatively high. This indeed is the reason why V_t has not been scaled by the same factor as that used for the channel length. Although the off current is low (10 pA to 100 pA) and the power dissipation per inverter is small, the problem becomes serious in chips with a billion transistors!
2. The nonzero current of a normally off transistor can cause the discharge of capacitors in dynamic MOS circuits. Dynamic logic and memory circuits rely on charge storage on capacitors for their proper operation. Thus, subthreshold conduction can disrupt the operation of such circuits.

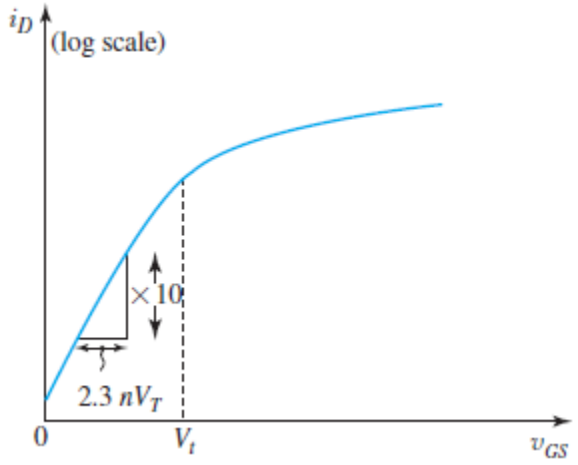

Figure x9.5 The i_D–v_{GS} characteristic of a short-channel MOSFET. To show the details of subthreshold conduction a logarithmic scale is needed for i_D.

EXERCISE

x9.3 (a) Refer to Fig. x9.5 and Eq. (x9.13). Show that the inverse of the slope of the straight line representing subthreshold conduction is given by $2.3nV_T$ V per decade of current change.

(b) If measurements indicate $n = 1.22$ and $i_D = 100$ nA at $v_{GS} = 0.21$ V, find i_D at $v_{GS} = 0$.

(c) For a chip having 500 million transistors, find the current drawn from the 1.2-V supply V_{DD} as a result of subthreshold conduction. Hence estimate the resulting power dissipation.

Ans. (b) 0.1 nA; (c) 50 mA, 60 mW

x9.3 Digital IC Technologies, Logic-Circuit Families, and Design Methodologies

In our study of digital circuits in Chapters 16, 17, and 18 of the eighth edition, we mainly concentrate on CMOS. This is reasonable in view of its dominance. Nevertheless, we will now take a broader view and survey other available digital circuit technologies. Not only will this help place CMOS in its proper context, it will also motivate the study of a number of other useful logic-circuit types. As well, we will briefly consider the methods digital IC designers employ to produce complex chips containing billions of transistors.

x9.3.1 Digital IC Technologies and Logic-Circuit Families

The chart in Figure x9.6 shows the major IC technologies and logic-circuit families that are currently in use. The concept of a logic-circuit family warrants a few words of explanation. The basic element of a logic-circuit family is the inverter. A family would include a variety of logic-circuit types made with the same technology, having a similar circuit structure and exhibiting the same basic features. Each logic-circuit family offers a unique set of advantages and disadvantages. In the conventional style of designing systems, one selects an appropriate logic family (e.g., TTL, CMOS, or ECL) and attempts to implement as much of the system as possible using circuit modules (packages) that belong to this family. This makes interconnection of the various packages relatively straightforward. If, on the other hand, packages from more than one family are used, one has to design suitable *interface circuits*. The selection of a logic family is based on such considerations as logic flexibility, speed of operation, availability of complex functions, noise immunity, operating-temperature range, power dissipation, and cost. Here we make some brief remarks on each of the four technologies listed in Fig. x9.6.

CMOS Although shown as one of four possible technologies, this is not an indication of digital IC market share: CMOS technology is, by a very large margin, the most dominant of all the IC technologies available for digital circuit design. Although early microprocessors were made using NMOS logic, CMOS has completely replaced NMOS. There are a number of reasons for this development, the most important of which is the much lower power dissipation of CMOS circuits. CMOS has also replaced bipolar as the technology of choice in digital system design and has made possible levels of integration (or circuit-packing densities) and a range of applications, neither of which would have been possible with bipolar technology. Furthermore, CMOS continues to advance, whereas there appear to be few innovations at the present time in bipolar digital circuits.

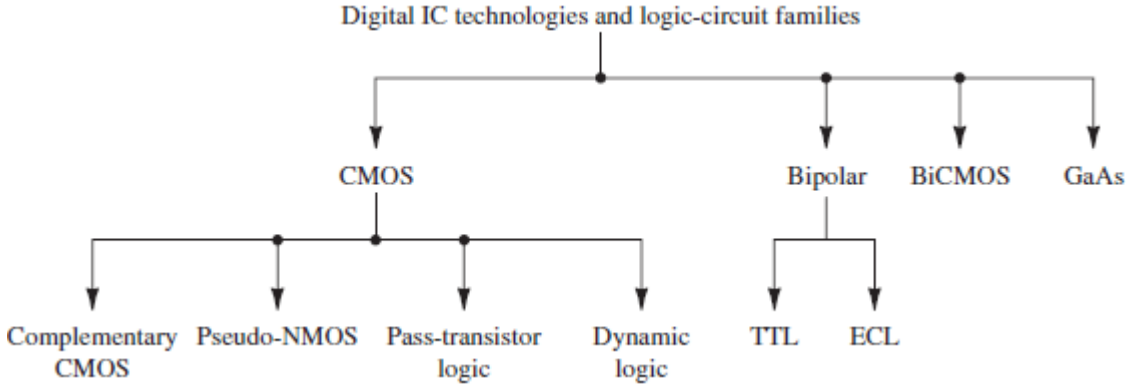

Figure x9.6 Digital IC technologies and logic-circuit families.

Some of the reasons for CMOS displacing bipolar technology in digital applications are as follows.

1. CMOS logic circuits dissipate much less power than bipolar logic circuits, and so we can pack more CMOS circuits on a chip than we can with bipolar circuits.
2. The high input impedance of the MOS transistor allows the designer to use charge storage as a means for the temporary storage of information in both logic and memory circuits. This technique cannot be used in bipolar circuits.
3. The feature size (i.e., minimum channel length) of the MOS transistor has decreased dramatically over the years, with some recently reported designs utilizing channel lengths as short as 7 nm and below. This permits very tight circuit packing and, correspondingly, very high levels of integration. A microprocessor chip reported in 2014 had 4.31 billion transistors.

Of the various forms of CMOS, complementary CMOS circuits, studied in Chapter 16 of the eighth edition, are the most widely used. They are available both as **small-scale-integrated (SSI)** circuit packages (containing 1–10 logic gates) and **medium-scale-integrated (MSI)** circuit packages (10–100 gates per chip) for assembling digital systems on printed-circuit boards. More significantly, complementary CMOS is used in **very-large-scale-integrated (VLSI)** logic (with millions of gates per chip) and memory-circuit design. In some applications, complementary CMOS is supplemented by one (or both) of two other MOS logic-circuit forms. These are pseudo-NMOS, so-named because of the similarity of its structure to NMOS logic, and pass-transistor logic, both of which will be studied in Section x9.4 of the bonus materials.

A fourth type of CMOS logic circuit utilizes dynamic techniques to obtain faster circuit operation, while keeping the power dissipation very low. Dynamic CMOS logic, which we shall study in Section x9.5 of the bonus materials, represents an area of growing importance. Lastly, CMOS technology is used in the design of memory chips, as detailed in Chapter 18 of the eighth edition.

Bipolar Two logic-circuit families based on the bipolar junction transistor are in some use at present: TTL and ECL. Transistor–transistor logic (TTL or T_2L) was for many years the most widely used logic-circuit family. Its decline was precipitated by the advent of the VLSI era. TTL manufacturers, however, fought back with the introduction of low-power and high-speed versions. In these relatively newer versions, the higher speeds of operation are made possible by preventing the BJT from saturating and thus avoiding the slow turnoff process of a saturated bipolar transistor. These nonsaturating versions of TTL utilize the Schottky diode and are called Schottky TTL or variations of this name. Despite all these efforts, TTL is no longer a significant logic-circuit family and will not be studied in this book. However, the interested reader can find significant amounts of material in Sections x8.1, x8.2, and x8.3 of the bonus materials.

The other bipolar logic-circuit family in present use is emitter-coupled logic (ECL). It is based on a current-switch implementation of the inverter. The basic element of ECL is the differential BJT pair. Because ECL is basically a current-steering logic, and, correspondingly, also called **current-mode logic** (CML), in which saturation is avoided, very high speeds of operation are possible. Indeed, of all the commercially available logic-circuit families, ECL is the fastest. ECL is also used in VLSI circuit design when very high operating speeds are required and the designer is willing to accept higher power dissipation and increased silicon area. As such, ECL is considered an important specialty technology and is discussed in Section 8.4 of the bonus materials.

BiCMOS BiCMOS combines the high operating speeds possible with BJTs (because of their inherently higher transconductance) with the low power dissipation and other excellent characteristics of CMOS. Like CMOS, BiCMOS allows for the implementation of both analog and digital circuits on the same chip. At present, BiCMOS is used to great advantage in special applications, where its high performance as a high-speed capacitive-current driver justifies themore complex process technology it requires. A very brief discussion of BiCMOS is provided in Section x8.5 of the bonus materials.

Gallium Arsenide (GaAs) The high carrier mobility in GaAs results in very high speeds of operation. This has been demonstrated in a number of digital IC chips utilizing GaAs technology. It should be pointed out, however, that GaAs remains an "emerging technology," one that appears to have great potential but has not yet achieved such potential commercially. As such, it will not be studied in this book. Nevertheless, considerable material on GaAs devices and circuits, including digital circuits, can be found in the bonus materials.

THE INVISIBLE COMPUTER

One may think that computer integrated circuits appear only in desktops, laptops, and mobile phones, but that is not the case! Virtually invisible to the casual observer is a vast number of computer chips called **microcontrollers**, which include a relatively high-speed processor (often of 8 bits, but increasingly 16 or 32), along with flash memory and flexible input/output circuitry. The input/output circuitry often includes A/D conversion. Microcontrollers operate within almost every modern appliance and computer peripheral: late-model automobiles include large numbers of networked microcontrollers for engine control, safety systems, stability, braking, and diagnostics. For example, in the 2012 Toyota Lexus there are about 100 such controllers.

x9.3.2 Styles for Digital System Design

The conventional approach to designing digital systems consists of assembling the system using standard IC packages of various levels of complexity (and hence integration). Many systems have been built this way using, for example, TTL, SSI, and MSI packages. The advent of VLSI, in addition to providing the system designer with more powerful off-the-shelf components such as microprocessors and memory chips, has made possible alternative design styles. One such alternative is to opt for implementing part or all of the system using one or more *custom VLSI* chips. However, custom IC design is usually economically justified only when the production volume is large (greater than about 100,000 parts).

An intermediate approach, known as *semicustom design*, utilizes *gate-array* chips. These are integrated circuits containing 100,000 or more unconnected logic gates. Their interconnection can be achieved by a final metallization step (performed at the IC fabrication facility) according to a pattern specified by the user to implement the user's particular functional need.Amore recently available type of gate array, known as a **field-programmable gate array (FPGA)**, can, as its name indicates, be programmed directly by the user. FPGAs provide a very convenient means for the digital system designer to implement complex logic functions in VLSI form without having to incur either the cost or the "turnaround time" inherent in custom and, to a lesser extent, in semicustom IC design.

x9.3.3 Design Abstraction and Computer Aids

The design of very complex digital systems, whether on a single IC chip or using off-the-shelf components, is made possible by the use of different levels of design abstraction, and the use of a variety of computer aids. To appreciate the concept of design abstraction, consider the process of designing a digital system using off-the-shelf packages of logic gates. The designer consults data sheets (in data books or on websites) to determine the input and output characteristics of the gates, their fan-in and fan-out limitations, and so on. In connecting the gates, the designer needs to adhere to a set of rules specified by the manufacturer in the data sheets. The designer does not need to consider, in a direct way, the circuit inside the gate package. In effect, the circuit has been abstracted in the form of a functional block that can be used as a component. This greatly simplifies system design. The digital IC designer follows a similar process. Circuit blocks are designed, characterized, and stored in a library as **standard cells**. These cells can then be used by the IC designer to assemble a larger subsystem (e.g., an adder or a multiplier), which in turn is characterized and stored as a functional block to be used in the design of an even larger system (e.g., an entire processor).

At every level of design abstraction, the need arises for simulation and other computer programs that help make the design process as automated as possible. Whereas SPICE is employed in circuit simulation, other software tools are utilized at other levels and in other phases of the design process. Although digital system design and design automation are outside the scope of this book, it is important that the reader appreciate the role of design abstraction and computer aids in digital design. They are what make it humanly possible to design a billion-transistor digital IC. Unfortunately, analog IC design does not lend itself to the same level of abstraction and automation. Each analog IC to a large extent has to be "handcrafted." As a result, the complexity and density of analog ICs remain much below what is possible in a digital IC.

Whatever approach or style is adopted in digital design, some familiarity with the various digital circuit technologies and design techniques is essential.

x9.4 Pseudo-NMOS Logic Circuits

x9.4.1 The Pseudo-NMOS Inverter

Figure x9.7 shows a modified form of the CMOS inverter. Here, only Q_N is driven by the input voltage while the gate of Q_P is grounded, and Q_P acts as an active load for Q_N. Even before we examine the operation of this circuit in detail, an advantage over standard CMOS is obvious: Each input needs to be connected to the gate of only one transistor or, alternatively, only one additional transistor (an NMOS) will be needed for each additional gate input. Thus the area and delay penalties arising from increased fan-in in a standard CMOS will be reduced. This is indeed the motivation for exploring this modified inverter circuit.

The inverter circuit of Fig. x9.7(a) resembles other forms of NMOS logic that consist of a driver transistor (Q_N) and a load transistor (in this case, Q_P); hence the name pseudo-NMOS. For comparison, we will briefly mention two older forms of NMOS logic. The earliest, popular in the mid-1970s, used an enhancement MOSFET for the load element, in a topology whose basic inverter is shown in Fig. x9.7(b). It can be shown that its disadvantages include a relatively small logic swing, small noise margins, and high static power dissipation. For these reasons, this logic-circuit technology is virtually obsolete. It was replaced in the late 1970s and early 1980s with depletion-load NMOS circuits, in which a depletion NMOS transistor with its gate connected to its source is used as the load element. The topology of the basic depletion-load inverter is shown in Fig. x9.7(c).

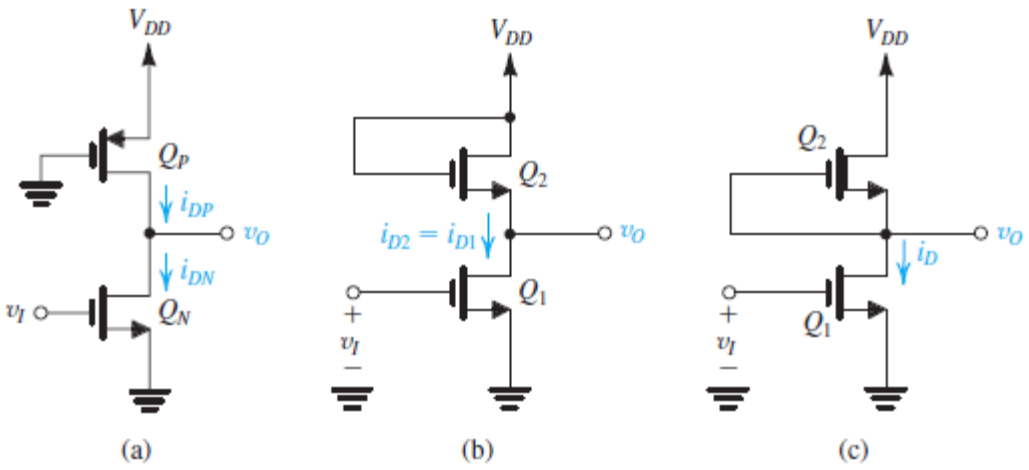

Figure x9.7 **(a)** The pseudo-NMOS logic inverter. **(b)** The enhancement-load (or saturated-load) NMOS inverter. **(c)** The depletion-load NMOS inverter.

It was initially expected that the depletion NMOS with $V_{GS} = 0$ would operate as a constant-current source and would thus provide an excellent load element.[3] However, it was quickly realized that the body effect in the depletion transistor causes its operation to deviate considerably from that of a constant-current source. Nevertheless, depletion-load NMOS circuits feature significant improvements over their enhancement-load counterparts, enough to justify the extra processing step required to fabricate the depletion devices (namely, ion-implanting the channel). Although depletion-load NMOS has been virtually replaced by CMOS, one can still see some depletion-load circuits in specialized applications.

The pseudo-NMOS inverter that we are about to study is similar to depletion-load NMOS, but with rather improved characteristics. It also has the advantage of being directly compatible with standard CMOS circuits.

x9.4.2 Static Characteristics

The static characteristics of the pseudo-NMOS inverter can be derived in a manner similar to that used for standard CMOS. Toward that end, we note that the drain currents of Q_N and Q_P are given by

$$i_{DN} = \frac{1}{2}k_n(v_I - V_t)^2, \quad \text{for } v_O \geq v_I - V_t \quad \text{(saturation)} \tag{x9.14}$$

$$i_{DN} = k_n\left[(v_I - V_t)v_O - \frac{1}{2}v_O^2\right] \quad \text{for } v_O \leq v_I - V_t \quad \text{(triode)} \tag{x9.15}$$

$$i_{DP} = \frac{1}{2}k_p(V_{DD} - V_t)^2, \quad \text{for } v_O \leq V_t \quad \text{(saturation)} \tag{x9.16}$$

$$i_{DP} = k_p\left[(V_{DD} - V_t)(V_{DD} - v_O) - \frac{1}{2}(V_{DD} - v_O)^2\right], \quad \text{for } v_O \geq V_t \text{ (triode)} \qquad \text{(x9.17)}$$

where we have assumed that $V_{tn} = -V_{tp} = V_t$, and have used $k_n = k'_n(W/L)_n$ and $k_p = k'_p(W/L)_p$ to simplify matters.

To obtain the voltage-transfer characteristic of the inverter, we superimpose the load curve represented by Eqs. (x9.16) and (x9.17) on the i_D–v_{DS} characteristics of Q_N, which can be relabeled as i_{DN}–v_O and drawn for various values of $v_{GS} = v_I$. Such a graphical construction is shown in Fig. x9.8, where, to keep the diagram simple, we show the Q_N curves for only the two extreme values of v_I, namely, 0 and V_{DD}. Two observations follow:

1. The load curve represents a much lower saturation current (Eq. x9.16) than is represented by the corresponding curve for Q_N, namely, that for $v_I = V_{DD}$. This is a result of the fact that the pseudo-NMOS inverter is usually designed so that k_n is greater than k_p by a factor of 4 to 10. As we will show shortly, this inverter is of the so-called ratioed type, and the ratio $r \equiv k_n/k_p$ determines all the breakpoints of the VTC, that is, V_{OL}, V_{IL}, V_{IH}, and so on, and thus determines the noise margins. Selection of a relatively high value for r reduces V_{OL} and widens the noise margins.
2. Although one tends to think of Q_P as acting as a constant-current source, it actually operates in saturation for only a small range of v_O, namely, $v_O \leq V_t$. For the remainder of the v_O range, Q_P operates in the triode region.

Consider first the two extreme cases of v_I : When $v_I = 0$, Q_N is cut off and Q_P is operating in the triode region, though with zero current and zero drain–source voltage. Thus the operating point is that labeled A in Fig. x9.8, where $v_O = V_{OH} = V_{DD}$, the static current is zero, and the static power dissipation is zero. When $v_I = V_{DD}$, the inverter will operate at the point labeled E in Fig. x9.8. Observe that unlike standard CMOS, here V_{OL} is not zero, an obvious disadvantage. Another disadvantage is that the gate conducts current (I_{stat}) in the low-output state, and thus there will be static power dissipation ($P_D = I_{stat} \times V_{DD}$).

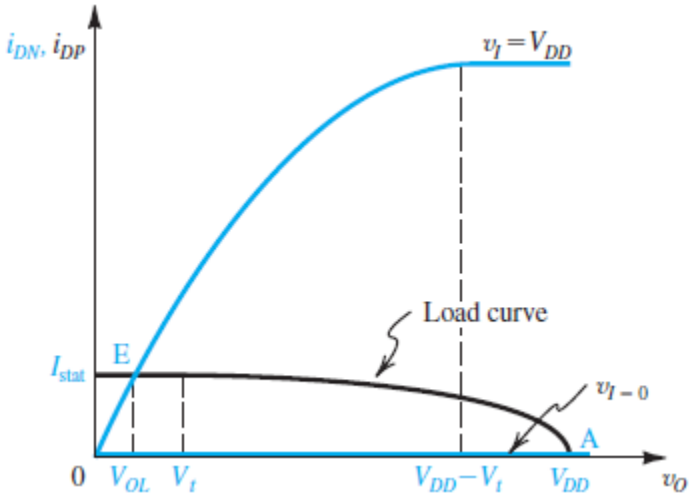

Figure x9.8 Graphical construction to determine the VTC of the inverter in Fig. x9.7(a).

x9.4.3 Derivation of the VTC

Figure x9.9 shows the VTC of the pseudo-NMOS inverter. As indicated, it has four distinct regions, labeled I through IV, corresponding to the different combinations of possible modes of operation of Q_N and Q_P. The four regions, the corresponding transistor modes of operation, and the conditions that define the regions are listed in Table x9.1. We shall utilize the information in this table together with the device equations given in Eqs. (x9.14) through (x9.17) to derive expressions for the various segments of the VTC and in particular for the important parameters that characterize the static operation of the inverter.

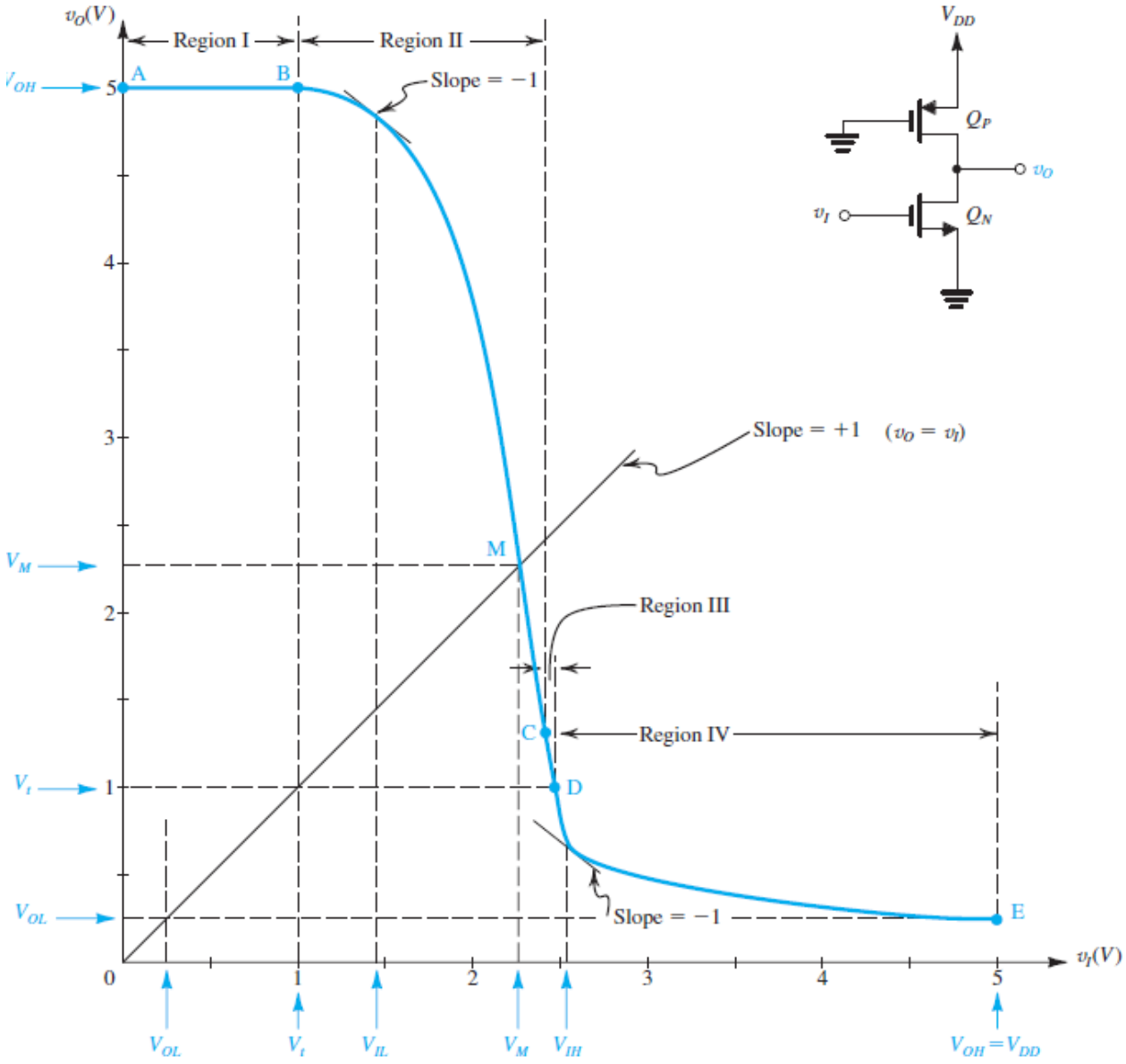

Figure x9.9 VTC for the pseudo-NMOS inverter. This curve is plotted for V_{DD}=5 V, V_{tn}=−V_{tp}= 1 V, and r = 9.

Table x9.1 Regions of Operation of the Pseudo-NMOS Inverter

Region	Segment of VTC	Q_N	Q_P	Condition
I	AB	Cutoff	Triode	$v_I < V_t$
II	BC	Saturation	Triode	$v_O \geq v_I - V_t$
III	CD	Triode	Triode	$V_t \leq v_O \leq v_I - V_t$
IV	DE	Triode	Saturation	$v_O \leq V_t$

- **Region I (segment AB):**

$$v_O = V_{OH} = V_{DD} \tag{x9.18}$$

- **Region II (segment BC):**

Equating iDN from Eq. (x9.14) and iDP from Eq. (x9.17) together with substituting kn = rkp, and with some manipulations, we obtain

$$v_O = V_t + \sqrt{(V_{DD} - V_t)^2 - r(v_I - V_t)^2} \tag{x9.19}$$

The value of V_{IL} can be obtained by differentiating this equation and substituting $\partial v_O/\partial v_I = -1$ and $v_I = V_{IL}$:

$$V_{IL} = V_t + \frac{V_{DD} - V_t}{\sqrt{r(r+1)}} \tag{x9.20}$$

The threshold voltage V_M is by definition the value of v_I for which $v_O = v_I$,

$$V_M = V_t + \frac{V_{DD} - V_t}{\sqrt{r+1}} \tag{x9.21}$$

Finally, the end point of the region II segment (point C) can be found by substituting $v_O = v_I - V_t$ in Eq. (x9.19), the condition for Q_N leaving saturation and entering the triode region.

- **Region III (segment CD)**

This is a short segment that is not of great interest. Point D is characterized by $v_O = V_t$.

- **Region IV (segment DE)**

Equating i_{DN} from Eq. (x9.15) to i_{DP} from Eq. (x9.16) and substituting $k_n = rk_p$ results in

$$v_O = (v_I - V_t) - \sqrt{(v_I - V_t)^2 - \frac{1}{r}(V_{DD} - V_t)^2} \tag{x9.22}$$

The value of V_{IH} can be determined by differentiating this equation and setting $\partial v_O/\partial v_I = -1$ and $v_I = V_{IH}$,

$$V_{IH} = V_t + \frac{2}{\sqrt{3r}}(V_{DD} - V_t) \tag{x9.23}$$

The value of V_{OL} can be found by substituting $v_I = V_{DD}$ into Eq. (x9.22),

$$V_{OL} = (V_{DD} - V_t)\left[1 - \sqrt{1 - \frac{1}{r}}\right] \tag{x9.24}$$

The static current conducted by the inverter in the low-output state is found from Eq. (x9.16) as

$$I_{\text{stat}} = \frac{1}{2}k_p(V_{DD} - V_t)^2 \tag{x9.25}$$

Finally, we can use Eqs. (x9.20) and (x9.24) to determine NM_L and Eqs. (x9.18) and (x9.23) to determine NM_H:

$$NM_L = V_t - (V_{DD} - V_t)\left[1 - \sqrt{1 - \frac{1}{r}} - \frac{1}{\sqrt{r(r+1)}}\right] \tag{x9.26}$$

$$NM_H = (V_{DD} - V_t)\left(1 - \frac{2}{\sqrt{3r}}\right) \tag{x9.27}$$

As a final observation, we note that since V_{DD} and V_t are determined by the process technology, the only design parameter for controlling the values of V_{OL} and the noise margins is the ratio r.

x9.4.4 Dynamic Operation

Analysis of the inverter transient response to determine t_{PLH} with the inverter loaded by a capacitance C is identical to that of the complementary CMOS inverter. The capacitance will be charged by the current i_{DP}; we can determine an estimate for t_{PLH} by using the average value of i_{DP} over the range $v_O = 0$ to $v_O = V_{DD}/2$. The result is:

$$t_{PLH} = \frac{\alpha_p C}{k_p V_{DD}} \tag{x9.28}$$

where

$$\alpha_p = 2 \Big/ \left[\frac{7}{4} - 3\left(\frac{V_t}{V_{DD}}\right) + \left(\frac{V_t}{V_{DD}}\right)^2 \right] \tag{x9.29}$$

The case for the capacitor discharge is somewhat different because the current i_{DP} has to be subtracted from i_{DN} to determine the discharge current. The result is

$$t_{PLH} \simeq \frac{\alpha_n C}{k_n V_{DD}} \tag{x9.30}$$

where

$$\alpha_n = 2 \Big/ \left[1 + \frac{3}{4}\left(1 - \frac{1}{r}\right) - \left(3 - \frac{1}{r}\right)\left(\frac{V_t}{V_{DD}}\right) + \left(\frac{V_t}{V_{DD}}\right)^2 \right] \tag{x9.31}$$

which, for a large value of r, reduces to

$$\alpha_n \simeq \alpha_p \tag{x9.32}$$

Although these are similar formulas to those for the standard CMOS inverter, the pseudo-NMOS inverter has a special problem: Since k_p is r times smaller than k_n, t_{PLH} will be approximately r times larger than t_{PHL}. Thus the circuit exhibits an asymmetrical delay performance. Recall, however, that for gates with large fan-in, pseudo-NMOS requires fewer transistors and thus C can be smaller than in the corresponding standard CMOS gate.

x9.4.5 Design

The design involves selecting the ratio r and the W/L for one of the transistors. The value of W/L for the other device can then be obtained using r. The design parameters of interest are V_{OL}, NM_L, NM_H, I_{stat}, P_D, t_{PLH}, and t_{PHL}. Important design considerations are as follows:

1. The ratio r determines all the breakpoints of the VTC; the larger the value of r, the lower V_{OL} is (Eq. x9.24) and the wider the noise margins are (Eqs. x9.26 and x9.27). However, a larger r increases the asymmetry in the dynamic response and, for a given $(W/L)_p$, makes the silicon area larger. Thus, selecting a value for r represents a compromise between noise margins on the one hand and silicon area and t_P on the other. Usually, r is selected in the range 4 to 10.
2. Once r has been determined, a value for $(W/L)_p$ or $(W/L)_n$ can be selected and the other determined. Here, one would select a small $(W/L)_n$ to keep the gate area small and thus obtain a small value for C. Similarly, a small $(W/L)_p$ keeps I_{stat} and P_D low. On the other hand, one would want to select larger W/L ratios to obtain low t_P and thus fast response. For usual (high-speed) applications, $(W/L)_p$ is selected so that I_{stat} is in the range of 50 μA to 100 μA, which for V_{DD} = 1.8 V results in P_D in the range of 90 μW to 180 μW.

x9.4.6 Gate Circuits

Except for the load device, the pseudo-NMOS gate circuit is identical to the PDN of the complementary CMOS gate. Four-input, pseudo-NMOS NOR and NAND gates are shown in Fig. x9.10. Note that each requires five transistors compared to the eight used in standard CMOS. In pseudo-NMOS, NOR gates are preferred over NAND gates because the former do not utilize transistors in series and thus can be designed with minimum-size NMOS devices.

x9.4.7 Concluding Remarks

Pseudo-NMOS is particularly suited for applications in which the output remains high most of the time. In such applications, the static power dissipation can be reasonably low (since the gate dissipates static power only in the low-output state). Further, the output transitions that matter would presumably be high-to-low ones, where the propagation delay can be made as short as necessary. A particular application of this type can be found in the design of address decoders for memory chips and in read-only memories (see Chapter 18 of the eighth edition).

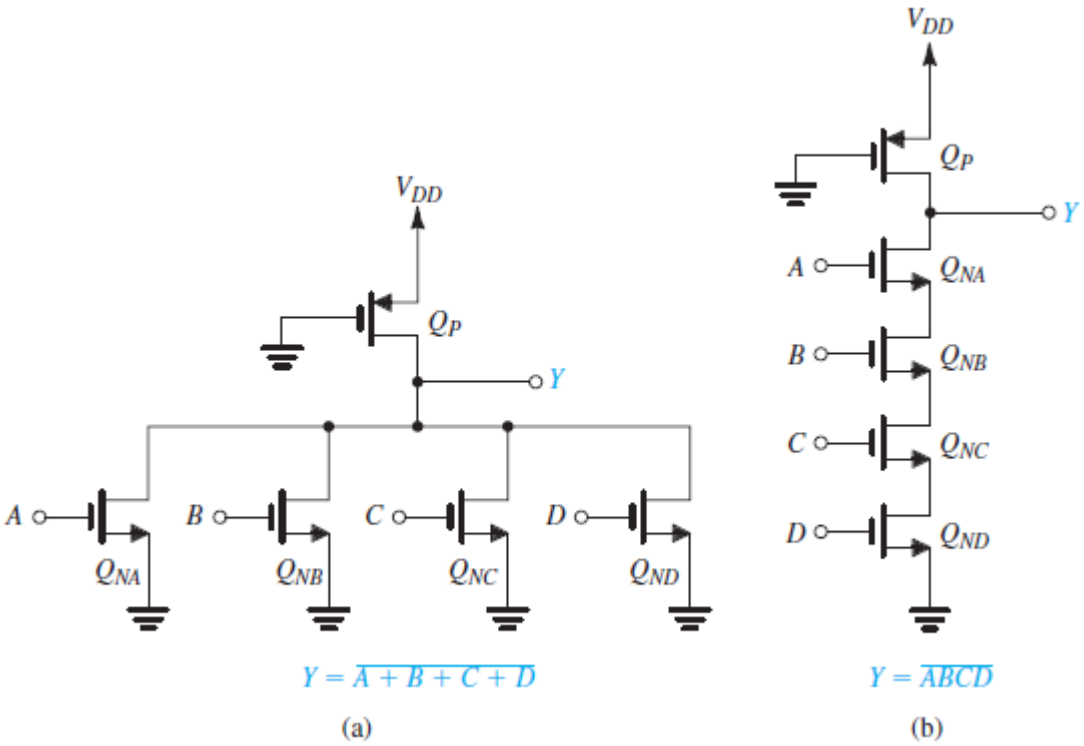

Figure x9.10 NOR and NAND gates of the pseudo-NMOS type.

Example x9.2

Consider a pseudo-NMOS inverter fabricated in a 0.25-μm CMOS technology for which $\mu_n C_{ox}$ = 115 μA/V2, $\mu_p C_{ox}$ = 30 μA/V^2, $V_{tn} = -V_{tp}$ = 0.5 V, and V_{DD} = 2.5 V. Let the W/L ratio of Q_P be (0.25 μm/0.25 μm) and r = 9. Find:

(a) V_{OH}, V_{OL}, V_{IL}, V_{IH}, V_M, N_{MH}, and N_{ML}

(b) $(W/L)_n$

(c) I_{stat} and P_D

(d) t_{PLH}, t_{PHL}, and t_p, assuming a total capacitance at the inverter output of 7 fF

Solution

(a) $V_{OH} = V_{DD} = 2.5$ V

V_{OL} is determined from Eq. (x9.24) as

$$V_{OL} = (2.5 - 0.5)\left[1 - \sqrt{1 - \frac{1}{9}}\right] = 0.11 \text{ V}$$

V_{IL} is determined from Eq. (x9.20) as

$$V_{IL} = 0.5 + \frac{1.5 - 0.5}{\sqrt{9(9+1)}} = 0.71 \text{ V}$$

V_{IH} is determined from Eq. (x9.23) as

$$V_{IH} = 0.5 + \frac{2}{\sqrt{3 \times 9}} \times (2.5 - 0.5) = 1.27 \text{ V}$$

V_M is determined from Eq. (x9.21) as

$$V_M = 0.5 + \frac{2.5 - 0.5}{\sqrt{9+1}} = 1.13 \text{ V}$$

The noise margins can now be determined as

$$NM_H = V_{OH} - V_{IH} = 2.5 - 1.27 = 1.23 \text{ V}$$
$$NM_L = V_{IL} - V_{OL} = 0.71 - 0.11 = 0.60 \text{ V}$$

Observe that the noise margins are not equal and that NM_L is rather low.

(b) The W/L ratio of Q_N can be found from

$$\frac{\mu_n C_{ox}(W/L)_n}{\mu_p C_{ox}(W/L)_p} = 9$$
$$\frac{115 \times (W/L)_n}{30 \times 1} = 9$$

Thus,

$$(W/L)_n = 2.35$$

(c) The dc current in the low-output state can be determined from Eq. (x9.25) as

$$I_{\text{stat}} = \frac{1}{2} \times 30 \times 1(2.5 - 0.5)^2 = 60\ \mu\text{A}$$

The static power dissipation can now be found from

$$\begin{aligned} P_D &= I_{\text{stat}} V_{DD} \\ &= 60 \times 2.5 = 150\ \mu\text{W} \end{aligned}$$

(d) The low-to-high propagation delay can be found by using Eqs. (x9.28) and (x9.29):

$$\alpha_p = 1.68$$

$$t_{PLH} = \frac{1.68 \times 7 \times 10^{-15}}{30 \times 10^{-6} \times 1 \times 2.5} = 0.16\ \text{ns}$$

The high-to-low propagation delay can be found by using Eqs. (x9.30) and (x9.31):

$$\alpha_p = 1.77$$

$$t_{PLH} = \frac{1.77 \times 7 \times 10^{-15}}{115 \times 10^{-6} \times 2.35 \times 2.5} = 0.02\ \text{ns}$$

Now, the propagation delay can be determined, as

$$t_{PHL} = \frac{1}{2}(0.16 + 0.02) = 0.09\ \text{ns}$$

Although the propagation delay is considerably greater than that of a standard CMOS inverter, this is not an entirely fair comparison: Recall that the advantage of pseudo-NMOS occurs in gates with large fan-in, not in a single inverter.

EXERCISES

x9.4 While keeping r unchanged, redesign the inverter circuit of Example x9.2 to lower its static power dissipation to half the value found. Find the W/L ratios for the new design. Also find t_{PLH}, t_{PHL}, and t_P, assuming that C remains unchanged. Would the noise margins change?

Ans. $(W/L)_n = 1.18$; $(W/L)_p = 0.5$; 0.32 ns; 0.04 ns; 0.18 ns; no

x9.5 Redesign the inverter of Example x9.2 using $r = 4$. Find V_{OL} and the noise margins. If $(W/L)_n = 0.375\ \mu\text{m}/0.25\ \mu\text{m}$, find $(W/L)_p$, I_{stat}, P_D, t_{PLH}, t_{PHL}, and t_P. Assume $C = 7$ fF.

Ans. $V_{OL} = 0.27$ V; $NM_L = 0.68$ V; $NM_H = 0.85$ V; $(W/L)_p = 1.44$; $I_{\text{stat}} = 86.3\ \mu$A; $P_D = 0.22$ mW; $t_{PLH} = 0.11$ ns; $t_{PHL} = 0.03$ ns; $t_P = 0.07$ ns

x9.5 Dynamic MOS Logic Circuits

The logic circuits that we have studied thus far are of the static type. In a static logic circuit, every node has, at all times, a low-resistance path to V_{DD} or ground. By the same token, the voltage of each node is well defined at all times, and no node is left floating. Static circuits do not need clocks (i.e., periodic timing signals) for their operation, although clocks may be present for other purposes. In contrast, the dynamic logic circuits we are about to discuss rely on the storage of signal voltages on parasitic capacitances at certain circuit nodes. Since charge will leak away with time, the circuits need to be *periodically refreshed;* thus the presence of a clock with a certain specified minimum frequency is essential.

To place dynamic logic-circuit techniques into perspective, let's take stock of the various styles we have studied for logic circuits. Standard CMOS excels in nearly every performance category: It is easy to design, has the maximum possible logic swing, is robust from a noise-immunity standpoint, dissipates no static power, and can be designed to provide equal low-to-high and high-to-low propagation delays. Its main disadvantage is the requirement of two transistors for each additional gate input, which for high fan-in gates can make the chip area large and increase the total capacitance and, correspondingly, the propagation delay and the dynamic power dissipation. Pseudo-NMOS reduces the number of required transistors at the expense of static power dissipation. Pass-transistor logic can result in simple small-area circuits but is limited to special applications and requires the use of CMOS inverters to restore signal levels, especially when the switches are simple NMOS transistors. The dynamic logic techniques studied in this section maintain the low device count of pseudo-NMOS while reducing the static power dissipation to zero. As will be seen, this is achieved at the expense of more complex, and less robust, design.

x9.5.1 The Basic Principle

Figure x9.11(a) shows the basic dynamic logic gate. It consists of a pull-down network (PDN) that realizes the logic function in exactly the same way as the PDN of a standard CMOS gate or a pseudo-NMOS gate. Here, however, we have two switches in series that are periodically operated by the clock signal ϕ whose waveform is shown in Fig. x9.11(b). When ϕ is low, Q_p is turned on, and the circuit is said to be in the setup or **precharge phase**. When ϕ is high, Q_p is off and Q_e turns on, and the circuit is in the **evaluation phase**. Finally, note that C_L denotes the total capacitance between the output node and ground.

During precharge, Q_p conducts and charges capacitance C_L so that at the end of the precharge interval, the voltage at Y is equal to V_{DD}. Also during precharge, the inputs A, B, and C are allowed to change and settle to their proper values. Observe that because Q_e is off, no path to ground exists.

During the evaluation phase, Q_p is off and Q_e is turned on. Now, if the input combination is one that corresponds to a high output, the PDN does not conduct (just as in a standard CMOS gate) and the output remains high at V_{DD}; thus $V_{OH} = V_{DD}$. Observe that no low-to-high propagation delay is required, thus $t_{PLH} = 0$. On the other hand, if the combination of inputs is one that corresponds to a low output, the appropriate NMOS transistors in the PDN will conduct and establish a path between the output node and ground through the on transistor Q_e. Thus C_L will be discharged through the PDN, and the voltage at the output node will reduce to $V_{OL} = 0$ V. The high-to-low propagation delay t_{PHL} can be calculated in exactly the same way as for a standard CMOS circuit, except that here we have an additional transistor, Q_e, in the series path to ground. Although this will

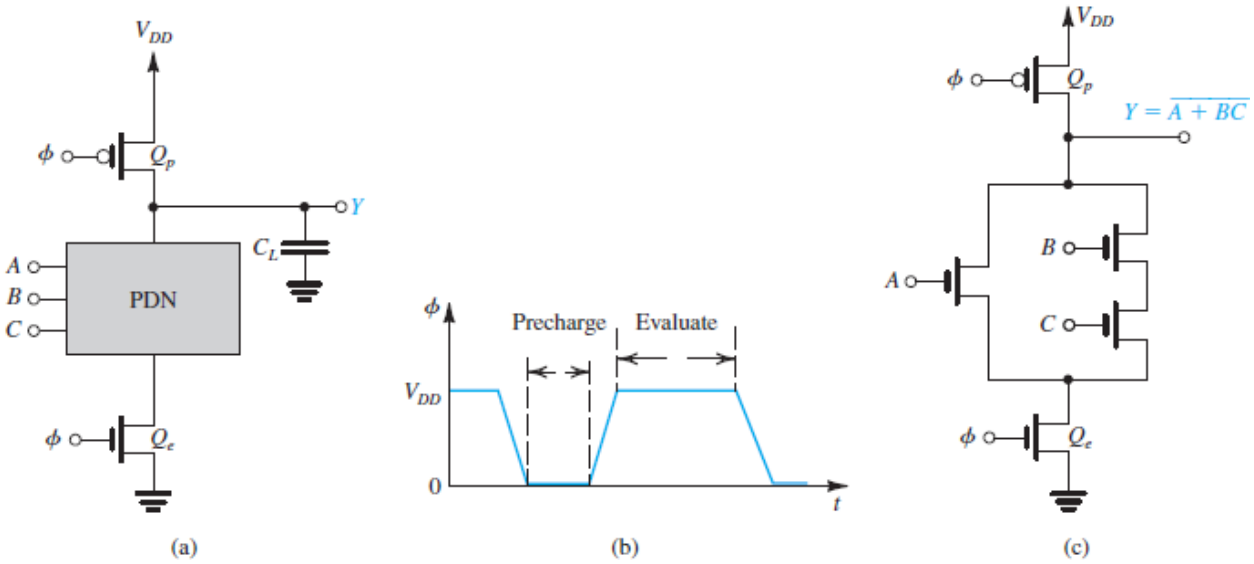

Figure x9.11 (a) Basic structure of dynamic MOS logic circuits. (b) Waveform of the clock needed to operate the dynamic logic circuit. (c) An example circuit.

increase the delay slightly, the increase will be more than offset by the reduced capacitance at the output node as a result of the absence of the PUN.

As an example, we show in Fig. x9.11(c) the circuit that realizes the function $Y = \overline{A + BC}$. Sizing of the PDN transistors often follows the same procedure employed in the design of static CMOS. For Q_p, we select a W/L ratio large enough to ensure that C_L will be fully charged during the precharge interval, but small enough so that the capacitance C_L will not be increased significantly. This is a ratioless form of MOS logic, where the output levels do not depend on the transistors' W/L ratios (unlike pseudo-NMOS, for instance).

Example x9.3

Consider the four-input, dynamic logic NAND gate shown in Fig. x9.12(a). Assume that the gate is fabricated in a 0.18-μm CMOS technology for which V_{DD} = 1.8 V, V_t = 0.5 V, and $\mu_n C_{ox}$ = 4 $\mu_p C_{ox}$ = 300 μA/V^2. To keep C_L small, NMOS devices with W/L = 0.27 μm/0.18 μm are used (including transistor Q_e). The PMOS precharge transistor Q_p has W/L = 0.54 μm/0.18 μm. The total capacitance C_L is found to be 20 fF.

(a) Consider the precharge operation [Fig. x9.12(b)] with the gate of Q_p at 0 V, and assume that at t = 0, C_L is fully discharged. Calculate the rise time of the output voltage, defined as the time for v_Y to rise from 10% to 90% of the final voltage V_{DD}.

(b) For $A = B = C = D = 1$, find the value of t_{PHL}.

Solution

(a) From Fig. x9.12(b) we see that at $v_Y = 0.1\ V_{DD}$ = 0.18 V, Q_p will be operating in the saturation region and i_D will be

$$i_D(0.1\,V_{DD}) = \frac{1}{2}\mu_p C_{ox}\left(\frac{W}{L}\right)_p \left(V_{DD} - |V_{tp}|\right)^2$$
$$= \frac{1}{2} \times 75 \times \frac{0.54}{0.18}(1.8 - 0.5)^2$$
$$= 190.1\ \mu\text{A}$$

At $v_Y = 0.9\,V_{DD} = 1.62$ V, Q_p will be operating in the triode region; thus,

$$i_D(0.9\,V_{DD}) = \mu_p C_{ox}\left(\frac{W}{L}\right)_p \left[(V_{DD} - |V_{tp}|)(V_{DD} - 0.9\,V_{DD}) - \frac{1}{2}(V_{DD} - 0.9\,V_{DD})^2\right]$$
$$= 75 \times \frac{0.54}{0.18}\left[(1.8 - 0.5)(1.8 - 1.62) - \frac{1}{2}(1.8 - 1.62)^2\right]$$
$$= 49\ \mu\text{A}$$

Thus the average capacitor charging current is

$$I_{av} = \frac{1}{2}(190.1 + 49) = 119.6\ \mu\text{A}$$

The rise time t_r of v_Y can now be determined from

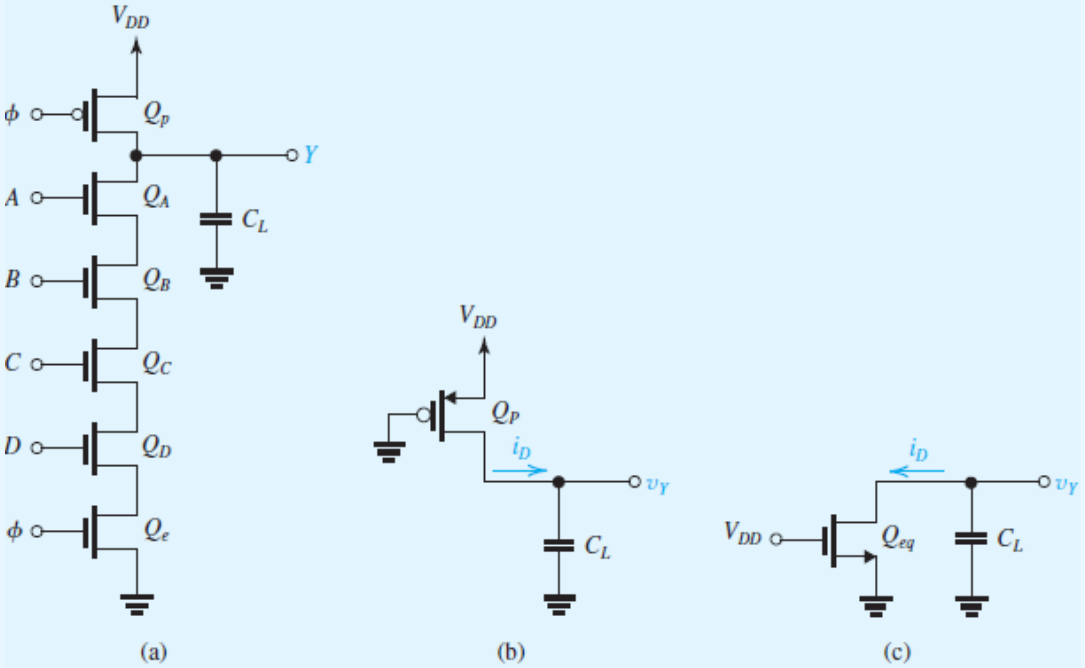

Figure x9.12 Circuits for Example x9.3.

$$t_r = \frac{C\Delta v_Y}{I_{av}}$$
$$= \frac{C(0.9\,V_{DD} - 0.1\,V_{DD})}{I_{av}}$$

Thus,

$$t_r = \frac{20 \times 10^{-15} \times 0.8 \times 1.8}{119.6 \times 10^{-6}} = 0.24 \text{ ns}$$

(b) When $A = B = C = D = 1$, all the NMOS transistors will be conducting during the evaluation phase. Replacing the five identical transistors with an equivalent device Q_{eq} with $(W/L)_{eq} = 1/5 \times 1.5 = 0.3$, we obtain the equivalent circuit for the capacitor discharge, shown in Fig. x9.12(c). At $v_Y = V_{DD}$, Q_{eq} will be operating in saturation; thus,

$$i_D(V_{DD}) = \frac{1}{2}(\mu_n C_{ox})\left(\frac{W}{L}\right)_{\text{eq}}(V_{DD} - V_t)^2$$
$$= \frac{1}{2} \times 300 \times 0.3(1.8 - 0.5)^2$$
$$= 76.1\ \mu\text{A}$$

At $v_Y = V_{DD}/2$, Q_{eq} will be operating in the triode region, thus,

$$i_D(V_{DD}/2) = (\mu_n C_{ox})\left(\frac{W}{L}\right)_{\text{eq}}\left[(V_{DD} - V_t)\frac{V_{DD}}{2} - \frac{1}{2}\left(\frac{V_{DD}}{2}\right)^2\right]$$
$$= 300 \times 0.3\left[(1.8 - 0.5)\left(\frac{1.8}{2}\right) - \frac{1}{2}\left(\frac{1.8}{2}\right)^2\right]$$
$$= 68.9\ \mu\text{A}$$

Thus the average capacitor-discharge current is

$$I_{av} = \frac{76.1 + 68.9}{2} = 72.5\ \mu\text{A}$$

and t_{PHL} can be found from

$$t_{PHL} = \frac{C(V_{DD} - V_{DD}/2)}{I_{\text{av}}}$$
$$= \frac{20 \times 10^{-15}(1.8 - 0.9)}{72.5 \times 10^{-6}}$$
$$= 0.25 \text{ ns}$$

EXERCISE

x9.6 In an attempt to reduce t_{PHL} of the NAND gate in Example x9.3, the designer doubles the value of W/L of each of the NMOS devices. If C increases to 30 fF, what is the new value of t_{PHL}?

Ans. 0.19 ns

GRAND-SCALE GRAPHICS

IC chips for specialized graphics processing achieve new levels of integration. Among recent announcements from Nividia is a graphics chip (GPU) incorporating 7.1 billion MOS transistors on a 551-mm^2 die in 28-nm CMOS technology from Taiwan Semiconductor Manufacturing Company (TSMC). Besides gaming and graphics, live-streaming video and other applications abound.

x9.5.2 Nonideal Effects

We now briefly consider various sources of nonideal operation of dynamic logic circuits.

Noise Margins Since, during the evaluation phase, the NMOS transistors begin to conduct for $V_i = V_{tn}$,

$$VI_L \simeq V_{IH} \simeq V_{tn}$$

and thus the noise margins will be

$$NM_L = V_{IL} - V_{OL} = V_{tn} - 0 = V_{tn}$$
$$NM_H = V_{OH} - V_{IH} = V_{DD} - Vtn$$

Thus the noise margins are far from equal, and NM_L is rather low. Although NM_H is high, other nonideal effects reduce its value, as we shall shortly see. At this time, however, observe that the output node is a high-impedance node and thus will be susceptible to noise pickup and other disturbances.

Output Voltage Decay due to Leakage Effects In the absence of a path to ground through the PDN, the output voltage will ideally remain high at V_{DD}. This, however, is based on the assumption that the charge on C_L will remain intact. In practice, there will be leakage current that will cause C_L to slowly discharge and v_Y to decay. The principal source of leakage is the reverse current of the reverse-biased junction between the drain diffusion of transistors connected to the output node and the substrate. Such currents can be in the range of 10^{-12} A to 10^{-15} A, and they increase rapidly with temperature (approximately doubling for every 10°C rise in temperature). Thus the circuit can malfunction if the clock is operating at a very low frequency and the output node is not "refreshed" periodically. This exact same point is encountered for dynamic memory cells in Chapter 18 of the eighth edition.

Charge Sharing There is another and often more serious way for C_L to lose some of its charge and thus cause v_Y to fall significantly below V_{DD}. To see how this can happen, refer to Fig. x9.13(a), which shows only Q_1 and Q_2, the two top transistors of the PDN, together with the precharge transistor Q_p. Here, C_1 is the capacitance between the common node of Q_1 and Q_2 and ground. At the beginning of the evaluation phase, after Q_p has turned off and with C_L charged to V_{DD} [Fig. x9.13(a)], we assume that C_1 is initially discharged and that the inputs are such that at the gate of Q_1 we have a high signal, whereas at the gate of Q_2 the signal is low. We can easily see that Q_1 will turn on and its drain current, i_{D1}, will flow as indicated. Thus i_{D1} will discharge C_L and charge C_1. Although eventually i_{D1} will reduce to zero, C_L will have lost some of its charge, which will have been transferred to C_1. This phenomenon is known as charge sharing.

We shall not pursue the problem of charge sharing any further here, except to point out a couple of the techniques usually employed to minimize its effect. One approach involves adding a *p*-channel device that continuously conducts a small current to replenish the charge lost by C_L, as shown in Fig. x9.13(b). This arrangement should remind us of pseudo-NMOS. Indeed, adding this transistor will cause the gate to dissipate static power. On the positive side, however, the added transistor will lower the impedance level of the output node and make it less susceptible to noise and will solve the leakage and charge-sharing problems. Another approach to solving the charge-sharing problem is to precharge the internal nodes: that is, to precharge capacitor C_1. The price paid in this case is increased circuit complexity and node capacitances.

Clock Feedthrough Another problem can arise when the PDN remains off during the evaluation phase. As ϕ rises and turns off Q_p, the output node Y becomes a floating node. However, Y is capacitively coupled to ϕ through C_{gd} of Q_p, and hence the clock signal ϕ can cause a slight rise in output voltage.

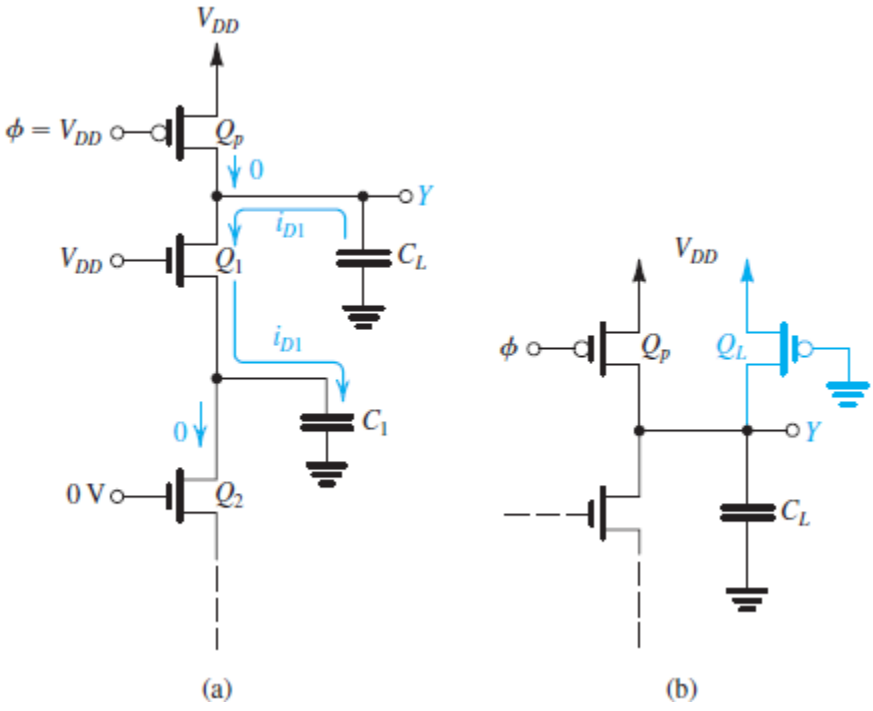

Figure x9.13 **(a)** Charge sharing. **(b)** Adding a permanently turned-on transistor Q_L solves the charge-sharing problem at the expense of static power dissipation.

Cascading Dynamic Logic Gates A serious problem arises if one attempts to cascade dynamic logic gates. Consider the situation depicted in Fig. x9.14, where two single-input dynamic gates are connected in cascade. During the precharge phase, C_{L1} and C_{L2} will be charged through Q_{p1} and Q_{p2}, respectively. Thus, at the end of the precharge interval, $v_{Y1} = V_{DD}$ and $v_{Y2} = V_{DD}$. Now consider what happens in the evaluation phase for the case of high input A. Obviously, the correct result will be Y_1 low ($v_{Y1} = 0$ V) and Y_2 high ($v_{Y2} = V_{DD}$). What happens, however, is somewhat different. As the evaluation phase begins, Q_1 turns on and C_{L1} begins to discharge. However, simultaneously, Q_2 turns on and C_{L2} also begins to discharge. Only when v_{Y1} drops below V_{tn} will Q_2 turn off. Unfortunately, however, by that time, C_{L2} will have lost a significant amount of its charge, and v_{Y2} will be less than the expected value of V_{DD}. (Here it is important to note that in dynamic logic, once charge has been lost, it cannot be recovered.) This problem is sufficiently serious to make simple cascading an impractical proposition. As usual, however, the ingenuity of circuit designers has come to the rescue, and a number of schemes have been proposed to make cascading possible in dynamic logic circuits. We shall discuss one such scheme after considering Exercise x9.7.

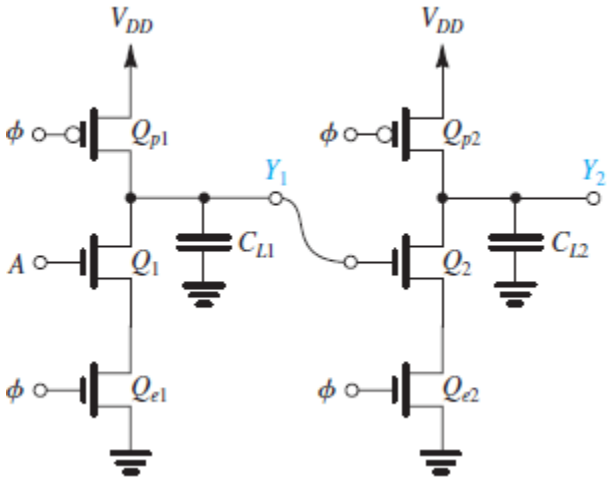

Figure x9.14 Two single-input dynamic logic gates connected in cascade. With the input A high, during the evaluation phase C_{L2} will partially discharge and the output at Y_2 will fall lower than V_{DD}, which can cause logic malfunction.

EXERCISE

x9.7 To gain further insight into the cascading problem described above, let us determine the decrease in the output voltage v_{Y2} for the circuit in Fig. x9.14. Specifically, consider the circuit as the evaluation phase begins: At $t = 0$, $v_{Y1} = v_{Y2} = V_{DD}$ and $v_\phi = v_A = V_{DD}$. Transistors Q_{p1} and Q_{p2} are cut off and can be removed from the equivalent circuit. Furthermore, for the purpose of this approximate analysis, we can replace the series combination of Q_1 and Q_{e1} with a single device having an appropriate W/L, and similarly for the combination of Q_2 and Q_{e2}. The result is the approximate equivalent circuit in Fig. xE9.7. We are interested in the operation of this circuit in the interval Δt during which v_{Y1} falls from V_{DD} to V_t , at which time Q_{eq2} turns off and C_{L2} stops discharging. Assume that the process technology has the parameter values specified in Example x9.3 and that for all NMOS transistors in the circuit of Fig. x9.14, $W/L = 4$ µm/2 µm and $C_{L1} = C_{L2} = 40$ fF.

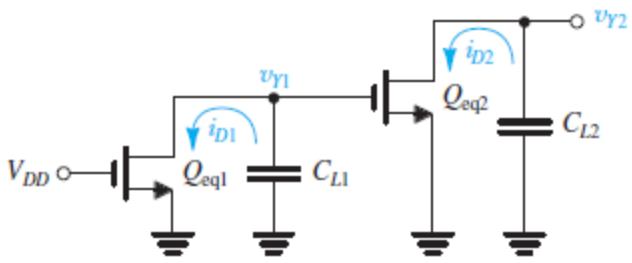

Figure xE9.7

(a) Find $(W/L)_{eq1}$ and $(W/L)_{eq2}$.

(b) Find the values of i_{D1} at $v_{Y1} = V_{DD}$ and at $v_{Y1} = V_t$. Hence determine an average value for i_{D1}.

(c) Use the average value of i_{D1} found in (b) to determine an estimate for the interval Δt.

(d) Find the average value of i_{D2} during Δt. To simplify matters, take the average to be the value of i_{D2} obtained when the gate voltage v_{Y1} is midway through its excursion (i.e., $v_{Y1} = 3$ V). (Hint: Q_{eq2} will remain in saturation.)

(e) Use the value of Δt found in (c) together with the average value of i_{D2} determined in (d) to find an estimate of the reduction in v_{Y2} during Δt. Hence determine the final value of v_{Y2}.

Ans. (a) 1, 1; (b) 400 μA and 175 μA, for an average value of 288 μA; (c) 0.56 ns; (d) 100 μA; (e) $v_{Y2} = 1.4$ V, thus v_{Y2} decreases to 3.6 V

x9.5.3 Domino CMOS Logic

Domino CMOS logic is a form of dynamic logic that results in cascadable gates. Figure x9.15 shows the structure of the Domino CMOS logic gate. We observe that it is simply the basic dynamic logic gate of Fig. x9.11(a) with a static CMOS inverter connected to its output. Operation of the gate is straightforward. During precharge, X will be raised to V_{DD}, and the gate output Y will be at 0 V. During evaluation, depending on the combination of input variables, either X will remain high and thus the output Y will remain low ($t_{PHL} = 0$) or X will be brought down to 0 V and the output Y will rise to V_{DD} (t_{PLH} finite). Thus, during evaluation, the output either remains low or makes only one low-to-high transition.

To see why Domino CMOS gates can be cascaded, consider the situation in Fig. x9.16(a), where we show two Domino gates connected in cascade. For simplicity, we show single-input gates. At the end of precharge, X_1 will be at V_{DD}, Y_1 will be at 0 V, X_2 will be at V_{DD}, and Y_2 will be at 0 V. As in the preceding case, assume that A is high at the beginning of evaluation. Thus, as ϕ goes up, capacitor C_{L1} will begin discharging, pulling X_1 down. Meanwhile, the low input at the gate of Q_2 keeps Q_2 off, and C_{L2} remains fully charged. When v_{X1} falls below the threshold voltage of inverter I_1, Y_1 will go up, turning Q_2 on, which in turn begins to discharge C_{L2} and pulls X_2 low. Eventually, Y_2 rises to V_{DD}.

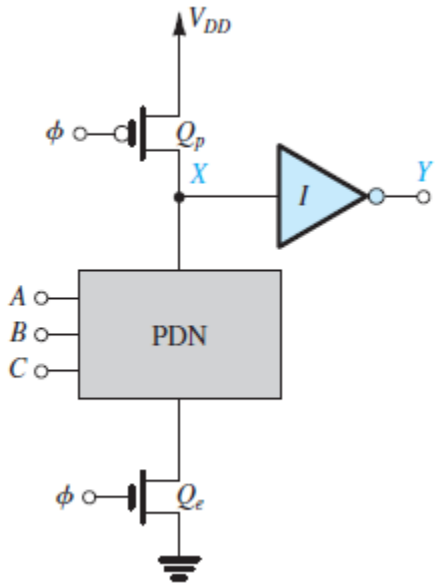

Figure x9.15 The Domino CMOS logic gate. The circuit consists of a dynamic MOS logic gate with a static CMOS inverter connected to the output. During evaluation, Y either will remain low (at 0 V) or will make one 0-to-1 transition (to V_{DD}).

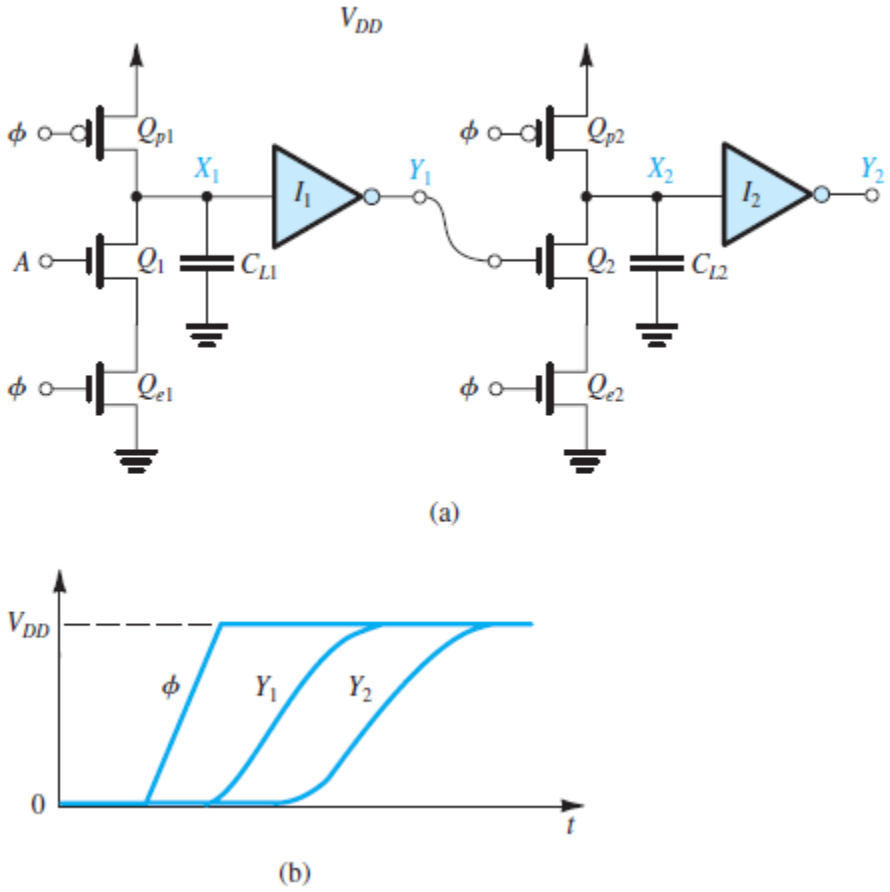

Figure x9.16 **(a)** Two single-input Domino CMOS logic gates connected in cascade. **(b)** Waveforms during the evaluation phase.

From this description, we see that because the output of the Domino gate is low at the beginning of evaluation, no premature capacitor discharge will occur in the subsequent gate in the cascade. As indicated in Fig. x9.16(b), output Y_1 will make a 0-to-1 transition t_{PLH} seconds after the rising edge of the clock. Subsequently, output Y_2 makes a 0-to-1 transition after another t_{PLH} interval. The propagation of the rising edge through a cascade of gates resembles contiguously placed dominoes falling over, each toppling the next, which is the origin of the name Domino CMOS logic. Domino CMOS logic finds application in the design of address decoders in memory chips, for example.

x9.5.4 Concluding Remarks

Dynamic logic presents many challenges to the circuit designer. Although it can provide high-speed operation, as well as considerable reduction in the chip-area requirement and zero (or little) static power dissipation, the circuits are prone to many nonideal effects, some of which have been discussed here. It should also be remembered that dynamic power dissipation is an important issue in dynamic logic. Another factor that should be considered is the "dead time" during precharge when the output of the circuit is not yet available.

x9.6 Semiconductor Memories: Types and Architectures

A computer system, whether a large machine or a microcomputer, requires memory for storing data and program instructions. Furthermore, within a given computer system there usually are various types of memory utilizing a variety of technologies and having different *access times*. Broadly speaking, computer memory can be divided into two types: **main memory** and **mass-storage** memory. The main memory is usually the most rapidly accessible memory and the one from which most, often all, instructions in programs are executed. The main memory is usually of the random-access type. A **random-access memory** (RAM) is one in which the time required for storing (writing) information and for retrieving (reading) information is independent of the physical location (within the memory) in which the information is stored.

Random-access memories should be contrasted with *serial* or *sequential* memories, such as disks and tapes, from which data are available only in the sequence in which the data were originally stored. Thus, in a serial memory the time to access particular information depends on the memory location in which the required information is stored, and the average access time is longer than the access time of random-access memory. In a computer system, serial memory is used for mass storage. Items not frequently accessed, such as large parts of the computer operating system, are usually stored in a *moving-surface memory* such as magnetic disk.

Another important classification of memory relates to whether it is a **read/write** or a **read-only memory**. Read/write (R/W) memory permits data to be stored and retrieved at comparable speeds. Computer systems require random-access read/write memory for data and program storage.

Read-only memories (**ROM**) permit reading at the same high speeds as R/W memories (or perhaps higher) but restrict the writing operation. ROMs can be used to store a microprocessor operating-system program. They are also employed in operations that require table lookup, such as finding the values of mathematical functions. A popular application of ROMs is their use in video game cartridges. It should be noted that read-only memory is usually of the random-access type. Nevertheless, in the digital circuit

jargon, the acronym RAM usually refers to read/write, random-access memory, while ROM is used for read-only memory.

The regular structure of memory circuits has made them an ideal application for the design of circuits of the very-large-scale integrated (VLSI) type. Indeed, at any moment, memory chips represent the state of the art in packing density and hence integration level. Beginning with the introduction of the 1-Kbit chip in 1970, memory-chip density has quadrupled about every 3 years. At the present time (2013), chips containing 4 Gbit are available. In this and the next two sections, we shall study some of the basic circuits employed in VLSI RAM chips. Read-only memory circuits are studied in Section x9.7 of the bonus materials.

x9.6.1 Memory-Chip Organization

The bits on a memory chip are addressable either individually or in groups of 4 to 16. As an example, a 64-Mbit chip in which all bits are individually addressable is said to be organized as 64M words × 1 bit (or simply 64M × 1). Such a chip needs a 26-bit address (2^{26} = 67,108,864 = 64M). On the other hand, the 64-Mbit chip can be organized as 16M words × 4 bits (16M × 4), in which case a 24-bit address is required. For simplicity we shall assume in our subsequent discussion that all the bits on a memory chip are individually addressable.

The bulk of the memory chip consists of the cells in which the bits are stored. Each **memory cell** is an electronic circuit capable of storing one bit. Such circuits are studied in Chapter 18 of the eighth edition. For reasons that will become clear shortly, it is desirable to physically organize the storage cells on a chip in a square or a nearly square matrix. Figure x9.17 illustrates such an organization. The cell matrix has 2^M rows and 2^N columns, for a total storage capacity of 2^{M+N}. For example, a 1M-bit square matrix would have 1024 rows and 1024 columns ($M = N = 10$). Each cell in the array is connected to one of the 2^M row lines, known rather loosely, but universally, as **word lines**, and to one of the 2^N column lines, known as **digit lines** or, more commonly, **bit lines**. A particular cell is **selected** for reading or writing by activating its word line and its bit line.

Activating one of the 2^M word lines is performed by the **row decoder**, a combinational logic circuit that selects (raises the voltage of) the particular word line whose M-bit address is applied to the decoder input. The address bits are denoted $A_0, A_1, \ldots, A_{M-1}$. When the Kth word line is activated for, say, **a read operation**, all 2^N cells in row K will provide their contents to their respective bit lines. Thus, if the cell in column L (Fig. x9.17) is storing a 1, the voltage of bit-line number L will be raised, usually by a small voltage, say 0.1 V to 0.2 V. The readout voltage is small because the cell is small, a deliberate design decision, since the number of cells is very large. The small readout signal is applied to a **sense amplifier** connected to the bit line. As Fig. x9.17 indicates, there is a sense amplifier for every bit line. The sense amplifier provides a full-swing digital signal (from 0 to V_{DD}) at its output. This signal, together with the output signals from all the other cells in the selected row, is then delivered to the **column decoder**. The column decoder selects the signal of the particular column whose N-bit address is applied to the decoder input (the address bits are denoted $A_M, A_{M+1}, \ldots, A_{M+N-1}$) and causes this signal to appear on the chip input/output (I/O) data line.

A **write operation** proceeds in a similar manner: The data bit to be stored (1 or 0) is applied to the I/O line. The cell in which the data bit is to be stored is selected through the combination of its row address and its column address. The sense amplifier of the selected column acts as a **driver** to write the applied signal into the selected cell. Circuits for sense amplifiers and address decoders are studied in Chapter 18 of the eighth edition.

Before leaving the topic of memory organization (or memory-chip architecture), we wish to mention a relatively recent innovation in organization dictated by the exponential increase in chip density. To appreciate the need for a change, note that as the number of cells in the array increases, the physical lengths of the word lines and the bit lines increase. This has occurred even though for each new generation of memory chips, the transistor size has decreased (currently, CMOS process technologies with 22-nm feature size are utilized). The net increase in word-line and bit-line lengths increases their total resistance and capacitance, and thus slows down their transient response. That is, as the lines lengthen, the exponential rise of the voltage of the word line becomes slower, and it takes longer for the cells to be activated. This problem has been solved by partitioning the memory chip into a number of blocks. Each of the blocks has an organization identical to that in Fig. x9.17. The row and column addresses are broadcast to all blocks, but the data selected come from only one of the blocks. Block selection is achieved by using an appropriate number of the address bits as a block address. Such an architecture can be thought of as three-dimensional: rows, columns, and blocks.

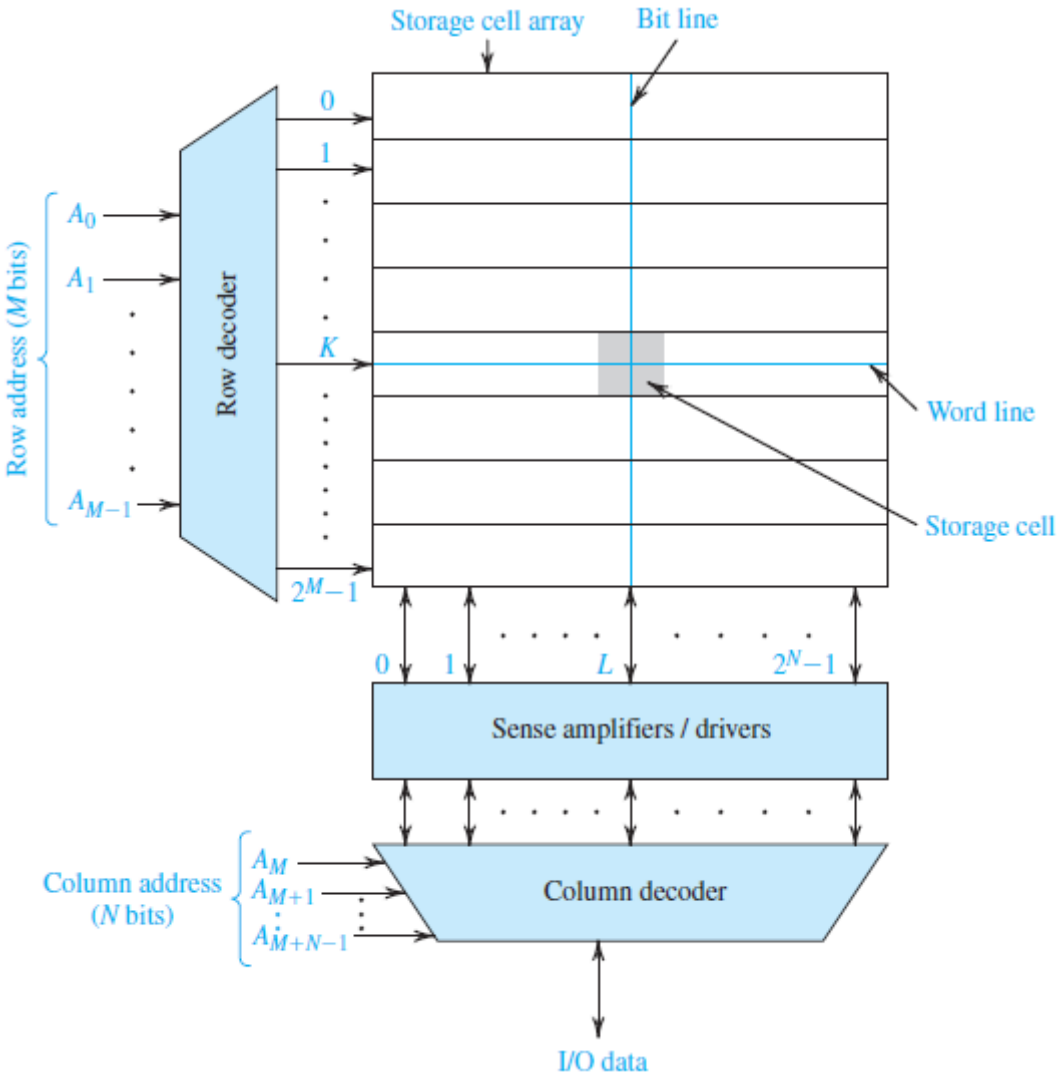

Figure x9.17 A 2^{M+N}-bit memory chip organized as an array of 2^M rows × 2^N columns.

x9.6.2 Memory-Chip Timing

The **memory access time** is the time between the initiation of a read operation and the appearance of the output data. The **memory cycle time** is the minimum time allowed between two consecutive memory operations. To be on the conservative side, a memory operation is usually taken to include both read and write (in the same location). MOS memories have access and cycle times in the range of a few to a few hundred nanoseconds.

EXERCISES

x9.8 A 4-Mbit memory chip is partitioned into 32 blocks, with each block having 1024 rows and 128 columns. Give the number of bits required for the row address, column address, and block address.

Ans. 10; 7; 5

x9.9 The word lines in a particular MOS memory chip are fabricated using polysilicon (see Appendix A). The resistance of each word line is estimated to be 5 kΩ, and the total capacitance between the line and ground is 2 pF. Find the time for the voltage on the word line to reach $V_{DD}/2$, assuming that the line is driven by a voltage V_{DD} provided by a low-impedance inverter. (*Note:* The line is actually a distributed network that we are approximating by means of a lumped circuit consisting of a single resistor and a single capacitor.)

Ans. 6.9 ns

x9.7 Read-Only Memory (ROM)

Read-only memory (ROM) is memory that contains fixed data patterns. It is used in a variety of digital system applications. Currently, a very popular application is the use of ROM in microprocessor systems to store the instructions of the system's basic operating program. ROM is particularly suited for such an application because it is nonvolatile; that is, it retains its contents when the power supply is switched off.

A ROM can be viewed as a combinational logic circuit for which the input is the collection of address bits of the ROM and the output is the set of data bits retrieved from the addressed location. This viewpoint leads to the application of ROMs in code conversion—that is, in changing the code of the signal from one system (say, binary) to another. Code conversion is employed, for instance, in secure communication systems, where the process is known as *scrambling*. It consists of feeding the code of the data to be transmitted to a ROM that provides corresponding bits in a (supposedly) secret code. The reverse process, which also uses a ROM, is applied at the receiving end.

In this section we will study various types of read-only memory. These include fixed ROM, which we refer to simply as ROM, programmable ROM (PROM), erasable programmable ROM (EPROM), and flash memory.

x9.7.1 A MOS ROM

Figure x9.18 shows a simplified 32-bit (or 8-word × 4-bit) MOS ROM. As indicated, the memory consists of an array of *n*-channel MOSFETs whose gates are connected to the word lines, whose sources are grounded, and whose drains are connected to the bit lines. Each bit line is connected to the power supply via a PMOS load transistor, in the manner of pseudo-NMOS logic (Section x9.4 of the bonus materials). An NMOS transistor exists in a particular cell if the cell is storing a 0; a cell storing a 1 has no MOSFET. This ROM can be thought of as 8 words of 4 bits each. The row decoder selects one of the 8 words by raising the voltage of the corresponding word line. The cell transistors connected to this word line will then conduct, thus pulling the voltage of the bit lines (to which transistors in the selected row are connected) down from V_{DD} to a voltage close to ground voltage (the logic-0 level). The bit lines that correspond to cells (of the selected word) without transistors (i.e., the cells that are storing a logic 1) will remain at the power-supply voltage (logic 1) because of the action of the pull-up PMOS load devices. In this way, the bits of the addressed word can be read.

A disadvantage of the ROM circuit in Fig. x9.18 is that it dissipates static power. When a word is selected, the transistors in this particular row will conduct static current supplied by the PMOS load transistors. Static power dissipation can be eliminated by a simple change. Rather than grounding the gate terminals of the PMOS transistors, we can connect them to a precharge line ϕ that is normally high. Just before a read operation, ϕ is lowered and the bit lines are precharged to V_{DD} through the PMOS transistors. The precharge signal ϕ then goes high, and the word line is selected. The bit lines that have transistors in the selected word are then discharged, indicating stored zeros, whereas those lines with no transistor present remain at V_{DD}, indicating stored ones.

EXERCISE

x9.10 Consider the ROM in Fig. x9.18 with the gates of the PMOS devices disconnected from ground and connected to a precharge control signal ϕ. Let all the NMOS devices have W/L = 6 μm/2 μm and all the PMOS devices have W/L = 24 μm/2 μm. Assume that $\mu_n C_{ox}$ = 50 μA/V2, $\mu_p C_{ox}$ = 20 μA/V2, $V_{tn} = -V_{tp}$ = 1 V, and V_{DD} = 5 V.

(a) During the precharge interval, ϕ is lowered to 0 V. Estimate the time required to charge a bit line from 0 V to 5 V. Use, as an average charging current, the current supplied by a PMOS transistor at a bit-line voltage halfway through the 0-V to 5-V excursion (i.e., 2.5 V). The bit-line capacitance is 2 pF. Note that all NMOS transistors are cut off at this time.

(b) After completion of the precharge interval and the return of ϕ to V_{DD}, the row decoder raises the voltage of the selected word line. Because of the finite resistance and capacitance of the word line, the voltage rises exponentially toward V_{DD}. If the resistance of each of the polysilicon word lines is 3 kΩ and the capacitance between the word line and ground is 3 pF, what is the (10% to 90%) rise time of the word-line voltage? What is the voltage reached at the end of one time constant?

(c) We account for the exponential rise of the word-line voltage by approximating the word-line voltage by a step equal to the voltage reached in one time constant. Find the interval Δt required for an NMOS transistor to discharge the bit line and lower its voltage by 0.5 V. (It is assumed that the sense amplifier needs a 0.5-V change at its input to detect a low bit value.)

Ans. (a) 6.1 ns; (b) 19.8 ns, 3.16 V; (c) 2.9 ns

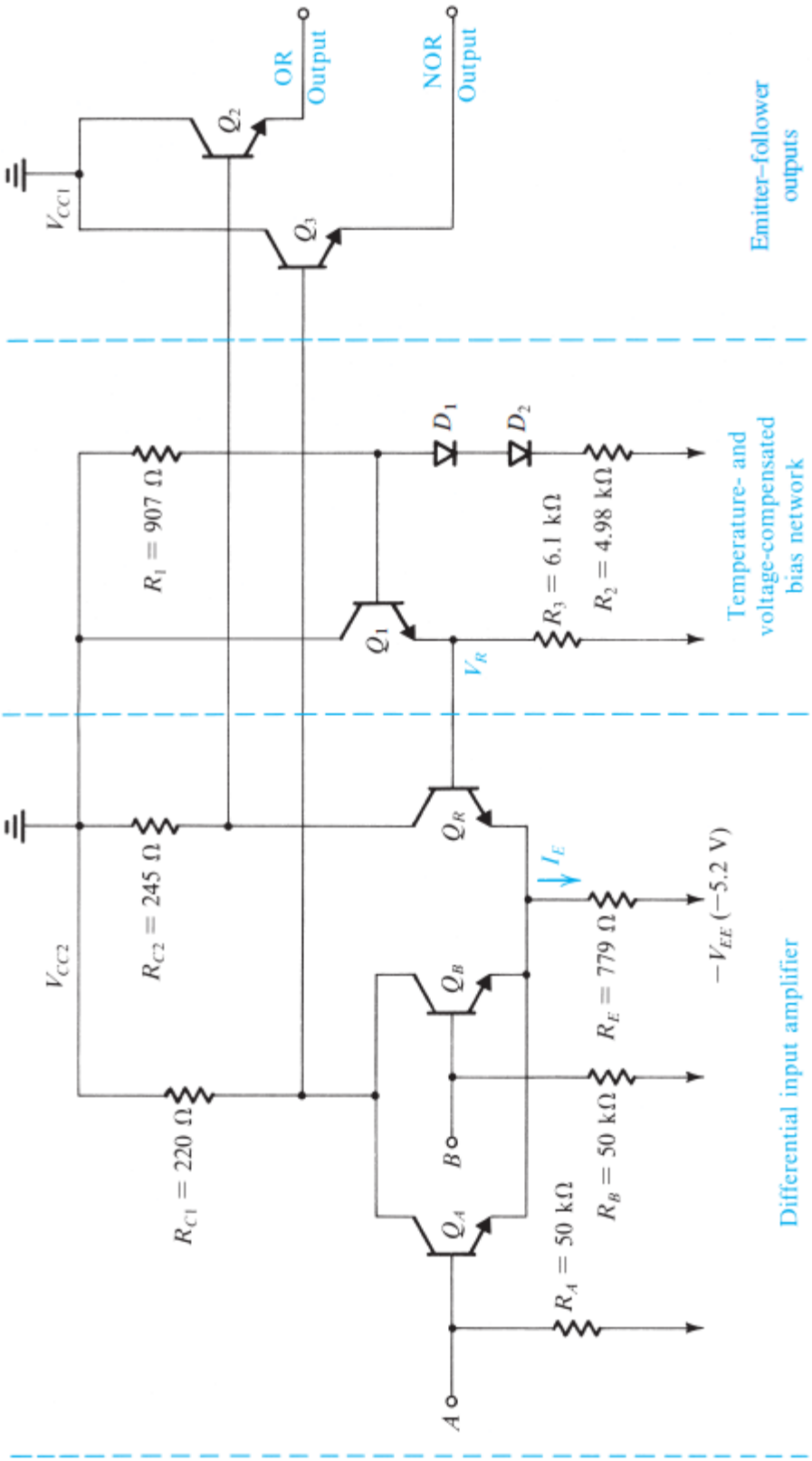

Figure x9.18 Basic circuit of the ECL 10K logic-gate family.

x9.7.2 Mask-Programmable ROMs

The data stored in the ROMs discussed thus far is determined at the time of fabrication, according to the user's specifications. However, to avoid having to custom-design each ROM from scratch (which would be extremely costly), ROMs are manufactured using a process known as **mask programming**. As explained in Appendix A, integrated circuits are fabricated on a wafer of silicon using a sequence of processing steps that include photomasking, etching, and diffusion. In this way, a pattern of junctions and interconnections is created on the surface of the wafer. One of the final steps in the fabrication process consists of coating the surface of the wafer with a layer of aluminum and then selectively (using a mask) etching away portions of the aluminum, leaving aluminum only where interconnections are desired. This last step can be used to program (i.e., to store a desired pattern in) a ROM. For instance, if the ROM is made of MOS transistors as in Fig. x9.18, MOSFETs can be included at all bit locations, but only the gates of those transistors where 0s are to be stored are connected to the word lines; the gates of transistors where 1s are to be stored are not connected. This pattern is determined by the mask, which is produced according to the user's specifications.

The economic advantages of the mask programming process should be obvious: All ROMs are built similarly; customization occurs only during the final steps in fabrication.

x9.7.3 Programmable ROMs (PROMs, EPROMs, and Flash)

PROMs are ROMs that can be programmed by the user, but only once. A typical arrangement employed in BJT PROMs involves using polysilicon fuses to connect the emitter of each BJT to the corresponding digit line. Depending on the desired content of a ROM cell, the fuse can be either left intact or blown out using a large current. The programming process is obviously irreversible.

An erasable programmable ROM, or EPROM, is a ROM that can be erased and reprogrammed as many times as the user wishes. It is therefore the most versatile type of read-only memory. It should be noted, however, that the process of erasure and reprogramming is time consuming and is intended to be performed only infrequently.

State-of-the-art EPROMs use variants of the memory cell whose cross section is shown in Fig. x9.19(a). The cell is basically an enhancement-type *n*-channel MOSFET with two gates made of polysilicon material. One of the gates is not electrically connected

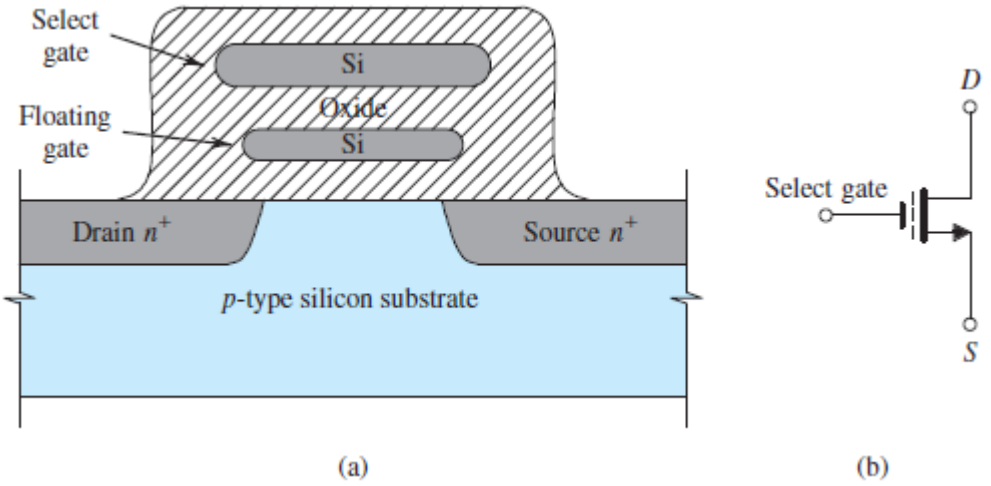

Figure x9.19 (a) Cross section and (b) circuit symbol of the floating-gate transistor used as an EPROM cell.

to any other part of the circuit; rather, it is left floating and is appropriately called a **floating gate**. The other gate, called a **select gate**, functions in the same manner as the gate of a regular enhancement MOSFET.

The MOS transistor of Fig. x9.19(a) is known as a **floating-gate transistor** and is given the circuit symbol shown in Fig. x9.19(b). In this symbol the broken line denotes the floating gate. The memory cell is known as the **stacked-gate cell**.

Let us now examine the operation of the floating-gate transistor. Before the cell is programmed (we will shortly explain what this means), no charge exists on the floating gate and the device operates as a regular *n*-channel enhancement MOSFET. It thus exhibits the i_D–v_{GS} characteristic shown as curve (*a*) in Fig. x9.20. Note that in this case the threshold voltage (V_t) is rather low. This state of the transistor is known as the **not-programmed state**. It is one of two states in which the floating-gate transistor can exist. Let us arbitrarily take the not-programmed state to represent a stored 1. That is, a floating-gate transistor whose i_D–v_{GS} characteristic is that shown as curve (*a*) in Fig. x9.20 will be said to be storing a 1.

To program the floating-gate transistor, a large voltage (16–20 V) is applied between its drain and source. Simultaneously, a large voltage (about 25 V) is applied to its select gate. Figure x9.21 shows the floating-gate MOSFET during programming. In the absence of any charge on the floating gate, the device behaves as a regular *n*-channel enhancement MOSFET: An *n*-type inversion layer (channel) is created at the wafer surface as a result of the large positive voltage applied to the select gate. Because of the large positive voltage at the drain, the channel has a tapered shape.

The drain-to-source voltage accelerates electrons through the channel. As these electrons reach the drain end of the channel, they acquire high kinetic energy and are referred to as *hot electrons*. The large positive voltage on the select gate (greater than the drain voltage) establishes an electric field in the insulating oxide. This electric field attracts the hot electrons and accelerates them (through the oxide) toward the floating gate. In this way the floating gate is charged, and the charge that accumulates on it becomes trapped.

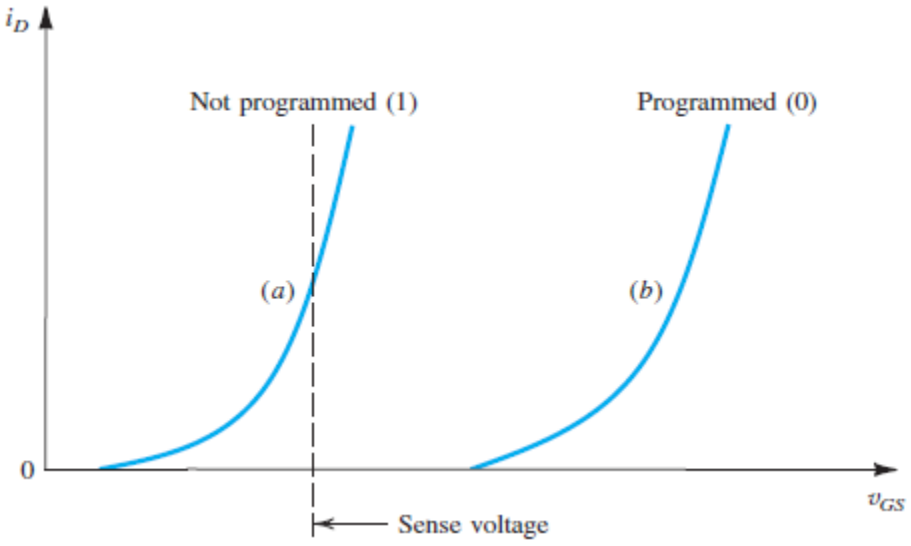

Figure x9.20 Illustrating the shift in the i_D–v_{GS} characteristic of a floating-gate transistor as a result of programming.

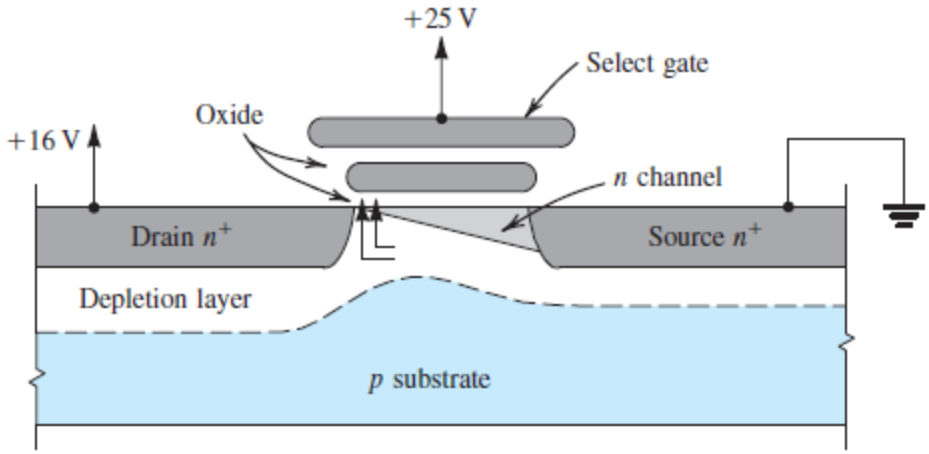

Figure x9.21 The floating-gate transistor during programming.

Fortunately, the process of charging the floating gate is self-limiting. The negative charge that accumulates on the floating gate reduces the strength of the electric field in the oxide to the point that it eventually becomes incapable of accelerating any more of the hot electrons.

Let's now consider the effect of the floating gate's negative charge on the operation of the transistor. The negative charge trapped on the floating gate will cause electrons to be repelled from the surface of the substrate. This implies that to form a channel, the positive voltage that has to be applied to the select gate will have to be greater than that required when the floating gate is not charged. In other words, the threshold voltage V_t of the programmed transistor will be higher than that of the not-programmed device. In fact, programming causes the i_D–v_{GS} characteristic to shift to the curve labeled (*b*) in Fig. x9.20. In this state, known as the *programmed state*, the cell is said to be storing a 0.

Once programmed, the floating-gate device retains its shifted *i*–*v* characteristic (curve *b*) even when the power supply is turned off. In fact, extrapolated experimental results indicate that the device can remain in the programmed state for as long as 100 years!

Reading the content of the stacked-gate cell is easy: A voltage V_{GS} somewhere between the low and high threshold values (see Fig. x9.20) is applied to the selected gate. While a programmed device (one that is storing a 0) will not conduct, a not-programmed device (one that is storing a 1) will conduct heavily.

To return the floating-gate MOSFET to its not-programmed state, the charge stored on the floating gate has to be returned to the substrate. This *erasure* process can be accomplished by illuminating the cell with ultraviolet light of the correct wavelength (2537 A°) for a specified duration. The ultraviolet light imparts sufficient photon energy to the trapped electrons to allow them to overcome the inherent energy barrier, and thus be transported through the oxide, back to the substrate. To allow this erasure process, the EPROM package contains a quartz window. Finally, it should be noted that the device is extremely durable and can be erased and programmed many times.

A more versatile programmable ROM is the electrically erasable PROM (or EEPROM). As the name implies, an EEPROM can be erased and reprogrammed electrically without the need for ultraviolet illumination. EEPROMs utilize a variant of the floating-gate MOSFET. An important class of EEPROMs using a floating-gate variant and implementing block erasure are referred to as **flash memories**. The name "flash" arises because many rows can be erased "in a flash," certainly very rapidly in comparison to the lengthy process of erasing by means of ultraviolet light. Flash memories have virtually replaced the EPROM variety and are currently very popular.

x9.8 CMOS Image Sensors

We conclude this chapter by presenting a very important functional block whose overall structure is very similar to that of a memory array: The CMOS image sensor is the basic image-capturing element in digital cameras (including smartphone cameras).

An image consists of a two-dimensional array of *pixels*, where each pixel indicates light intensity at its location in the array. A CMOS image sensor consists of a two-dimensional array of *pixel circuits*, where each pixel circuit measures light intensity and is usually a few square microns in size. Pixel circuits are accessed through a set of horizontal row-access lines, analogous to the word lines in memory arrays, and intensities are read out through a set of vertical lines, analogous to the bit lines in a memory array. However, here the vertical lines carry analog signals.

A pixel circuit, called an *active pixel sensor* (APS), is shown in Fig. x9.22. Prior to an image capture, transistor Q_P resets the internal node X to a high voltage. The photodiode D is thus reverse biased, and its current I_D is essentially proportional to light intensity. Over the sensing interval T, the discharge of the parasitic capacitor C by current I_D causes a voltage drop ΔV to occur at X. This voltage change is then read out onto the column line by activating a source follower Q_{N1} and a current source (not shown, connected to the column line) and a switch Q_{N2}. The resulting analog signal on the column line is then fed to an analog-to-digital converter (ADC) to provide a digital number corresponding to the light intensity of this pixel. The digital data thus produced can be used for further digital processing of the captured image.

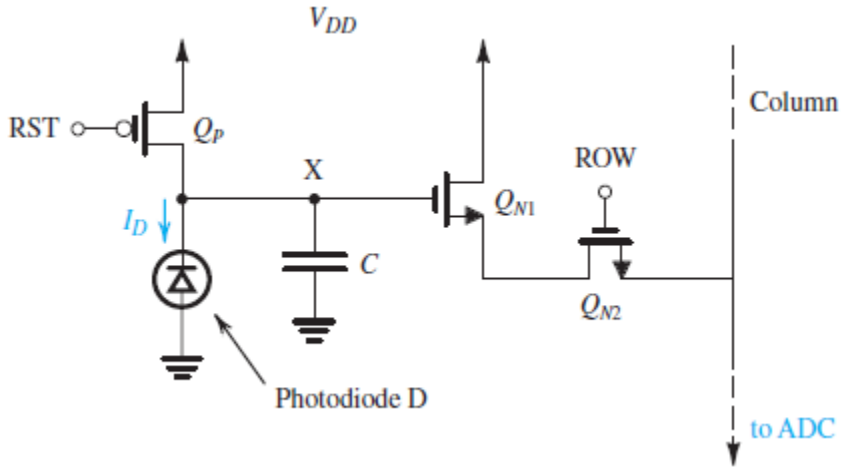

Figure x9.22 Pixel circuit in a CMOS image sensor.

BLINDING FLASH

Since its invention in 1980 by Toshiba, flash memory based on the floating-gate MOS transistor has expanded into every possible field of computing. Because of its nonvolatility, flash memory has become largely responsible for the dominance of mobile digital devices. Increasingly, flash-based solid-state drives (SSDs) are overtaking hard disk drives in enterprise memory systems. SSDs of more than 1 terabyte with no moving parts are becoming available for a few hundred dollars in technologies as small as 20 nm. In many applications, high data rates of up to 12 Gb/s allow total replacement of volatile DRAMs in handheld devices. At the other end of the scale, flash-filled USB drives with gigabyte capacities have effectively replaced the need for DVDs in today's laptops.