

Figure P8.3

[8.10] Consider the BiCMOS inverter shown in Figure 8.29. Suppose that we replace the bottom BJT Q_2 with a large nFET in its place, but leave Q_1 in as the pull-up driver.

Draw the resulting circuit including only the CMOS driver circuitry needed for Q_1 . What is the logic swing for this design?

[8.11] Construct a BiCMOS NOR2 circuit using the circuit in Figure 8.29 as a basis.

[8.12] Design a digital BiCMOS circuit that implements the function

$$f = \overline{a + b \cdot c} \quad (8.19)$$

[8.13] Can you design a BiCMOS circuit that has $V_{OH} = V_{DD}$ and $V_{OL} = 0$ by keeping the basic structure discussed, but modifying the output circuit? Hint: remember that a standard CMOS design has these values.

Advanced Techniques in CMOS Logic Circuits

9

A wide variety of CMOS circuit design styles have been published that are useful in the design of high-speed VLSI networks. All are based on simple logic gates, but operate in distinct ways. Most advanced techniques have been developed to overcome one or more problems that have arisen as VLSI applications have increased over the years. Some are very general, while others are used only for special cases. In this chapter we will unleash a sampling of the modern CMOS circuit techniques that are used in VLSI. This will provide a basis for applications in later chapters.

9.1 Mirror Circuits

Mirror circuits are based on series-parallel logic gates, but are usually faster and have a more uniform layout. The basic idea of a mirror is seen from the XOR truth table in Figure 9.1. Output 0's imply that an nFET chain is conducting to ground, while an output 1 means that a pFET group provides support from the power supply. The important aspect of this observation is that there are equal numbers of input combinations that produce 0's and 1's.

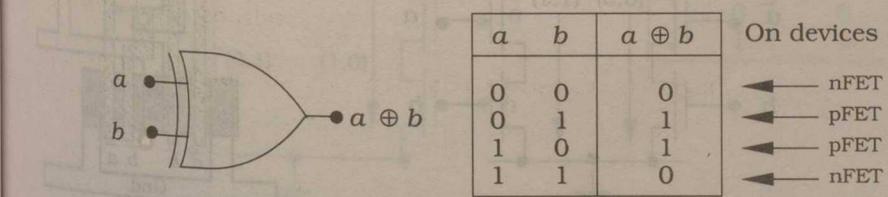


Figure 9.1 XOR function table

A mirror circuit uses the same transistor topology for the nFETs and pFETs. Applying this to the XOR function yields the circuit in Figure 9.2(a). The input combinations are shown for each branch. The "mirror effect" can be understood by placing a mirror along the output line, either up or down. The mirror image seen in the mirror will be the other side of the circuit. A layout for an XOR cell is shown in Figure 9.2(b); the pFETs are larger than the nFETs to compensate for the lower process transconductance (k') values.

The advantages of the mirror circuit are more symmetric layouts and shorter rise and fall times. The latter comment can be understood using the RC switch model in Figure 9.3. Every path between the output and power supply rail consists of two resistors and a parasitic interconnect capacitor. The Elmore time constant is of the form

$$\tau_x = C_{out}(2R_x) + C_x R_x$$

where the subscript x is either n or p , depending upon the branch. Approximating the output voltage as being exponential gives the rise and fall time expressions

$$t_r \approx 2.2\tau_p$$

$$t_f \approx 2.2\tau_n$$

While the form is the same as for an AOI network, the rise time will be smaller because the parasitic capacitance C_p is smaller. This is due to the fact that the mirror circuit has only two pFETs contributing the C_p while an AOI network has four transistors at that node.

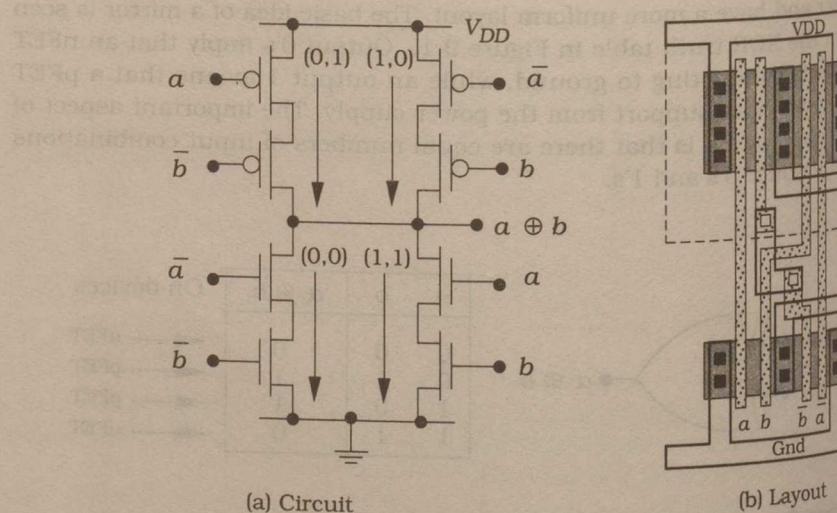


Figure 9.2 XOR mirror circuit

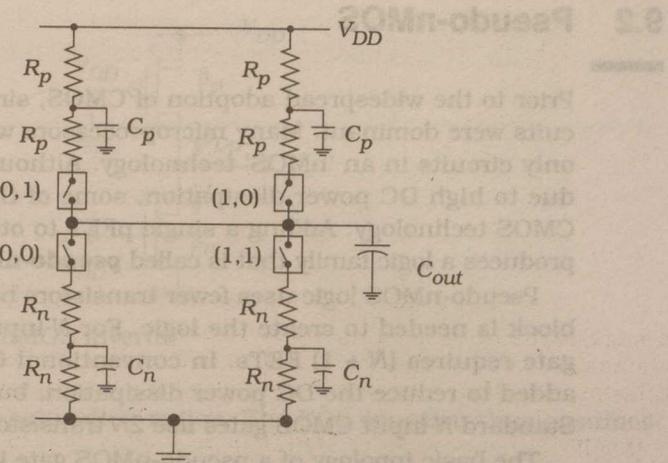


Figure 9.3 Switch model for transient calculations

The idea is easily used to create the XNOR circuit in Figure 9.4. It has the same basic features of the XOR. The relationship

$$a \oplus b = \overline{\overline{a} \cdot b + a \cdot \overline{b}} \quad (9.3)$$

shows that only the inputs a and \overline{a} need to be switched. Other mirror circuits will be introduced later in the text in the context of specific applications such as adder circuits.

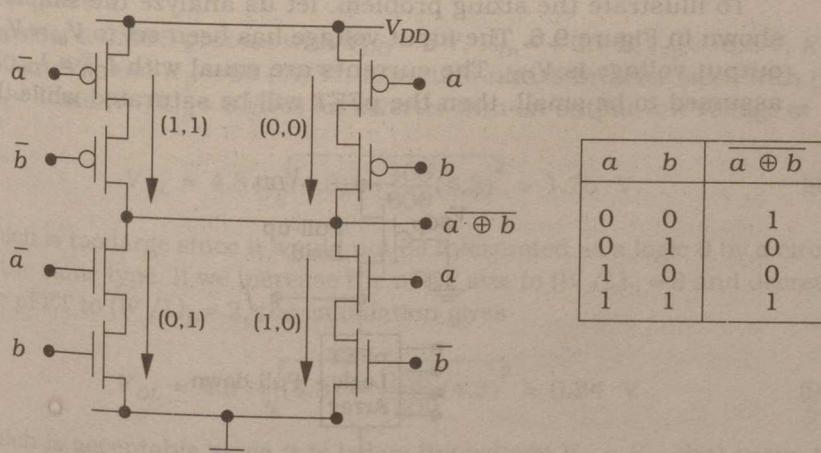


Figure 9.4 Exclusive-NOR (XNOR) mirror circuit

9.2 Pseudo-nMOS

Prior to the widespread adoption of CMOS, single-FET polarity logic circuits were dominant. Many microprocessors were designed using nFET-only circuits in an 'nMOS' technology. Although nMOS was abandoned due to high DC power dissipation, some of the main ideas are used in CMOS technology. Adding a single pFET to otherwise nFET-only circuits produces a logic family that is called **pseudo-nMOS**.

Pseudo-nMOS logic uses fewer transistors because only the nFET logic block is needed to create the logic. For N inputs, a pseudo-nMOS logic gate requires $(N + 1)$ FETs. In conventional CMOS, the pFET group is added to reduce the DC power dissipation, but the logic is superfluous. Standard N -input CMOS gates use $2N$ transistors.

The basic topology of a pseudo-nMOS gate is drawn in Figure 9.5. The single pFET is biased active since the grounded gate gives $V_{SGp} = V_{DD}$. It acts as a **pull-up** device that tries to pull the output f to the power supply voltage V_{DD} . Logic is performed by the nFET array that is designed using the same techniques we have seen. The array acts as a large switch between the output f and ground. If the switch is open, the pFET pulls up the output to a voltage $V_{OH} = V_{DD}$. If the nFET switch is closed, then the array acts as a **pull-down** device that tries to pull f down to ground. However, since the pFET is always biased on, V_{OL} can never achieve the ideal value of 0 V. It is tempting to use pseudo-nMOS circuits to reduce the FET count and area. However, this logic family is more complicated because the relative sizes of the transistors set the numerical value of V_{OL} and care must be taken to insure that V_{OL} is small enough to be an electronic logic 0 voltage.

To illustrate the sizing problem, let us analyze the simple inverter shown in Figure 9.6. The input voltage has been set to $V_{in} = V_{DD}$ so the output voltage is V_{OL} . The currents are equal with $I_{Dn} = I_{Dp}$. If V_{OL} is assumed to be small, then the pFET will be saturated while the nFET

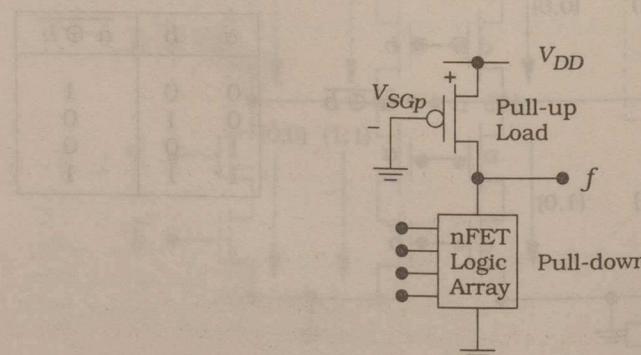


Figure 9.5 General structure of a pseudo-nMOS logic gate

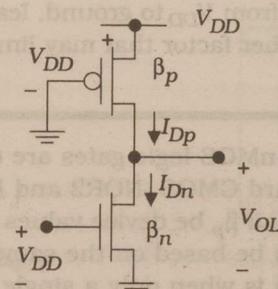


Figure 9.6 Pseudo-nMOS inverter

operates in the non-saturation region. The KCL equation thus assumes the form

$$\frac{\beta_n}{2}[2(V_{DD} - V_{Tn})V_{OL} - V_{OL}^2] = \frac{\beta_p}{2}(V_{DD} - |V_{Tp}|)^2 \quad (9.4)$$

which is a quadratic equation for V_{OL} . Solving gives the physical root

$$V_{OL} = (V_{DD} - V_{Tn}) - \sqrt{(V_{DD} - V_{Tn})^2 - \frac{\beta_p}{\beta_n}(V_{DD} - |V_{Tp}|)^2} \quad (9.5)$$

The value of V_{OL} thus depends on the ratio $(\beta_n/\beta_p) > 1$. Increasing the device ratio decreases the output low voltage. Because of this characteristic, pseudo-nMOS is a type of **ratioed** logic where the relative device sizes set V_{OL} or V_{OH} .

Example 9.1

Consider a CMOS process with $V_{DD} = 5$ V, $V_{Tn} = +0.7$ V, $V_{Tp} = -0.8$ V, $k'_n = 150 \mu\text{A}/\text{V}^2$, and $k'_p = 68 \mu\text{A}/\text{V}^2$. A pseudo-nMOS inverter sized with $(W/L)_n = 4$ and $(W/L)_p = 6$ gives an inverter with an output-low voltage of

$$V_{OL} = 4.3 - \sqrt{(4.3)^2 - \frac{408}{600}(4.2)^2} = 1.75 \text{ V} \quad (9.6)$$

which is too large since it would not be interpreted as a logic 0 by a circuit of the same type. If we increase the nFET size to $(W/L)_n = 8$ and decrease the pFET to $(W/L)_p = 2$, the calculation gives

$$V_{OL} = 4.3 - \sqrt{(4.3)^2 - \frac{136}{1200}(4.2)^2} = 0.24 \text{ V} \quad (9.7)$$

which is acceptable since it is below the voltage $V_{in} = V_{Tn}$ that turns the nFET on. This illustrates that the choice of aspect ratios is critical to this design style. It is important to note that when $V_{in} = V_{DD}$, a current flow

path is established from V_{DD} to ground, leading to a large DC power dissipation. This is another factor that may limit the use of pseudo-nMOS circuits.

General pseudo-nMOS logic gates are designed using the same nFET arrays as in standard CMOS. NOR2 and NAND2 examples are shown in Figure 9.7. Let β_n and β_p be device values for an inverter. The NOR2 gate in Figure 9.7(a) can be based on the same β -values since the worst-case pull-down situation is when only a single nFET is active. This argument can be extended to an N -input NOR gate. The NAND2 gate in Figure 9.7(b) is complicated by the series nFETs. To obtain the same pull-down characteristics of the inverter, the logic transistors must be increased to $2\beta_n$ to provide the same total nFET resistance from the output to ground. This is a general problem with pseudo-nMOS logic gates that require series nFETs.

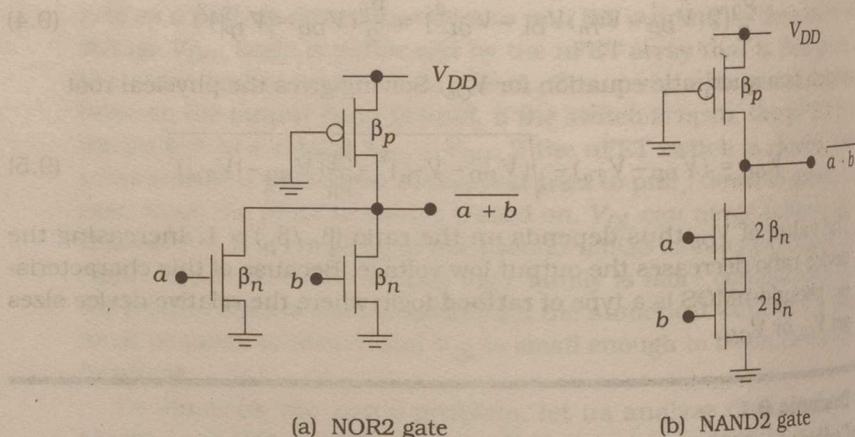


Figure 9.7 Pseudo-nMOS NOR and NAND gates

A basic AOI circuit is shown in Figure 9.8(a) using the same sizing philosophy. The advantage in producing smaller simpler layouts can be seen by the XOR circuit in Figure 9.8(b). Since only a single pFET is used, the interconnect is much simpler. However, the sizes need to be adjusted to insure proper electrical coupling to the next stage. The problems associated with pseudo-nMOS limit its usage to situations where the layout problems are critical, or to some special switching situations where it yields simpler circuitry.

9.3 Tri-State Circuits

A tri-state circuit produces the usual 0 and 1 voltages, but also has a third **high-impedance Z** (or Hi-Z) state that is the same as an open circuit.

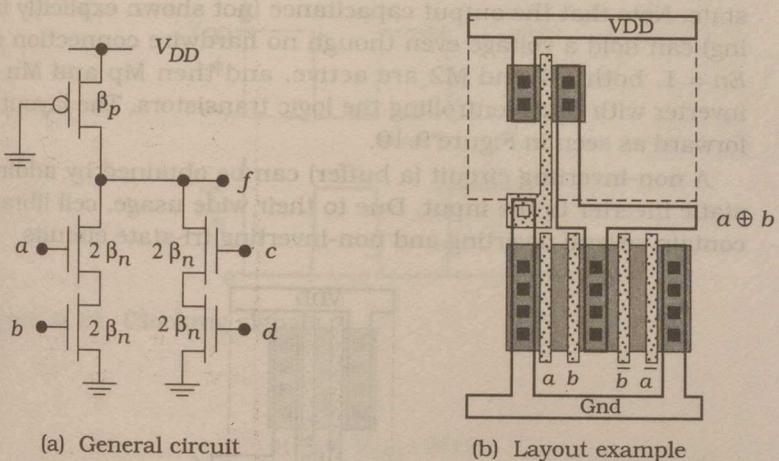


Figure 9.8 AOI gate in pseudo-nMOS logic

cuit. Tri-state circuits are useful for isolating circuits from common bus lines.

The symbol for a tri-state inverter is shown in Figure 9.9(a). The enable signal En controls the operation. With $En = 0$, the output is "tri-stated" which means that $f = Z$. Normal operation occurs with $En = 1$. A CMOS circuit is shown in Figure 9.9(b). FETs M1 and M2 are the tri-state devices. The \overline{En} signal is applied to the pFET M1, while En controls M2. With $En = 0$, both M1 and M2 are off, and the output is isolated from both the power supply and ground. This is the circuit condition of the Hi-Z

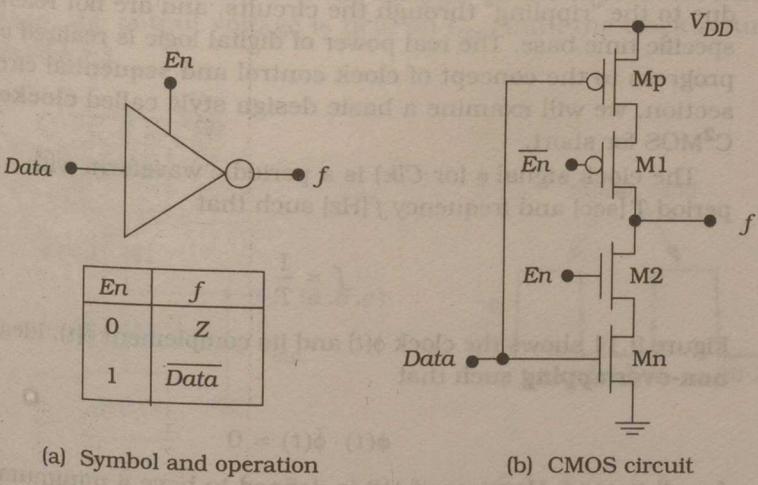


Figure 9.9 Tri-state inverter

state. Note that the output capacitance (not shown explicitly in the drawing) can hold a voltage even though no hardwire connection exists. When $En = 1$, both M1 and M2 are active, and then M_p and M_n act like an inverter with *Data* controlling the logic transistors. The layout is straightforward as seen in Figure 9.10.

A non-inverting circuit (a buffer) can be obtained by adding a regular static inverter to the input. Due to their wide usage, cell libraries usually contain several inverting and non-inverting tri-state circuits.

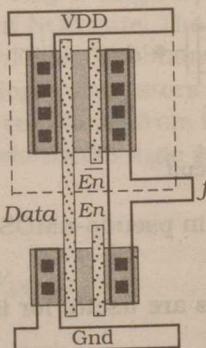


Figure 9.10 Tri-state layout

9.4 Clocked CMOS

Up to this point, all of the circuits we have examined have been completely **static** in nature. The output of a static logic gate is valid so long as the input values are valid and the circuit has stabilized. Logic delays are due to the "rippling" through the circuits, and are not referenced to a specific time base. The real power of digital logic is realized only when we progress to the concept of clock control and sequential circuits. In this section, we will examine a basic design style called **clocked CMOS** or **C²MOS** for short.

The clock signal ϕ (or *Clk*) is a periodic waveform with a well-defined period T [sec] and frequency f [Hz] such that

$$f = \frac{1}{T}$$

Figure 9.11 shows the clock $\phi(t)$ and its complement $\bar{\phi}(t)$. Ideally, these are **non-overlapping** such that

$$\phi(t) \cdot \bar{\phi}(t) = 0$$

for all times t . However, if $\phi(t)$ is defined to have a minimum value of V_{DD} and a maximum of V_{DD} , then

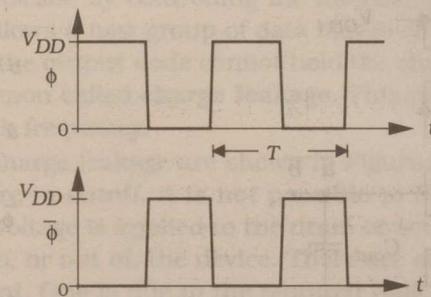


Figure 9.11 Clocking signals

$$\bar{\phi}(t) = V_{DD} - \phi(t) \quad (9.10)$$

so that the clocks overlap slightly during a transition. It may be advantageous to create a set of clocks that are truly non-overlapping for all times.

The general structure of a C²MOS gate is shown in Figure 9.12. It is composed of a static logic circuit with tri-state output network (made up of FETs M1 and M2) that is controlled by ϕ and $\bar{\phi}$. The operation of the circuit can be understood using the clocking waveform shown. When $\phi = 1$, both M1 and M2 are active. Since both the pFET and nFET logic blocks are connected to the output node, the circuit degenerates to a standard static logic gate. The output $f(a, b, c)$ is valid during this time, establishing the voltage V_{out} on the output capacitance C_{out} . When the clock changes to a value of $\phi = 0$, both M1 and M2 are in cutoff, so the output is in a high-impedance state Hi-Z. During this time interval, the FET logic arrays are not connected to the output, so the inputs have no effect. Instead, the output voltage is held on C_{out} until the clock returns to a value of $\phi = 1$.

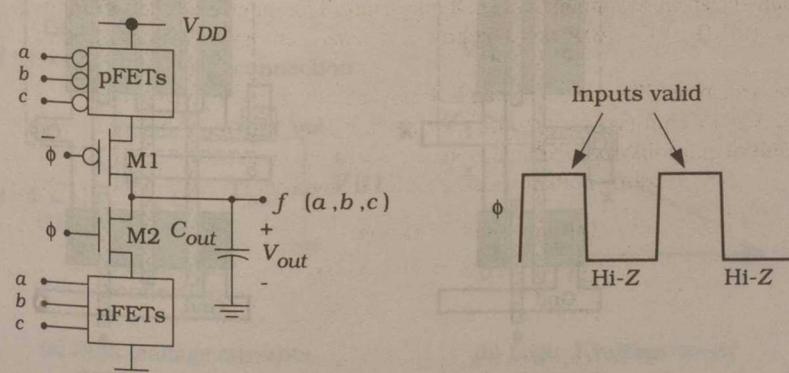


Figure 9.12 Structure of a C²MOS gate

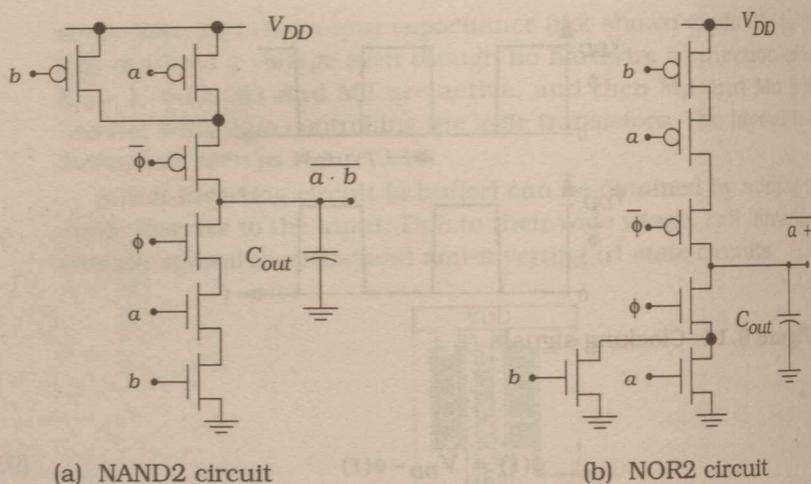
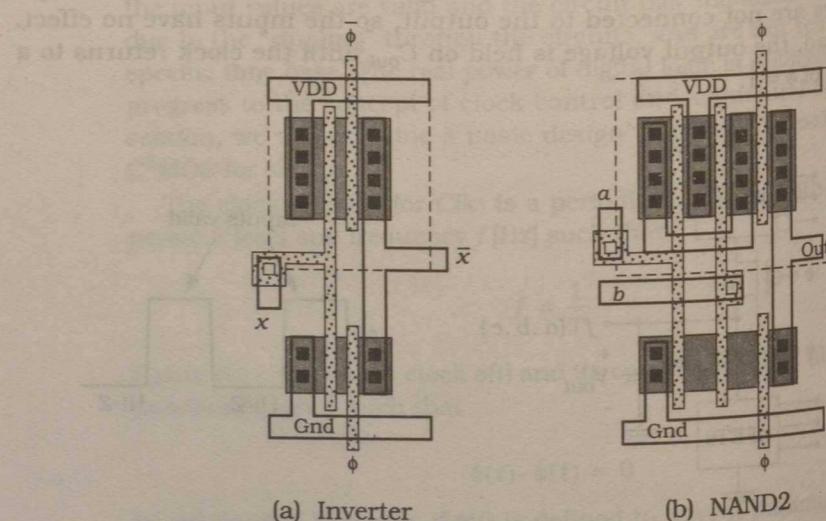


Figure 9.13 Example of clocked-CMOS logic gates

The transistor arrays are designed using the same techniques as for standard logic gates. The circuits for a NAND2 and a NOR2 are shown in Figure 9.13, subdrawings (a) and (b), respectively. Layout is similar to the tri-state circuit with the clock replacing the enable signal. The layouts in Figure 9.14 provide one approach to placing and connecting the transistors. Note that the presence of the series-connected clocking FETs automatically lengthens both the rise and fall times of the circuit.

Clocked CMOS is useful because we can synchronize the data flow

Figure 9.14 Layout examples of C²MOS circuits

through a logic cascade by controlling the internal operation of the gate. Every cycle of ϕ allows a new group of data bits to enter the network. One drawback is that the output node cannot hold the charge on V_{out} very long due to a phenomenon called **charge leakage**. This places a lower limit on the allowable clock frequency.

The basics of charge leakage are shown in Figure 9.15(a). Even though the transistors are in cutoff, it is not possible to block all current flow using a FET. If a voltage is applied to the drain or source, a small **leakage current** flows into, or out of, the device. There are many contributions to the leakage current. One is due to the required bulk connections that are shown in the drawing. The pFET bulk is the nWell region, which is connected to the power supply V_{DD} . Since the pFET source is a p+ region, this creates a pn junction (a diode) that admits a small leakage current i_p flowing on to the node. The nFET has the same problem, with i_n flowing from the output to the p-substrate. Denoting the current off of the capacitor by i_{out} , we may sum the contributions to obtain

$$\begin{aligned} i_{out} &= i_n - i_p \\ &= -C_{out} \frac{dV}{dt} \end{aligned} \quad (9.11)$$

where we have used the capacitor I-V relation in the second line; note the presence of a minus sign to indicate that i_{out} flows out of the positive terminal.

To see the effects of the leakage currents, suppose that we have an initial voltage $V(t=0) = V_1$ stored on the capacitor. If $i_n > i_p$, then $i_{out} = I_L$ is a positive number, indicating current flow off of the capacitor. Rewriting the equation as

$$I_L = -C_{out} \frac{dV}{dt} \quad (9.12)$$

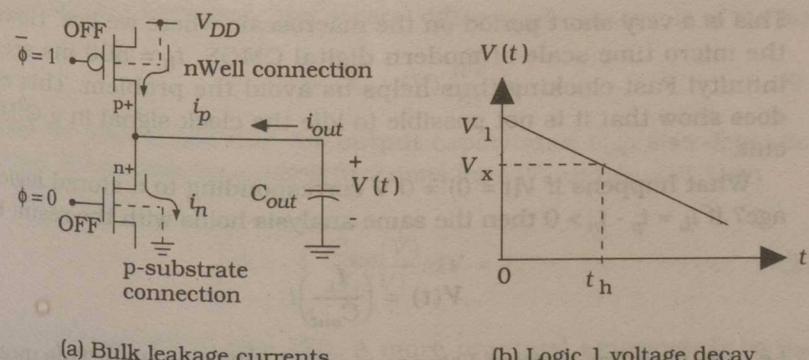


Figure 9.15 Charge leakage problem

we may rearrange it to read

$$\int_{V_1}^{V(t)} dV = - \int_0^t \left(\frac{I_L}{C_{out}} \right) dt \quad (9.13)$$

Assuming that I_L is a constant, the equation may be integrated to yield

$$V(t) = V_1 - \left(\frac{I_L}{C_{out}} \right) t \quad (9.14)$$

which is a linear decay of the voltage with time. This is plotted in Figure 9.15(b). As the voltage decreases, it eventually reaches a minimum logic 1 value that is shown as V_x in the plot. If V falls below this value, it will incorrectly be interpreted as a logic 0 voltage. The **hold time** t_h corresponds to the maximum time that the logic 1 voltage can be stored. By definition, this occurs when

$$V(t_h) = V_1 - \left(\frac{I_L}{C_{out}} \right) t_h = V_x \quad (9.15)$$

Rearranging,

$$t_h = \left(\frac{C_{out}}{I_L} \right) (V_1 - V_x) \quad (9.16)$$

gives the hold time for this case. An order of magnitude estimate of the hold time can be obtained by estimating the capacitance as 50 fF, the leakage current as 0.1 pA, and the voltage change as 1 V. These values give

$$t_h = \left(\frac{50 \times 10^{-15}}{10^{-13}} \right) (1) = 0.5 \text{ sec} \quad (9.17)$$

This is a very short period on the macroscale where we live. However, on the micro time scale of modern digital CMOS, $t_h = 500 \text{ ms}$ seems like infinity! Fast clocking thus helps us avoid the problem. This estimate does show that it is not possible to idle the clock signal in a C²MOS circuit.

What happens if $V(t=0) = 0 \text{ V}$ corresponding to a stored logic 0 voltage? If $I_L = i_p - i_n > 0$ then the same analysis holds with the result that

$$V(t) = \left(\frac{I_L}{C_{out}} \right) t \quad (9.18)$$

i.e., the charging current I_C increases the voltage in time. This means that the logic 0 voltage may drift, so that we again require a minimum clock

frequency.

In submicron devices, the charge leakage problem is exacerbated by the existence of another FET leakage current called the **subthreshold current** I_{sub} . This is a drain-source current that flows even though the gate voltage is less than V_T . A simple estimate for the subthreshold current is

$$I = I_{D0} \left(\frac{W}{L} \right) e^{-(V_{GS}-V_T)/(nV_{th})} \quad (9.19)$$

where I_{D0} varies with V_{DS} , V_{th} is the thermal voltage ($kT/q \approx 26 \text{ mV}$ at 300 K), and n is a parameter that varies with capacitance. A conservative value of I_{D0} is around 10^{-9} A , which noticeably reduces the hold time. With the previous values of capacitance and voltage and $V_{GS} = 0$, the hold time estimate is

$$t_h = \left(\frac{50 \times 10^{-15}}{10^{-9}} \right) (1) = 50 \mu\text{s} \quad (9.20)$$

for leakage through a unity aspect ratio FET. In addition, other contributions to the leakage current originate from the physical structure and the materials used to create the silicon circuit. It would not be unreasonable to find a total charge leakage current of $I_L = 0.1 \mu\text{A} = 10^{-7} \text{ A}$ in a sub-micron device. With this level of leakage, the hold time is reduced to

$$t_h = \left(\frac{50 \times 10^{-15}}{10^{-7}} \right) (1) = 0.5 \mu\text{sec} \quad (9.21)$$

This clearly indicates that charge storage on a capacitive node is a limited-time event, and places important constraints on our logic circuits.

Although we have been approximating the leakage currents as having a constant value for simplicity, a deeper analysis shows that they are voltage-dependent functions. The general differential equation is of the form

$$I_L(V) = -C_{out}(V) \frac{dV}{dt} \quad (9.22)$$

where we have noted that the output capacitance C_{out} also depends on voltage. If we know the explicit functions for $I_L(V)$ and $C_{out}(V)$, then

$$\int_0^t dt = \int_{V_x}^{V(t)} \frac{C_{out}(V)}{I_L(V)} dV = t \quad (9.23)$$

can be integrated to give $V(t)$. A more practical approach is to use a numerical solution. The dependence of the quantities on V results in a non-linear decay, such as the example illustrated in Figure 9.16. The hold time is still defined in the same manner. At the circuit design level, charge

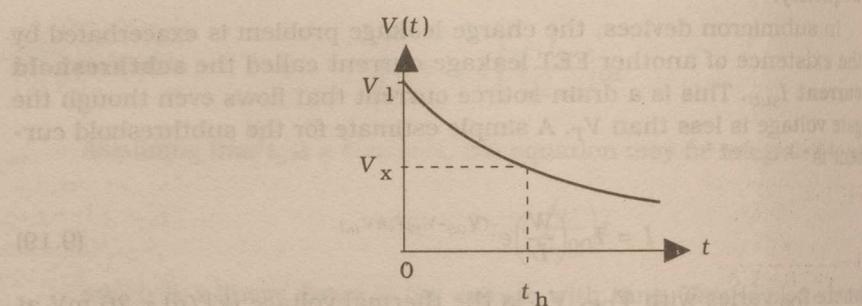


Figure 9.16 General voltage decay

leakage information is usually obtained from circuit simulations.

Charge leakage occurs whenever we attempt to hold charge on a node capacitance using a MOSFET in cutoff. Many of the advanced circuits in the remainder of this chapter have this characteristic, and it is important to remember to check for the problem. Simple SPICE models of MOSFETs do not accurately account for leakage currents. The best results to date are obtained using the BSIM equations.

Motivation for Future Research

While charge leakage is an important problem in dynamic circuits, this discussion highlights the problem of achieving an "open switch" using a MOSFET. As the dimensions shrink, the drain-to-source leakage current increases and the device looks less and less like the idealized switch that was used to design CMOS logic networks. This is one of the most critical problems in digital submicron VLSI. Device researchers are continually looking at the problem. In terms of silicon technology, two main approaches are prevalent. One technique is to reduce the leakage by refining the fabrication process using different materials and variations in the FET structures. Over the years, this has resulted in better devices that have "manageable" leakage current levels that circuit designers must work around.

The other approach is to develop new types of transistors to replace the standard MOSFET. Novel devices with improved characteristics have been proposed and built, and many promising structures have appeared in the literature. However, device research tends to be initially concerned with creating a single transistor, not a high-density VLSI chip. Manufacturing problems often limit the usage of the device in these applications. Another problem is that circuit and logic designers must learn the characteristics of a device before they can develop digital design methodologies. A technique that works with standard MOSFETs probably won't be the best choice for circuits based on transistors that have different I-V characteristics, if it works at all.

Shrinking the size of a MOSFET is often taken as natural evolution of the processing technology. The development of submicron sized FETs had a marked effect on circuit design techniques. Introducing new switching devices would affect all levels of the VLSI design hierarchy, and much research would have to be completed before high-density designs could be implemented. VLSI designers must be continually aware of changes in the field.

9.5 Dynamic CMOS Logic Circuits

A **dynamic logic gate** uses clocking and charge storage properties of MOSFETs to implement logic operations. The clock provides a synchronized data flow which makes the technique useful in designing sequential networks. The characterizing feature of a dynamic logic gate is that the result of a calculation is valid only for a short period of time. While this makes the circuits more difficult to design and use, they require fewer transistors and may be faster than static cascades.

Dynamic circuits are based on the circuit illustrated in Figure 9.17. The clock ϕ drives a complementary pair of transistors M_n and M_p ; these control the operation of the circuit and provide synchronization. Logic is implemented using an nFET array between the output node and ground. The output voltage V_{out} is taken across the output capacitor C_{out} .

The clocking signal ϕ defines two distinct modes of operation during every cycle. When $\phi = 0$ the circuit is in **precharge** with M_p on and M_n off. This establishes a conducting path between V_{DD} and the output, allowing C_{out} to charge to a voltage of $V_{out} = V_{DD}$. M_p is often called the precharge FET. Since the bottom of the nFET logic block is not connected to ground during precharge, the inputs have no effect.

A clock transition to $\phi = 1$ drives the circuit into the **evaluation** mode where M_p is off and M_n is on. The inputs are valid and control the switching in the nFET logic array; M_n is usually called the evaluate transistor. If

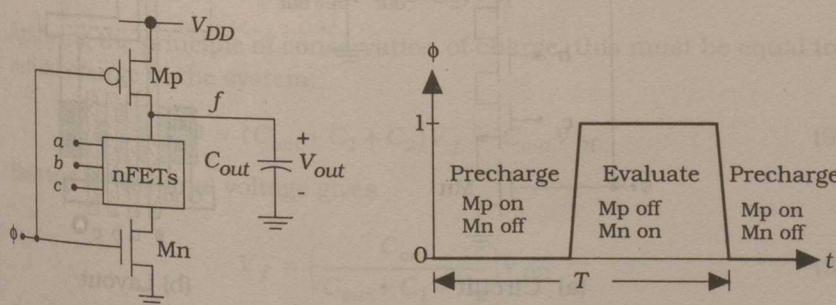


Figure 9.17 Basic dynamic logic gate

the logic block acts like a closed switch, then C_{out} can discharge through the logic array and M_n ; this gives the final result of $V_{out} = 0$ V, corresponding to a logic $f = 0$. If the inputs cause the block to behave like an open switch from top to bottom, the charge on C_{out} is held and $V_{out} = V_{DD}$. Electrically, this is an output of $f = 1$. Charge leakage eventually drops the output to $V_{out} \rightarrow 0$ V, which would be an incorrect logic value. The hold time is determined by the circuitry. In general, this consideration places a minimum frequency stipulation on the clock.

A dynamic NAND3 circuit is shown in Figure 9.18(a). Logic formation is achieved using the three series-connected FETs. The output

$$f = \overline{a \cdot b \cdot c} \quad (9.14)$$

is valid only during the evaluation period when $\phi = 1$. Layout is straightforward as shown by the example in Figure 9.18(b). Since the evaluation nFET M_n is in series with the logic block, C_{out} must discharge through four transistors. Increasing the sizes of the nFETs will reduce the fall time.

As mentioned above, charge leakage reduces the voltages held on the output node when $f = 1$. A detailed analysis of the circuit shows that another problem called **charge sharing** can occur when the clock makes the transition to $\phi \rightarrow 1$. It has the effect of reducing the output voltage even before charge leakage effects become noticeable.

The origin of the charge sharing problem is the parasitic node capacitance C_1 and C_2 between FETs as shown in Figure 9.19. The clock has been set at $\phi = 1$ so that M_p is off, isolating the output node from the power supply. The initial voltage on C_{out} at the start of the evaluation

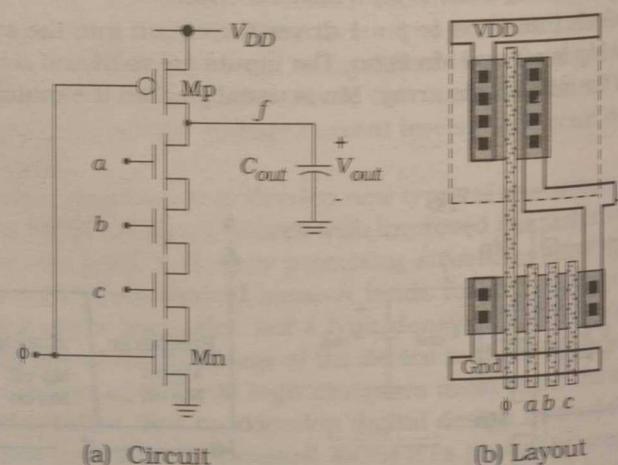


Figure 9.18 Dynamic logic gate example

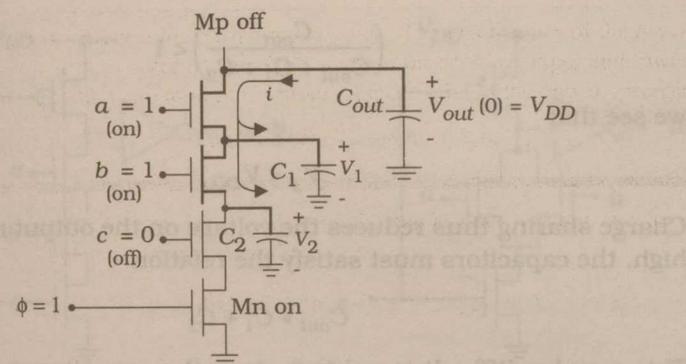


Figure 9.19 Charge sharing circuit

interval is $V_{out} = V_{DD}$ as shown. Assuming that the capacitor voltages V_1 and V_2 are both 0 V at this time, the total charge on the circuit is

$$Q = C_{out} V_{DD} \quad (9.25)$$

The worst-case charge sharing condition for this circuit is when the inputs are at $(a, b, c) = (1, 1, 0)$. With $c = 0$, there is no discharge path to ground, so that the output voltage should remain high. However, since the a - and b -input FETs are on, C_{out} is electrically connected to C_1 and C_2 as indicated by the darkened lines. The current i flows because V_{out} is initially larger than V_1 or V_2 . This corresponds to the transfer of charge from C_{out} to both C_1 and C_2 . Using the relationship $Q = CV$ shows that V_{out} decreases while V_1 and V_2 increase. The current flow ceases when the voltages are equal with a final value

$$V_{out} = V_2 = V_1 = V_f \quad (9.26)$$

The total charge on the circuit is then distributed according to

$$\begin{aligned} Q &= C_{out} V_f + C_1 V_f + C_2 V_f \\ &= (C_{out} + C_1 + C_2) V_f \end{aligned} \quad (9.27)$$

Applying the principle of conservation of charge, this must be equal to the initial charge in the system:

$$Q = (C_{out} + C_1 + C_2) V_f = C_{out} V_{DD} \quad (9.28)$$

Solving for the final voltage gives

$$V_f = \left(\frac{C_{out}}{C_{out} + C_1 + C_2} \right) V_{DD} \quad (9.29)$$

Since

$$\left(\frac{C_{out}}{C_{out} + C_1 + C_2} \right) < 1 \quad (9.30)$$

we see that

$$V_f < V_{DD} \quad (9.31)$$

Charge sharing thus reduces the voltage on the output node. To keep V_{out} high, the capacitors must satisfy the relation

$$C_{out} \gg C_1 + C_2 \quad (9.32)$$

This may be difficult to achieve since the capacitance values are determined by the layout dimensions. After charge sharing takes place, the node is still subject to charge leakage, which continues to drop the voltage with time.

9.5.1 Domino Logic

Domino logic is a CMOS logic style obtained by adding a static inverter to the output of the basic dynamic gate circuit. The resulting structure is shown in Figure 9.20. The precharge and evaluate events still occur, but now it is the capacitor C_X between the dynamic stage and the inverter that is affected. A clock value of $\phi = 0$ defines the precharge. During this time, C_X is charged to a voltage $V_X = V_{DD}$ which forces the output voltage to $V_{out} = 0$ V. Inputs are valid during the evaluation interval when $\phi = 1$. If C_X holds its charge, V_X remains high and $V_{out} = 0$ V indicates a logic 0 output. If C_X discharges, then $V_X \rightarrow 0$ V and $V_{out} \rightarrow V_{DD}$. This corresponds to a logic 1 output.

Domino logic gates are **non-inverting** because of the output inverter. Two examples of this characteristic are shown in Figure 9.21. The AND gate in Figure 9.21(a) is easily understood: if $a = b = 1$, then the internal node discharges to 0 V, forcing the output to a logic 1 (V_{DD}). Similarly, the OR gate in Figure 9.21(b) gives a 1 output if either $a = 1$ or $b = 1$. This

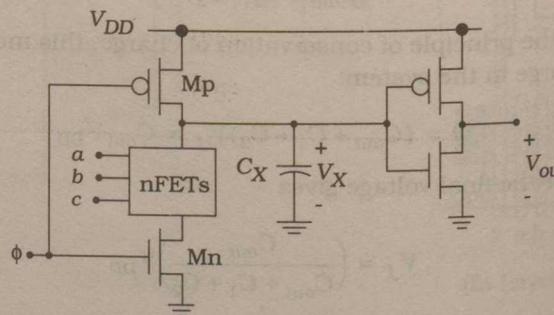


Figure 9.20 Domino logic stage

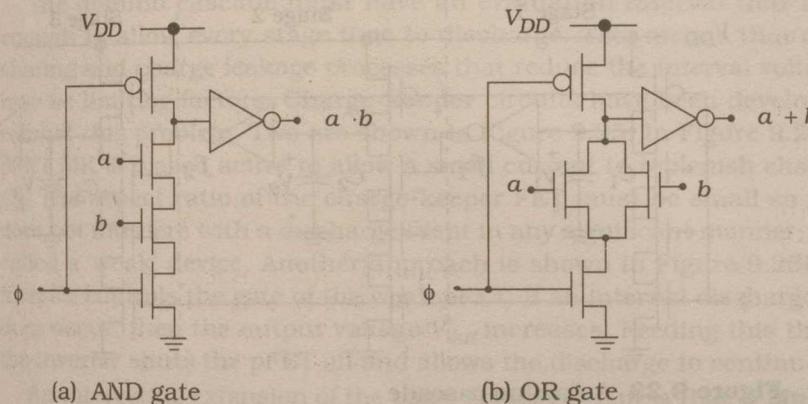


Figure 9.21 Non-inverting domino logic gates

makes logic design using only domino gates somewhat tricky since the NOT operation is required for a complete set of logic operations.¹ While one can add inverters, it is found that this causes the possibility of introducing a hardware glitch into the circuit, and is usually avoided. Inverters are used only at the beginning or the end of a domino chain. An example of a domino layout is shown in Figure 9.22 for an AND3 gate. This is just a dynamic NAND3 circuit cascaded into a static inverter, so the layout preserves the features of general dynamic logic.

Domino logic derives its name from the manner in which a cascade operates. A 3-stage network is shown in Figure 9.23. Every stage is con-

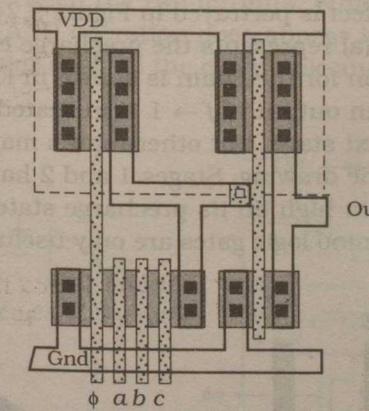


Figure 9.22 Layout for a domino AND gate

¹ A complete set of logic operations is one that is capable of producing any logic combination. Without the NOT operator, functions such as the XOR and XNOR are not possible.

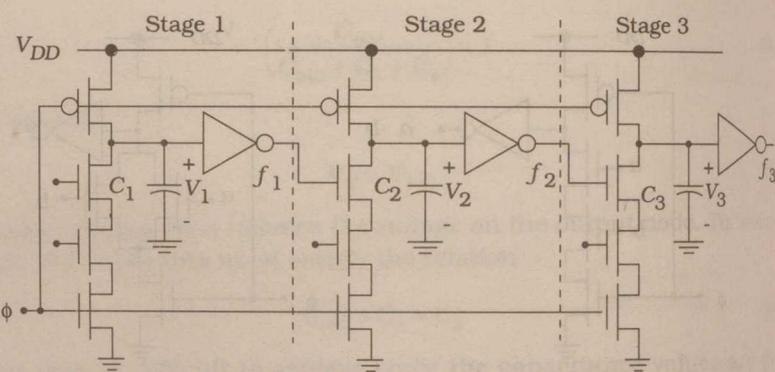


Figure 9.23 A domino cascade

trolled by the same clock phase ϕ . During a precharge event with $\phi = 0$, capacitors C_1 , C_2 , and C_3 are simultaneously charged to V_{DD} . This causes the outputs f_1 , f_2 , and f_3 to all be 0. When $\phi = 1$, the entire chain undergoes evaluation. In a domino cascade, this is like a "domino chain reaction" that must start at the first stage and then propagate stage by stage to the output. To understand this comment, suppose that we monitor the second stage output f_2 and see it switch from its precharge value $f_2 = 0$ to $f_2 = 1$ during the evaluation interval. The only way this could have happened is if C_2 discharged, but this requires that $f_1 = 1$ to turn on the nFET in the discharge chain. Applying the same logic to the first stage, f_1 can switch to 1 only if C_1 has discharged. Extending this argument, we see that $f_3 \rightarrow 1$ occurs only if both Stage 1 and Stage 2 have made the same transition.

The domino effect is portrayed in Figure 9.24 to help visualize the process. Figure 9.24(a) represents the precharge event by dominos standing on end. Evaluation for the chain is shown in Figure 9.24(b). A discharge event that gives an output of $f \rightarrow 1$ is indicated by a falling domino. This can topple the next stage, but other inputs may keep the discharge from taking place. In the drawing, Stages 1 and 2 have undergone a discharge, but Stage 3 remains high (in its precharge state). Note that the operation indicates that domino logic gates are only useful in cascades.

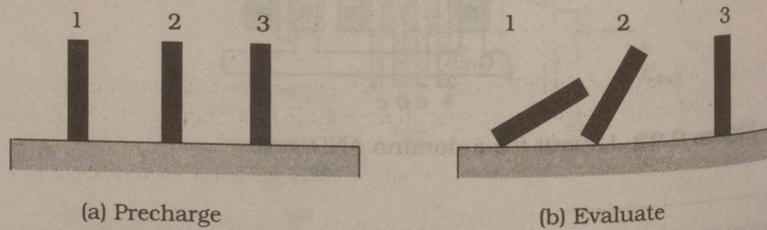


Figure 9.24 Visualization of the domino effect

The domino cascade must have an evaluation interval that is long enough to allow every stage time to discharge. This means that charge sharing and charge leakage processes that reduce the interval voltage V_X may be limiting factors. **Charge-keeper** circuits have been developed to combat this problem. Two are shown in Figure 9.25. In Figure 9.25(a), a pFET MK is biased active to allow a small current to replenish charge on C_X . The aspect ratio of the charge-keeper FET must be small so that it does not interfere with a discharge event in any significant manner; this is called a 'weak' device. Another approach is shown in Figure 9.25(b). An inverter controls the gate of the weak pFET. If an internal discharge of C_X does occur, then the output voltage V_{out} increases. Feeding this through the inverter shuts the pFET off and allows the discharge to continue.

An interesting extension of the basic domino circuit is that of **Multiple-Output Domino Logic (MODL)**. This type of circuit allows two or more outputs from a single logic gate, making it quite unique. The structure of a 2-output MODL stage is shown in Figure 9.26. The logic array has been split into two separate blocks denoted as F and G , which creates an additional output node. Adding an inverter and a precharge transistor results in the two outputs

$$\begin{aligned} f_1 &= G \\ f_2 &= F \cdot G \end{aligned} \quad (9.33)$$

This is easily understood by studying the logic network. If the G-logic block acts like a closed switch, then it produces an output of $f_1 = G$. If this occurs, then it is possible for the second logic block F to induce a discharge by also acting as a closed switch. This dependence produces the ANDing relation between the two outputs. While this is quite restrictive, the nesting of the AND operation does appear in several important computational algorithms such as the carry look-ahead adder.

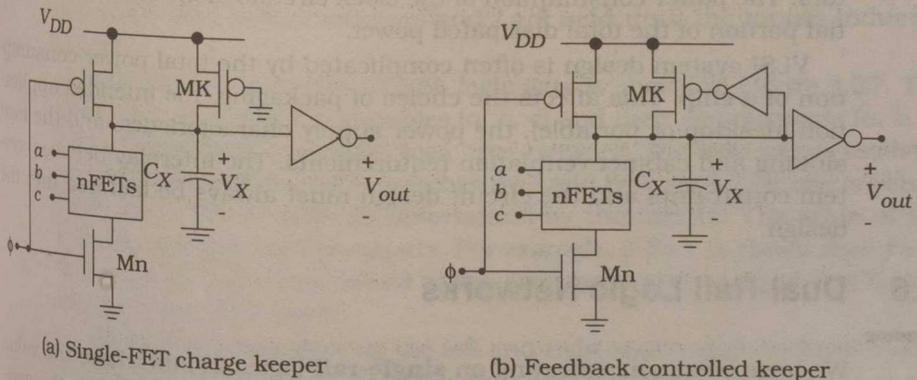


Figure 9.25 Charge-keeper circuits

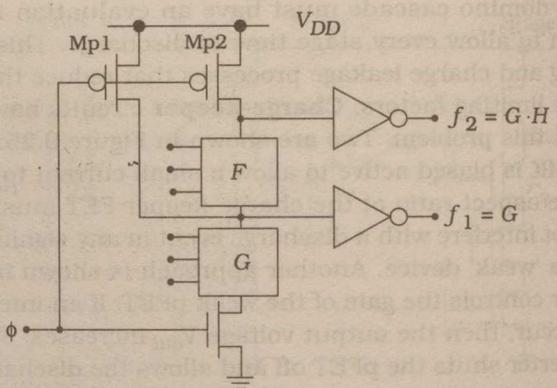


Figure 9.26 Structure of a MODL circuit

9.5.2 Power Dissipation of Dynamic Logic Circuits

CMOS dynamic logic circuits can be designed to provide very fast switching with modest real estate consumption. They have been successfully used in several well-known chips and are the basis of DRAMs and other important computer components. Unfortunately, they can be quite power hungry which may limit their usage.

In a dynamic circuit, the clock ϕ defines the precharge and evaluate operations in every cycle. Since charge cannot be held on a capacitive node, every precharge cycle will pull current from the voltage source, adding to the overall power dissipation of the circuit. The clock circuits themselves require dynamic power to drive the FETs. In the standard configuration, every stage presents a capacitance of

$$C_L = C_{Gp} + C_{Gn} \quad (9.34)$$

to the clock drivers corresponding to the precharge and evaluate transistors. The power consumption of the clock circuits alone can be a substantial portion of the total dissipated power.

VLSI system design is often complicated by the total power consumption of a chip. This affects the choice of packaging, the intended application (desktop or portable), the power supply characteristics, and the heat sinking and cabinet ventilation requirements. The interplay between system constraints and the circuit design must always be factored into the design.

9.6 Dual-Rail Logic Networks

We have been concentrating on **single-rail** logic circuits where the value of a variable is either a 0 or a 1 only. In **dual-rail** networks, both the variable x and its complement \bar{x} are used to form the difference

$$f_x = (x - \bar{x}) \quad (9.35)$$

Using the quantity f_x provides an increase in the switching speed. This can be seen by calculating the time derivative as

$$\frac{df_x}{dt} = \left(\frac{dx}{dt} - \frac{d\bar{x}}{dt} \right) \quad (9.36)$$

and noting that

$$\frac{d\bar{x}}{dt} \approx -\frac{dx}{dt} \quad (9.37)$$

since x increases while \bar{x} decreases, and vice versa. Thus

$$\frac{df_x}{dt} \approx 2 \left| \frac{dx}{dt} \right| \quad (9.38)$$

so that the rate of change of f_x is approximately twice that of a single variable. Translated into logic terms, this means that the switching speed is almost twice as fast as can be obtained in a single-rail circuit.

The complicating factor in dual-rail circuits is the increase in circuit complexity and wiring overhead. Every input and output is now a doublet consisting of the variable and its complement. The circuits are correspondingly more complicated, and can be tricky to deal with. However, the speed advantage makes them worth studying. Some even provide structured and compact layout schemes.

9.6.1 CVSL

Most dual-rail CMOS circuits are loosely based around **differential cascode voltage switch logic**, which goes under the acronyms **DCVS logic** or **differential CVSL**; we will adopt the latter one here. CVSL provides for dual-rail logic gates that have latching characteristics built into the circuit itself. The output results f and \bar{f} are held until the inputs induce a change.

The basic structure of a CVSL logic gate is shown in Figure 9.27. The input set consists of the variables (a, b, c) and their complements $(\bar{a}, \bar{b}, \bar{c})$ that are routed into an nFET 'logic tree' network. The logic tree is modeled as a pair of complementary switches Sw1 and Sw2 such that one is closed while the other is open as determined by the inputs. The state of the switches establishes the outputs. For example, if Sw1 is closed then $f = 0$. The opposite side (\bar{f}) is forced to the complementary state ($\bar{f} = 1$) by the action of the pFET latch.

The latch is controlled by the left and right source-gate voltages V_l and V_r shown in the drawing. Suppose that Sw2 is closed, forcing $\bar{f} = 0$ on the right side. In this case,