Chapter

14

Dynamic Logic Gates

Dynamic or clocked logic gates are used to decrease complexity, increase speed, and lower power dissipation. The basic idea behind dynamic logic is to use the capacitive input of the MOSFET to store a charge and thus remember a logic level for use later. Before we start looking into the design of dynamic logic gates, let's discuss leakage current and the design of clock circuits.

14.1 Fundamentals of Dynamic Logic

Consider the NMOS pass gate (PG) driving an inverter, as shown in Fig. 14.1. If we clock the gate of the PG high, the logic level on the input, point A, will be passed to the input of the inverter, point B. If this logic level is a "0," the input of the inverter will be forced to ground, while a logic "1" will force the input of the inverter to $VDD - V_{THN}$. When the clock signal goes low, the PG shuts off and the input to the inverter "remembers" the logic level. In other words, when the PG turns on, the input capacitance of the inverter is either charged to $VDD - V_{THN}$ or discharged to ground through the PG. As long as this charge is present on the parasitic input capacitance of the inverter, the logic value is remembered. What we are concerned with at this point are the leakage mechanisms which can leak the stored charge off the node. A node, such as the one labeled B in Fig. 14.1, is called a dynamic node or a storage node. Note that this node is a high-impedance node and is easily susceptible to noise (see Ex. 3.5).

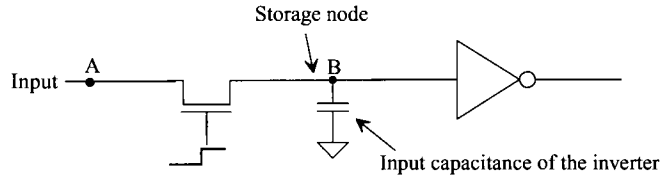


Figure 14.1 Example of a dynamic circuit and associated storage capacitance.

14.1.1 Charge Leakage

One of the important concerns with designing dynamic circuits is the MOSFET's off current, see Sec. 6.4.2. In Fig. 14.2 we show the simulated drain current of a 10/1 MOSFET in the 50 nm process plotted against gate-source voltage. The off current, with $V_{GS} = 0$, is taken from the plot as

$$\log I_D = -8.45 \rightarrow I_D = 3.55\ nA = I_{off} \cdot W \cdot scale \qquad (14.1)$$

noting that the worst case off current occurs when the minimum channel length is used ($L = 1$). The off current can be estimated as

$$I_{off} = 7.1\ nA/\mu m \qquad (14.2)$$

The off current is made up of leakage through the source/drain implant to substrate (the formed diode and the associated saturation currents), leakage around the edges of the poly at the edge of the field oxide, and by the source-to-drain leakage currents.

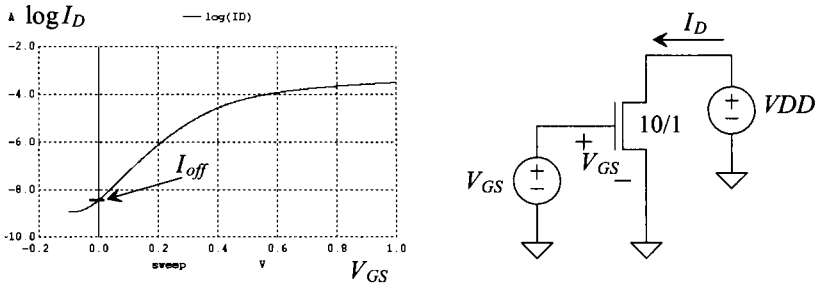


Figure 14.2 Drain current of an NMOS device plotted from weak to strong inversion. See Fig. 6.16. PMOS netlist is found at cmosedu.com.

The rate at which the storage node seen in Fig. 14.1 discharges is given by

$$\frac{dV}{dt} = \frac{I_{off} \cdot W \cdot scale}{C_{node}} \qquad (14.3)$$

The node capacitance, C_{node}, is the sum of the input capacitance of the inverter, the capacitance to ground of the metal or poly line connecting the inverter to the pass transistor, and the capacitance of the drain implant to substrate (the depletion capacitance). For practical applications, we assume that

$$C_{node} \approx C_{in} \text{ of the inverter} \qquad (14.4)$$

Example 14.1

Estimate the discharge rate of the 50 fF capacitor shown Fig. 14.3.

Using Eqs. (14.1) and (14.3), we can write the capacitor's discharge rate as

$$\frac{dV}{dt} = \frac{3.55\ nA}{50\ fF} = 71\ mV/\mu s$$

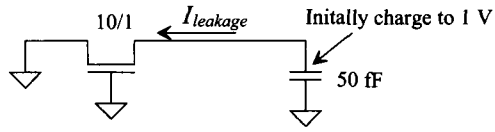


Figure 14.3 Circuit used in Ex. 14.1.

For the capacitor to discharge to ground (from an initial voltage of 1 V) takes, roughly, $(1\ V)/(71\ mV/\mu s) = 14\ \mu s$. Figure 14.4 shows the simulation results. Note how the time it takes the output voltage to discharge to ground is twice as long as what we calculated or roughly 30 μs. This is due to the fact that as the voltage between the drain and source of the MOSFET decreases so does the off current. The reduction in the off current causes the capacitor to discharge slower. To keep the voltage across the capacitor from discharging too much we might try holding one side of the MOSFET switch at *VDD*/2. Consider the following. ■

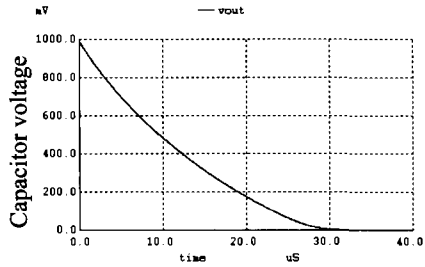


Figure 14.4 Time it takes the capacitor to discharge due to the off current of the MOSFET in Fig. 14.3.

Example 14.2
Repeat Ex. 14.1 with the input tied to *VDD*/2 instead of ground.

The schematic and simulation results are seen in Fig. 14.5. The leakage current is minimized by lowering the V_{DS} of the MOSFET. This is a common technique to

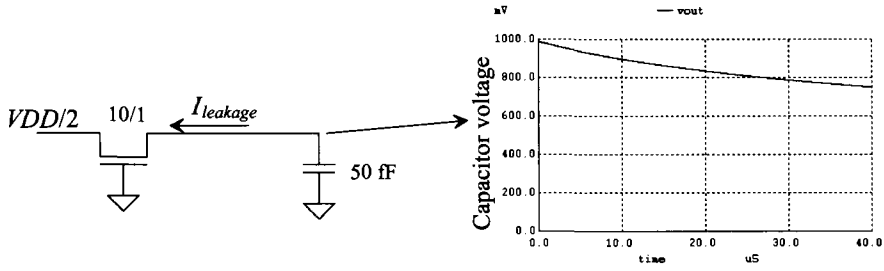


Figure 14.5 Circuit used in Ex. 14.2.

reduce leakage currents in DRAM (see Fig. 16.7 where the bitlines are equilibrated to *VDD*/2). Note that when the voltage across the capacitor (the storage node) is zero and the PG is off the leakage through the MOSFET causes the storage node to charge upwards (see Problem 14.2). ■

Example 14.3

Figure 14.6 shows a dynamic level-sensitive latch. Estimate the maximum time PG can be off before data is lost on the charge storage node. Compare the estimate to SPICE. Use the 50 nm process with 20/1 PMOS devices and 10/1 NMOS devices.

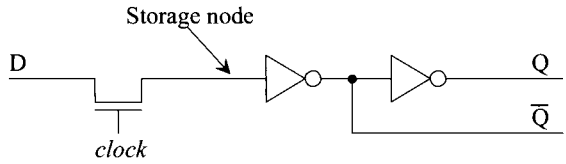


Figure 14.6 A dynamic level-sensitive latch.

The input capacitance of the inverter, using Table 10.2, is estimated as

$$C_{in} = \frac{3}{2} \cdot (1.25 + 0.625)\,fF = 2.8\,fF$$

Using Eq. (14.3), we can estimate the rate the storage node decays as

$$\frac{dV}{dt} = \frac{3.55\,nA}{2.8\,fF} = 1.27\ V/\mu s$$

If we want, at most, 100 mV of droop in the storage node's voltage, then we would want to ensure that the PG is clocked, at the minimum, every 100 ns. Figure 14.7 shows the simulation results. Compared to our hand calculations, the discharge rate is considerably faster. This is due to our calculation of the input capacitance. The values used for C_{in} were the switching input capacitance (see Figs. 10.2, 10.7, and 10.8), not the static input capacitance (see Fig. 6.4). ■

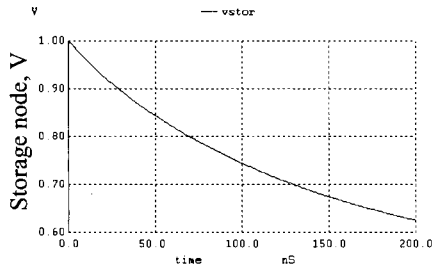


Figure 14.7 How the voltage on the storage node varies in the dynamic latch circuit seen in Fig. 14.6.

14.1.2 Simulating Dynamic Circuits

Because of the extremely small leakage currents involved, simulating dynamic circuits can be challenging. To begin, when SPICE simulates any circuit, it puts a resistor with a conductance value given by the parameter GMIN across every pn junction and MOSFET drain-to-source. The default value of GMIN is 10^{-12} mhos or a 1 TΩ resistor. A charge storage node at a potential of 1 V has a leakage current, due to GMIN, of 1 pA. The value of GMIN can be set, and increased, using the .OPTIONS command, at the cost of a longer or more difficult convergence time, to a smaller value, say 10^{-15}. Some SPICE simulators are also capable of adding a resistor at every node to ground called RSHUNT (to help with convergence). If RSHUNT is 10^9 and a node voltage is 1 V, then a current of 1 nA flows through the added RSHUNT resistor.

In practice, we can use the default values of SPICE when simulating dynamic circuits. The SPICE leakage paths (using the default values of GMIN and RSHUNT) have little effect on the estimates for the discharge times of storage nodes when simulating dynamic circuits. For example, the leakage current, for VDD = 1 V, due to the default value of GMIN is given by

$$I_{leakage} \approx 1\ pA = VDD \cdot GMIN \tag{14.5}$$

and

$$\frac{dV}{dt} = \frac{1\ pA}{C_{node}} = \frac{VDD \cdot GMIN}{C_{node}} \tag{14.6}$$

If C_{node} = 5 fF, then $dV/dt = 200 = 200\ \mu V/\mu s$. It takes approximately 5 ms for the voltage on the charge storage node to drop to ground. In all practical simulations, the leakage through a modern nanometer MOSFET is much more significant than the leakage through GMIN.

14.1.3 Nonoverlapping Clock Generation

Consider the string of PGs/inverters shown in Fig. 14.8. This circuit is called a dynamic shift register. When ϕ_1 goes high, the first and third stages of the register are enabled. Data is passed from the input to point A0 and from point A1 to A2. If ϕ_2 is low while ϕ_1 is high, the data cannot pass from A0 to A1 and from A2 to A3. If ϕ_1 goes low and ϕ_2 goes high, data is passed from A0 to A1 and from A2 to A3. If both ϕ_1 and ϕ_2 are high at the same time, the input of the shift register and the output are connected together, which is not desirable in a shift register application. The purpose of the inverter between PGs is to restore logic levels, since the NMOS PG passes a high with a threshold voltage drop. Two inverters would be used to eliminate the logic inversion between stages. *The clocks used in this dynamic circuit must be nonoverlapping, or logically*

$$\phi_1 \cdot \phi_2 = 0 \tag{14.7}$$

There should be a period of dead time between transitions of the clock signals, labeled Δ in Fig. 14.8. The rise and fall times of the clock signals should not occur at the same time.

Since the design and layout of the dynamic shift register is straightforward, let's concentrate on the generation of clock signals, ϕ_1 and ϕ_2. Note that a simple logic inversion will not generate nonoverlapping clock signals. Consider the schematic of the nonoverlapping clock generator shown in Fig. 14.9. This circuit takes a clock signal and

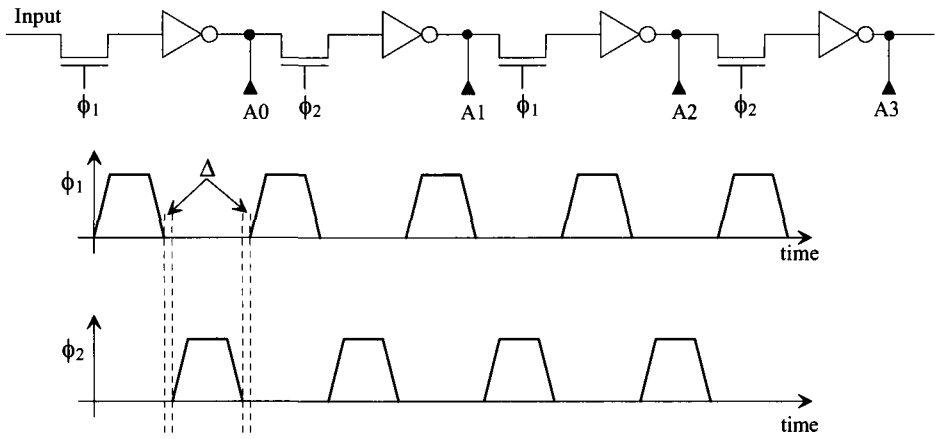


Figure 14.8 Dynamic shift register with associated nonoverlapping clock signals.

generates a two-phase nonoverlapping clock. The amount of separation is set by the delay through the NAND gate and the two inverters on the NAND gate output. Consider the input clock going high. This forces ϕ_1 high and ϕ_2 low. When the input clock goes low, ϕ_1 goes low. After ϕ_1 goes low, ϕ_2 can go high. When driving long transmission lines, like a wire implemented using polysilicon, where the rise time of the signals can be significant, a large number of inverters may be needed. Line drivers, a string of inverters used to drive a large capacitance, can be used as part of the delay in the nonoverlapping clock generation circuit.

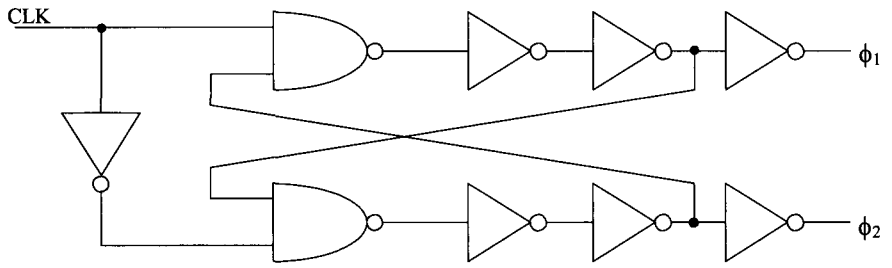


Figure 14.9 Nonoverlapping clock generation circuit.

14.1.4 CMOS TG in Dynamic Circuits

The CMOS TG used as a switch to charge or discharge the node capacitance of the charge storage node is shown in Fig. 14.10. Because understanding the charging and discharging of the input capacitance of the inverter follows many of the same analysis and discussions of Ch. 13, we concentrate here on the charge leakage from the TG.

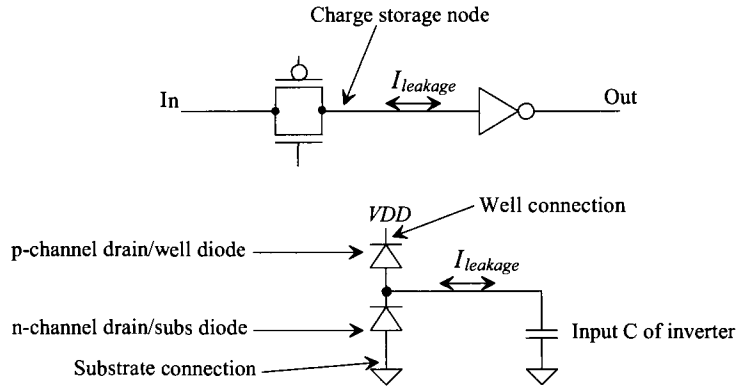


Figure 14.10 CMOS TG used in dynamic logic.

The leakage of charge off of or onto the input capacitance of the inverter in Fig. 14.10 can be attributed to the drain-well diode of the PMOS device and the drain-substrate diode of the NMOS device used in the TG. If these leakage currents were equal, then the leakage of charge off of the storage node would be zero. Notice that unlike the NMOS PG, using a TG can result in the charge storage node leaking to *VDD* or ground, depending on the size of the drain areas and the leakage currents in each device.

14.2 Clocked CMOS Logic

In this section we provide a brief overview of dynamic, or clocked, CMOS logic design.

Clocked CMOS Latch

Figure 14.11 shows the schematic of a clocked CMOS latch. When ϕ_1 is a low, M2 and M3 are on. The master stage simply behaves like an inverter. The D input drives, through the enabled master stage, the node N1. During this time, both M6 and M7 are off. The latch's output, Q, is a charge storage node. When ϕ_1 goes high, M2 and M3 shut off and N1 is the charge storage node. M6 and M7 are on, and the Q output is actively driven either high or low.

An Important Note

Notice, in Fig. 14.11, that if the node N1 starts to move away from either *VDD* or ground, when ϕ_1 is a high, that the slave stage can move towards its switching point. This can cause a significant current to flow from *VDD* to ground in the slave stage. Another example of where this problem can occur is seen in Fig. 14.6. If the storage node starts to wander, the inverter's input can move towards its switching point voltage. As seen in Fig. 11.4, again, a significant current can flow.

Note that the storage node's voltage can wander because of the MOSFET's off current (Fig. 14.2), a gate tunnel current leaking into the node (Fig. 16.67), or because of capacitive coupling from a noisy node (as discussed in Ex. 3.5). The issues of gate tunneling current and the reduction in parasitic capacitances present challenges when implementing dynamic logic in nanometer CMOS technologies.

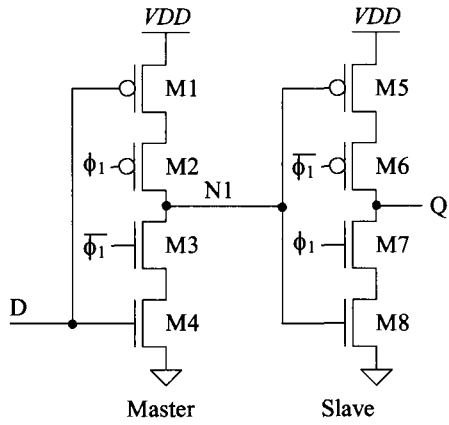


Figure 14.11 A clocked CMOS latch. The clock signals can be generated with an RS latch so that the edges occur essentially at the same moment in time.

PE Logic

This section presents precharge-evaluate logic, or PE logic. Consider the three-input NAND gate shown in Fig. 14.12. The operation of this gate relies on a single clock input. When ϕ_1 is low, the output node capacitance is charged to *VDD* through M5. During the evaluate phase, ϕ_1 is high, M1 is on, and if A0, A1, and A2 are high, the output is pulled low. The logic output is available only when ϕ_1 is high. The output is a logic one when ϕ_1 is low. One disadvantage of PE logic is that the gate logic output is available part of the time and not all of the time as in the static gates.

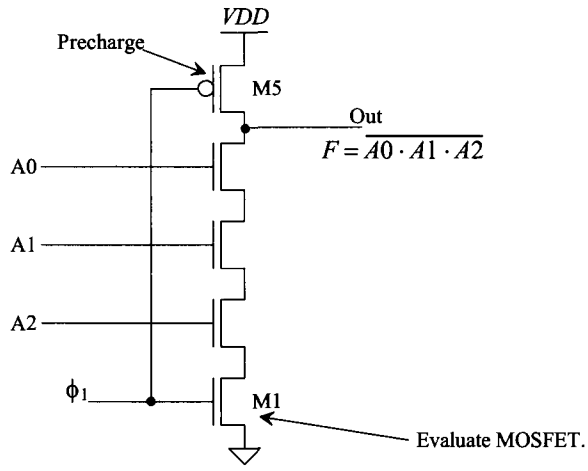


Figure 14.12 Precharge-evaluate three-input NAND gate.

Several important characteristics of the PE gate should be pointed out. The input capacitance of the PE gate is less than that of the static gate. Each input is connected to a single MOSFET where the static gate inputs are tied to two MOSFETs. Potentially the PE gate operates faster and dissipates less power.

The sizes of the MOSFETs used in a PE gate does not need adjusting for symmetrical switching point voltage. The absence of complementary devices and the fact that the output is pulled high during each half cycle makes the gate V_{SP} meaningless. However, we may need to size the devices to attain a certain speed for a given load capacitance. If the sizes of all NMOS transistors used in Fig. 14.12 are equal, then the t_{PHL} is approximately $4R_nC_{node}$ and the t_{PLH} is R_pC_{node}, where C_{node} is the total capacitance on the output node. This may include the interconnecting capacitance and the input capacitance of the next stage. Here we have neglected both the transmission line effects through a series connection of MOSFETs and the intrinsic switching speeds. A more complex logic function, $F = \overline{A0 + A1 \cdot A2 + A3 \cdot A4}$, implemented in PE logic is shown in Fig. 14.13.

Domino Logic

Consider the cascade of PE gates shown in Fig. 14.14. During the precharge phase of the clock, the output of each PE gate is a logic high. This high-level output is connected to the input of the next PE gate. Suppose that the logic out of the first PE gate during the evaluate phase is a low. This output will turn off any MOSFETs in the second PE gate. However, during the precharge phase, those same MOSFETs in the second PE gate will be turned on. The delay between the clock pulse going high and the valid output of the first gate will cause the second gates, output to glitch or show an invalid logic output. If we can hold the output voltage of the PE gate low instead of high, we can eliminate this race condition. Upon adding an inverter to the PE gate (Fig. 14.15), the condition for glitch-free operation is met. The PE gate with the addition of an inverter is called Domino logic. The name *Domino* comes from the fact that a gate in a series of Domino logic gates cannot change output states until the previous gate changes states. The change in output of the gates occurs similar to a series of falling dominoes. The inverter used in the Domino gate has the added advantage that it can be sized to drive large capacitive loads.

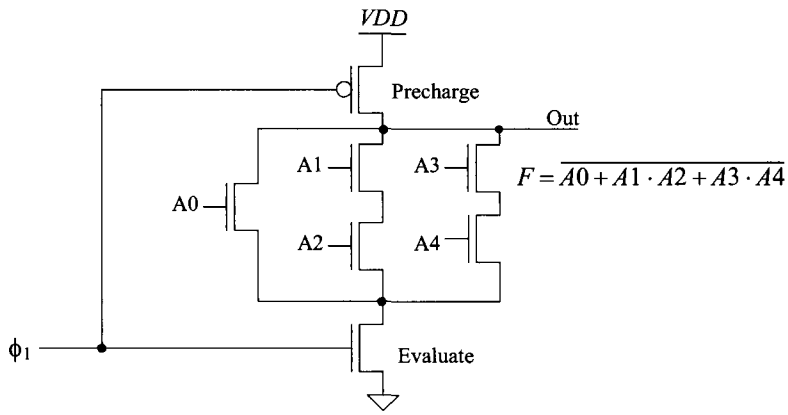


Figure 14.13 A complex PE gate.

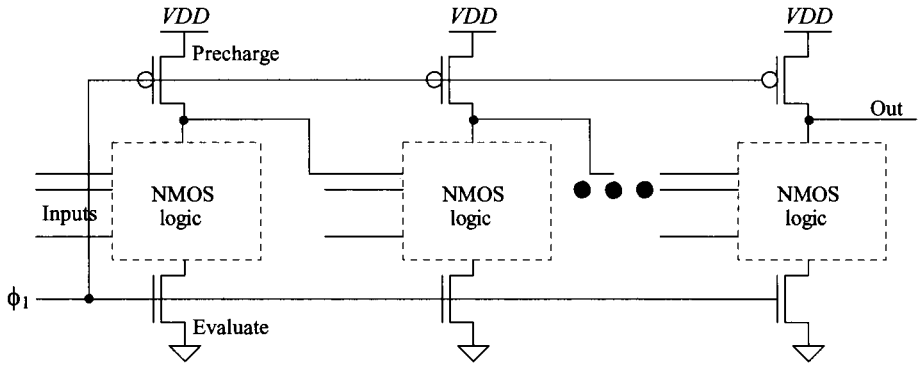


Figure 14.14 Problems with a cascade of PE gates.

One problem does exist with this scheme, however. Referring to Fig. 14.15, note that during the precharge phase, node A is charged to *VDD*. If the NMOS logic results in a logic high on node A during the evaluate phase, then that node is at a high impedance with no direct path to *VDD* or ground. The result is charge leakage off of node A when the PE output is a logic high. The circuit of Fig. 14.16a eliminates this problem. A "keeper" PMOS device is added to help keep node A at *VDD* when the NMOS logic is off. The *W/L* of this MOSFET is small (long length and minimum width), so that it provides enough current to compensate for the leakage but not so much that the NMOS logic can't drive node A down to ground. The long-length keeper MOSFET is said to be weak. Sometimes, to implement the keeper MOSFET, two MOSFETs are used in series, as seen in Fig. 14.16b. Connecting a long-length MOSFET, and thus large area MOSFET, to the inverter's output can add unneeded load capacitance. By using a regular switch in series with a long-length PMOS device, the loading on the output of the inverter can be eliminated.

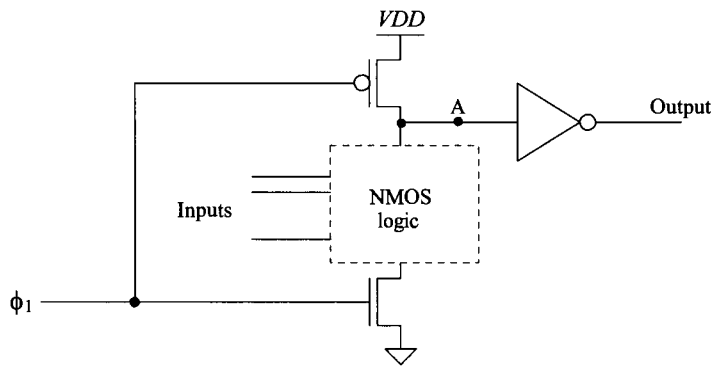
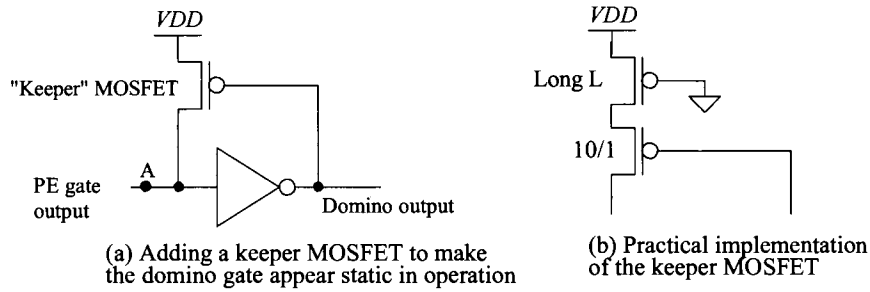


Figure 14.15 Domino logic gate.



(a) Adding a keeper MOSFET to make the domino gate appear static in operation

(b) Practical implementation of the keeper MOSFET

Figure 14.16 Keeper MOSFET used to hold node A in Fig. 14.15 at VDD when PE gate output is high.

NP Logic (Zipper Logic)

The idea behind implementing a logic function using NP logic is shown in Fig. 14.17. Staggering NMOS and PMOS stages eliminates the need for, and delay associated with, the inverter used in Domino logic, making higher speed operation possible. A circuit that can easily be implemented in NP logic is the full adder circuit of Fig. 12.20. The NMOS section of the carry circuit is implemented in the first section of the NP logic, while the PMOS section of the sum circuit is implemented in the PMOS section of the NP logic gate.

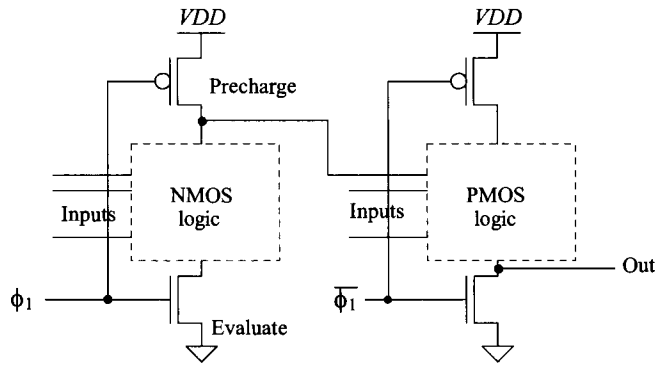


Figure 14.17 NP logic.

Pipelining

The NP logic adder just described adds two 1-bit words with carry during each clock cycle. Adding two 4-bit words can use pipelining, see Fig. 14.18. The bits of the word are delayed, both on the input and output of the adder, so that all bits of the sum reach the output of the adder at the same time. Note, however, that two new 4-bit words can be input to the adder at the beginning of each clock cycle and that it takes four clock cycles

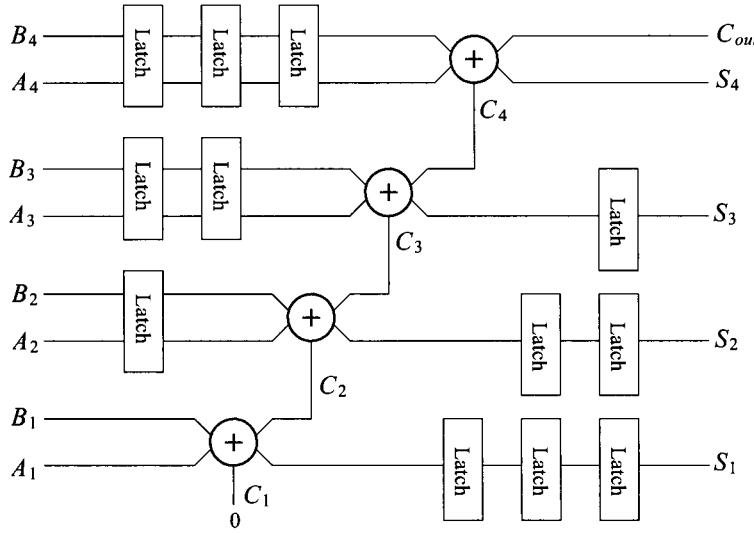


Figure 14.18 A pipelined adder. The latches (clocked) behave as delay elements.

to finish the addition of the two words. If this circuit were dedicated to continually performing the addition of two words, we could input the words at a very fast rate. However, since performing a single addition requires four clock cycles, applications of pipelining where two numbers are not added continuously can result in longer delay-times.

ADDITIONAL READING

[1] K. Bernstein, K. M. Carrig, C. M. Durham, P. R. Hansen, D. Hogenmiller, E. J. Nowak, and N. J. Rohrer, *High Speed CMOS Design Styles*, Springer, 1999. ISBN 978-0792382201.

[2] J. Yuen and C. Svensson, "New Single-Clock CMOS Latches and Flipflops with Improved Speed and Power Savings," *IEEE Journal of Solid-State Circuits,* vol. 32, no. 1, pp. 62–69, 1997.

[3] M. I. Elmasry, *Digital MOS Integrated Circuits II,* IEEE Press, 1992. ISBN 0-87942-275-0, IEEE order number: PC0269-1.

[4] J. P. Uyemura, *Circuit Design for Digital CMOS VLSI,* Kluwer Academic Publishers, 1992.

[5] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI-Design Techniques for Analog and Digital Circuits,* McGraw-Hill Publishing Co., 1990. ISBN 0-07-023253-9.

PROBLEMS

Unless otherwise stated, use the 50 nm, short-channel process.

14.1 Regenerate Fig. 14.2 for the PMOS device. From the results, determine the PMOS's I_{off}.

14.2 Repeat Ex. 14.2 if the storage node is a logic 0 (ground). Explain why the charge storage node charges up. What would happen if the PG's input were held at *VDD* instead of *VDD*/2.

14.3 Comment on the usefulness of dynamic logic in our 50 nm CMOS process-based on the results given in Ex. 14.3 with a clock frequency of 10 MHz.

14.4 Using the circuit in Fig. 14.6 and the SPICE simulation, show how the current drawn from *VDD*, by the inverters, changes with time. Do you see any concerns? If so, what?

14.5 Simulate the operation of the nonoverlapping clock generator circuit in Fig. 14.9. Assume that the input clock signal is running at 100 MHz. Show how both ϕ_1 and ϕ_2 are nonoverlapping.

14.6 Simulate the operation of the clocked CMOS latch shown in Fig. 14.11.

14.7 Design and simulate the operation of a PE gate that will implement the logical function $F = \overline{ABCD + E}$.

14.8 If the PE gate shown in Fig. 14.13 drives a 50 fF capacitor, estimate the worst-case t_{PHL}. Use a 20/1 PMOS and a 10/1 NMOS.

14.9 Implement an XOR gate using Domino logic. Simulate the operation of the resulting implementation.

14.10 The circuit shown in Fig. 14.19 results from the implementation of a high-speed adder cell (1-bit). What type of logic was used to implement this circuit? Using timing diagrams, describe the operation of the circuit.

14.11 Discuss the design of a 2-bit adder using the adder cell of Fig. 14.19 If a clock, running at 200 MHz, is used with the 2-bit adder, how long will it take to add two words? How long will it take if the word size is increased to 32 bits?

14.12 Sketch the implementation of an NP logic half adder cell.

14.13 Design (sketch the schematic of) a full adder circuit using PE logic.

14.14 Simulate the operation of the circuit designed in Problem 14.10.

14.15 Show that the dynamic circuit shown in Fig. 14.20 is an edge-triggered flip-flop [2]. Note that a single-phase clock signal is used.

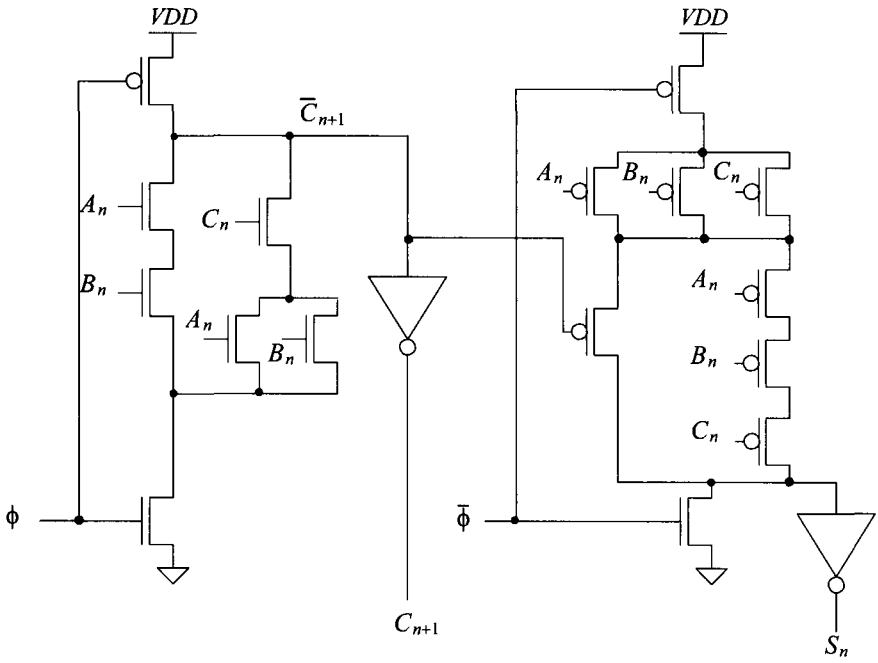


Figure 14.19 A high speed adder cell. See Problem 14.10.

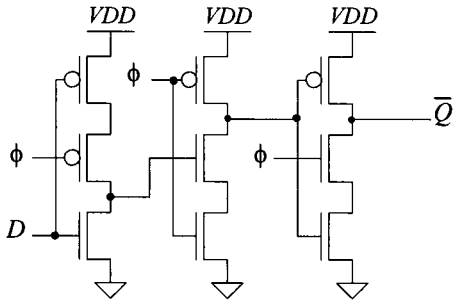


Figure 14.20 A true-single phase clocked FF, see Problem 14.15.