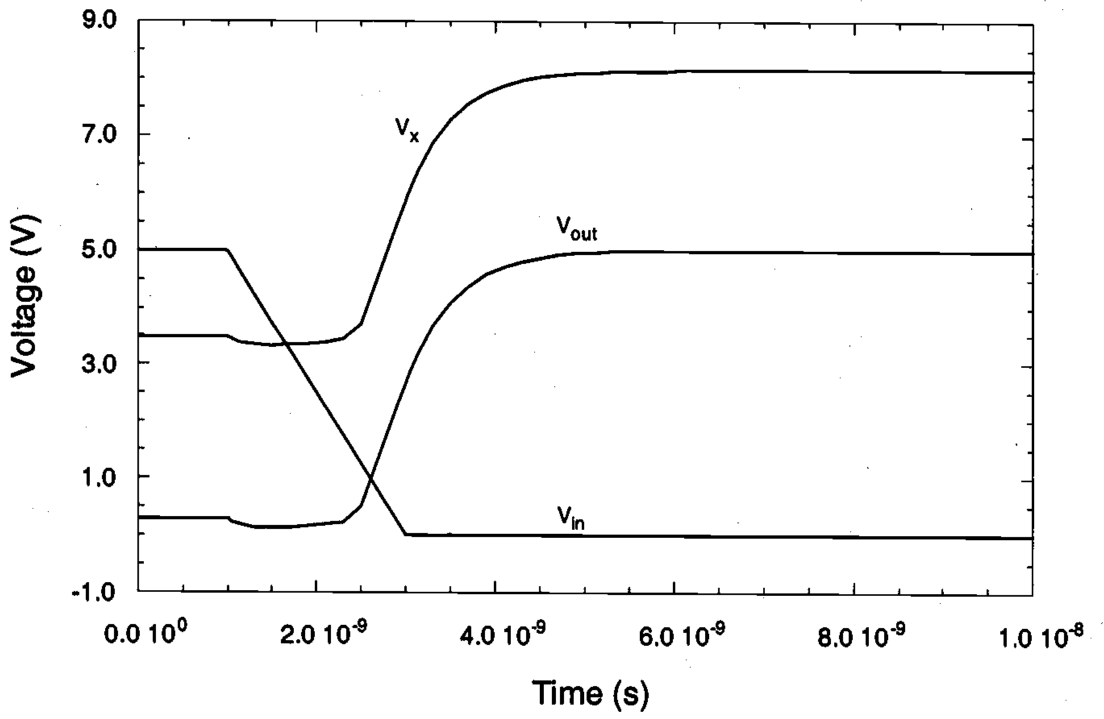
Since its drain and source terminals are connected together, the dummy transistor simply acts as an MOS capacitor between V_x and V_{out}. Although this circuit arrangement contains two additional transistors to achieve voltage bootstrapping, the resulting circuit-performance improvement is usually well worth the extra silicon area used for the bootstrapping devices.

Example 9.3.

The transient operation of the simple bootstrap circuit shown in Fig. 9.13 is simulated using SPICE in the following. To provide the needed bootstrap capacitance C_{boot}, a dummy nMOS device with channel length $L = 5$ µm and channel width $W = 50$ µm is used. Transistor M1 has a (W/L) ratio of 2, while M2 and M3 each have a (W/L) ratio of 1.



9.4. Synchronous Dynamic Circuit Techniques

Having examined the basic concepts associated with temporary storage of logic levels in capacitive circuit nodes, we now turn our attention to digital circuit design techniques which take advantage of this simple yet effective principle. In the following, we will investigate different examples of synchronous dynamic circuits implemented using depletion-load nMOS, enhancement-load nMOS, and CMOS building blocks.

Consider the generalized view of a multi-stage synchronous circuit shown in Fig. 9.14. The circuit consists of cascaded combinational logic stages, which are interconnected through nMOS pass transistors. All inputs of each combinational logic block are driven by a single clock signal. Individual input capacitances are not shown in this figure for simplicity, but the operation of the circuit obviously depends on temporary charge storage in the parasitic input capacitances.

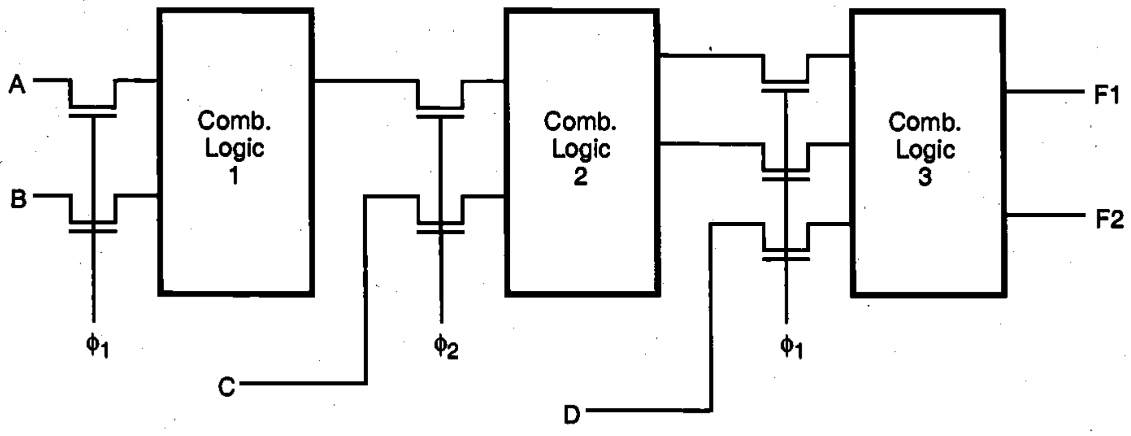


Figure 9.14. Multi-stage pass transistor logic driven by two nonoverlapping clocks.

To drive the pass transistors in this system, two nonoverlapping clock signals, ϕ_1 and ϕ_2, are used. The nonoverlapping property of the two clock signals guarantees that at any given time point, only one of the two clock signals can be active, as illustrated in Fig. 9.15. When clock ϕ_1 is active, the input levels of Stage 1 (and also of Stage 3) are applied through the pass transistors, while the input capacitances of Stage 2 retain their previously set logic levels. During the next phase, when clock ϕ_2 is active, the input levels of Stage 2 will be applied through the pass transistors, while the input capacitances of Stage 1 and Stage 3 retain their logic levels. This allows us to incorporate the simple dynamic memory function at each stage input, and at the same time, to facilitate synchronous operation by controlling the signal flow in the circuit using the two periodic clock signals. This signal timing scheme is also called *two-phase clocking* and is one of the most widely used timing strategies.

By introducing the two-phase clocking scheme, we have not made any specific assumptions about the internal structure of the combinational logic stages. It will be seen that depletion-load nMOS, enhancement-load nMOS, or CMOS logic circuits can be used for implementing the combinational logic. Figure 9.16 shows a depletion-load dynamic shift register circuit, in which the input data are inverted once and transferred, or *shifted* into the next stage during each clock phase.

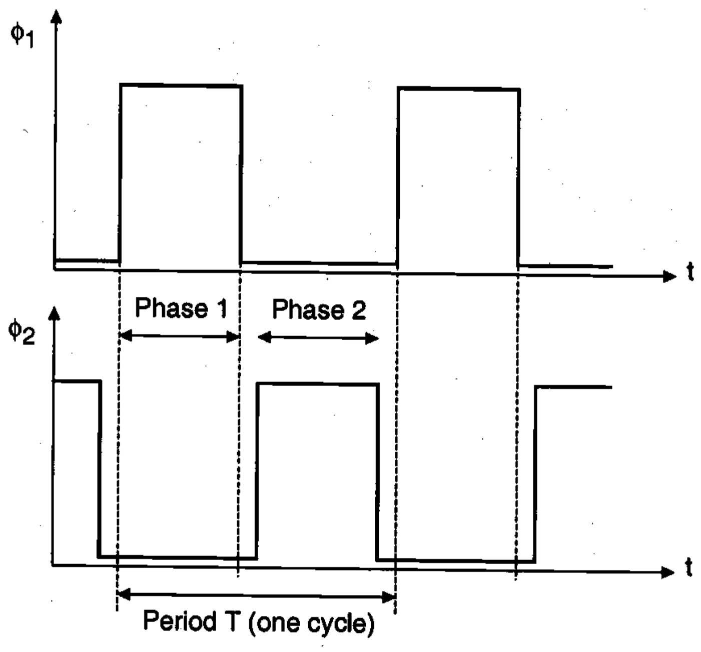


Figure 9.15. Nonoverlapping clock signals used for two-phase synchronous operation.

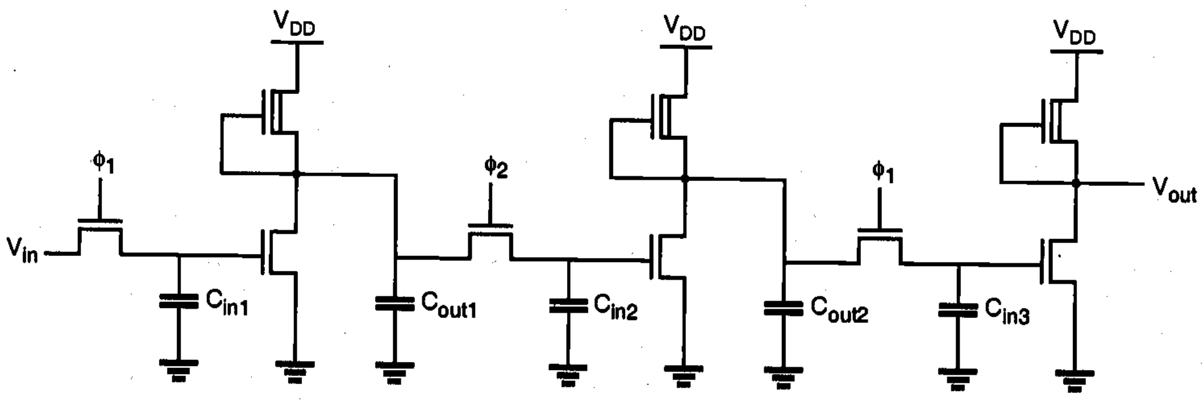


Figure 9.16. Three stages of a depletion-load nMOS dynamic shift register circuit driven with two-phase clocking.

The operation of the shift register circuit is as follows. During the active phase of ϕ_1, the input voltage level V_{in} is transferred into the input capacitance C_{in1}. Thus, the valid output voltage level of the first stage is determined as the inverse of the current input during this cycle. When ϕ_2 becomes active during the next phase, the output voltage level of the first stage is transferred into the second stage input capacitance C_{in2}, and the valid output voltage level of the second stage is determined. During the active ϕ_2 phase, the first-stage input capacitance continues to retain its previous level via charge storage. When ϕ_1 becomes active again, the original data bit *written* into the register during the previous cycle is transferred into the third stage, and the first stage can now accept the next data bit.

In this circuit, the maximum clock frequency is determined by the signal propagation delay through one inverter stage. One half-period of the clock signal must be long enough to allow the input capacitance C_{in} to charge up or down, and the logic level to propagate to the output by charging C_{out}. Also notice that the logic-high input level of each inverter stage in this circuit is one threshold voltage lower than the power supply voltage level.

The same operation principle used in the simple shift register circuit can easily be extended to synchronous complex logic. Figures 9.17 and 9.18 show a two-stage circuit example implemented using depletion-load nMOS complex logic gates.

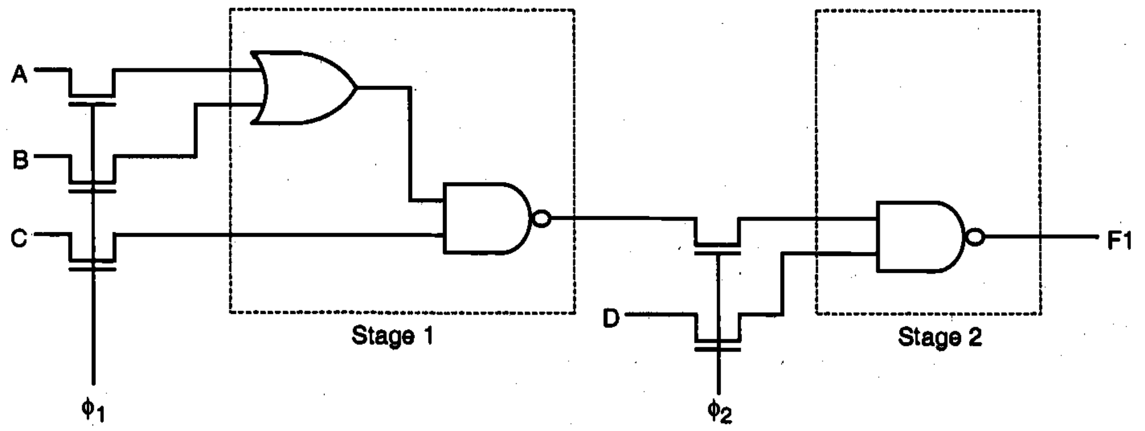


Figure 9.17. A two-stage synchronous complex logic circuit example.

In a complex logic circuit such as the one shown in Fig. 9.18, we see that the signal propagation delay of each stage may be different. Thus, in order to guarantee that correct logic levels are propagated during each active clock cycle, the half-period length of the clock signal must be longer than the largest single-stage signal propagation delay found in the circuit.

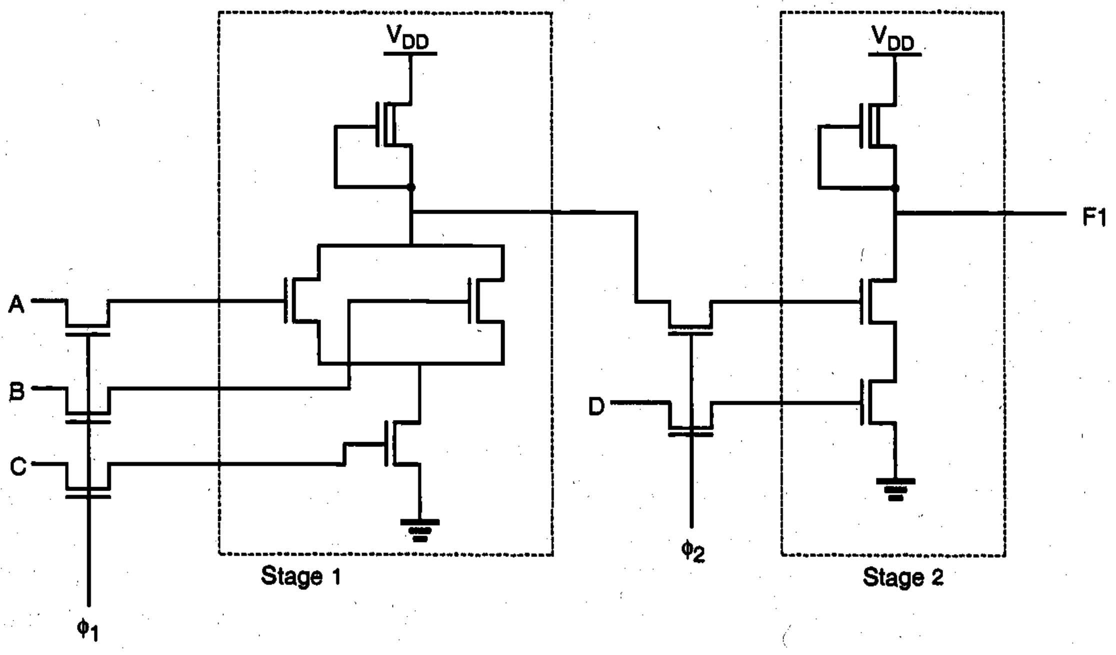


Figure 9.18. Depletion-load nMOS implementation of synchronous complex logic.

Now consider a different implementation of the simple shift register circuit, using enhancement-load nMOS inverters. One important difference is that, instead of biasing the load transistors with a constant gate voltage, we apply the clock signal to the gate of the load transistor as well. It can be shown that the power dissipation and the silicon area can be reduced significantly by using this dynamic (clocked) load approach. Two variants of the dynamic enhancement-load shift register will be examined in the following, both of which are driven by two non-overlapping clock signals. Figure 9.19 shows the first implementation, where in each stage the input pass transistor and the load transistor are driven by opposite clock phases, ϕ_1 and ϕ_2.

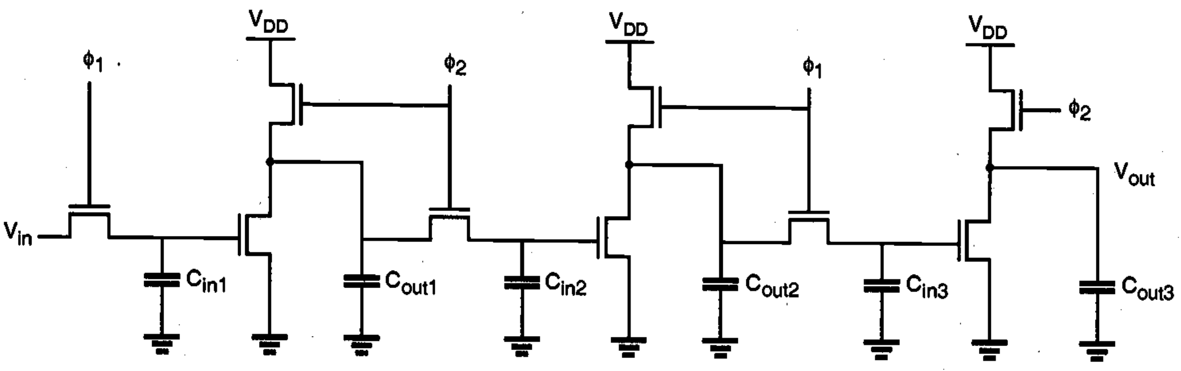


Figure 9.19. Enhancement-load dynamic shift register (ratioed logic).

When ϕ_1 is active, the input voltage level V_{in} is transferred into the first-stage input capacitance C_{in1} through the pass transistor. In this phase, the enhancement-type nMOS load transistor of the first-stage inverter is not active yet. During the next phase (active ϕ_2), the load transistor is turned on. Since the input logic level is still being preserved in C_{in1}, the output of the first inverter stage attains its valid logic level. At the same time, the input pass transistor of the second stage is also turned on, which allows this newly determined output level to be transferred into the input capacitance C_{in2} of the second stage. When clock ϕ_1 becomes active again, the valid output level across C_{out2} is determined, and transferred into C_{in3}. Also, a new input level can be accepted (*pipelined*) into C_{in1} during this phase.

In this circuit, the valid low-output voltage level V_{OL} of each stage is strictly determined by the driver-to-load ratio, since the output pass transistor (input pass transistor of next stage) turns on in phase with the load transistor. Therefore, this circuit arrangement is also called *ratioed dynamic logic*. The basic operation principle can obviously be extended to arbitrary complex logic, as shown in Fig. 9.20. Since the power supply current flows only when the load devices are activated by the clock signal, the overall power consumption of dynamic enhancement-load logic is generally lower than for depletion-load nMOS logic.

Next, consider the second dynamic enhancement-load shift register implementation where, in each stage, the input pass transistor and the load transistor are driven by the same clock phase (Fig. 9.21).

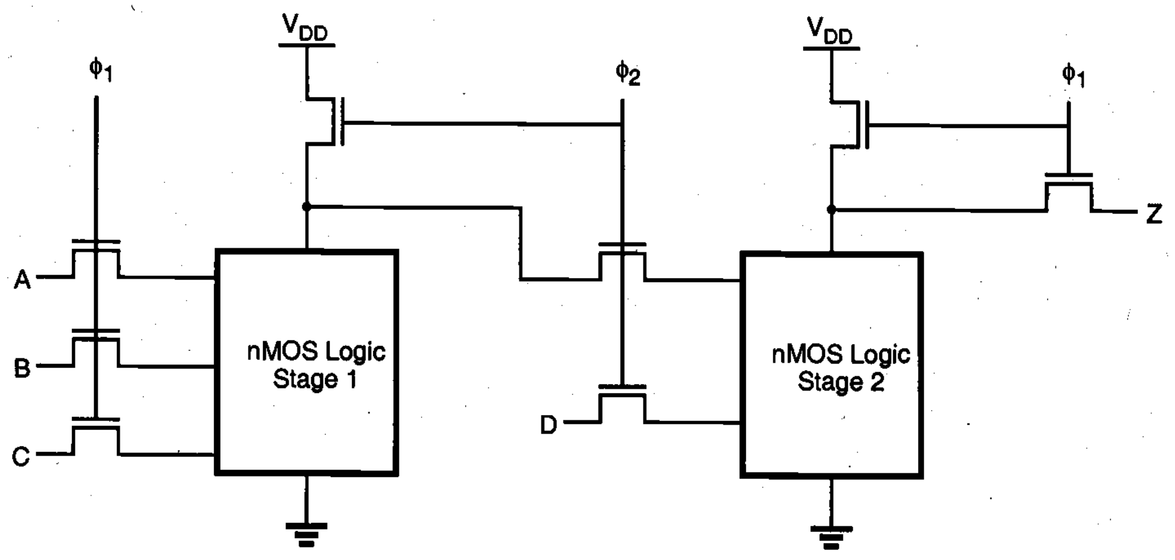


Figure 9.20. General circuit structure of ratioed synchronous dynamic logic.

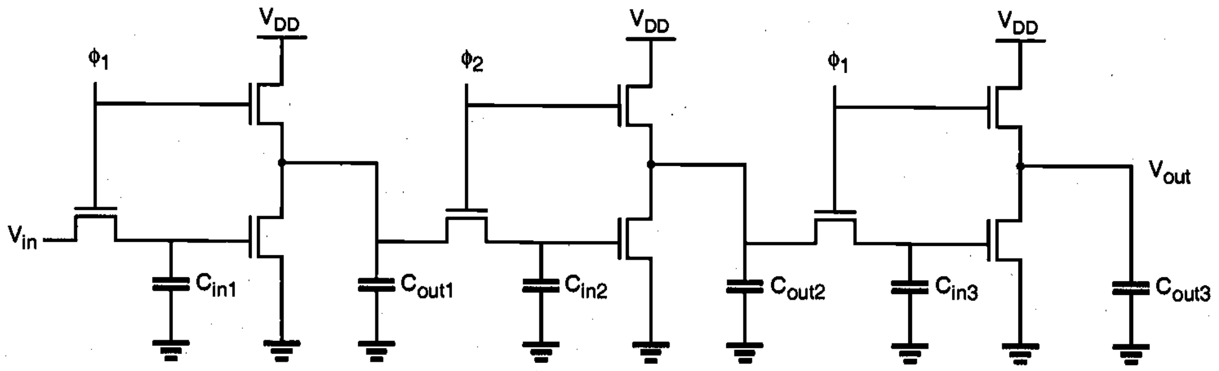


Figure 9.21. Enhancement-load dynamic shift register (ratioless logic).

When ϕ_1 is active, the input voltage level V_{in} is transferred into the first-stage input capacitance C_{in1} through the pass transistor. Note that at the same time, the enhancement-type nMOS load transistor of the first-stage inverter is active. Therefore, the output of the first inverter stage attains its valid logic level. During the next phase (active ϕ_2), the input pass transistor of the next stage is turned on, and the logic level is transferred onto the next stage. Here, we have to consider two cases, as follows.

If the output level across C_{out1} is logic-high at the end of the active ϕ_1 phase, this voltage level is transferred to C_{in2} via charge sharing over the pass transistor during the active ϕ_2 phase. Note that the logic-high level at the output node is subject to threshold voltage drop, i.e., it is one threshold voltage lower than the power supply voltage. To correctly transfer a logic-high level after charge sharing, the ratio of the capacitors (C_{out}/ C_{in}) must be made large enough during circuit design.

If, on the other hand, the output level of the first stage is logic-low at the end of the active ϕ_1 phase, then the output capacitor C_{out1} will be completely drained to a voltage of

$V_{OL} = 0$ V when ϕ_1 turns off. This can be achieved because a logic-high level is being stored in the input capacitance C_{in1} in this case, which forces the driver transistor to remain in conduction. Obviously, the logic-low level of $V_{OL} = 0$ V is also transferred into the next stage via the pass transistor during the active ϕ_2 phase.

When clock ϕ_1 becomes active again, the valid output level across C_{out2} is determined and transferred into C_{in3}. Also, a new input level can be accepted into C_{in1} during this phase. Since the valid logic-low level of $V_{OL} = 0$ V can be achieved regardless of the driver-to-load ratio, this circuit arrangement is called *ratioless dynamic logic*. The basic operation principle can be extended to arbitrary complex logic, as shown in Fig. 9.22.

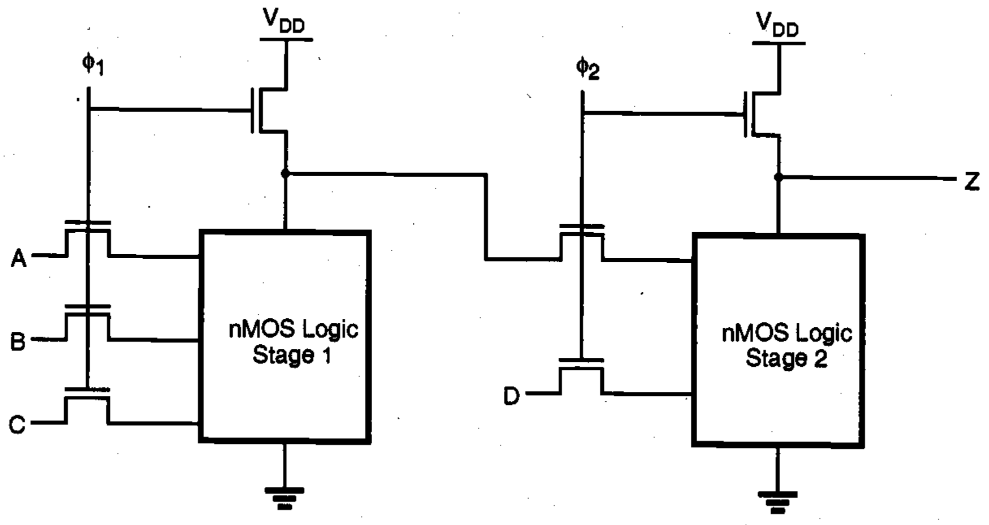


Figure 9.22. General circuit structure of ratioless synchronous dynamic logic.

CMOS Transmission Gate Logic

The basic two-phase synchronous logic circuit principle, in which individual logic blocks are cascaded via clock-controlled switches, can easily be adopted to CMOS structures as well. Here, static CMOS gates are used for implementing the logic blocks, and CMOS transmission gates are used for transferring the output levels of one stage to the inputs of the next stage (Fig. 9.23). Notice that each transmission gate is actually controlled by the clock signal *and* its complement. As a result, two-phase clocking in CMOS transmission gate logic requires that a total of four clock signals are generated and routed throughout the circuit.

As in the nMOS-based dynamic circuit structures, the operation of CMOS dynamic logic relies on charge storage in the parasitic input capacitances during the inactive clock cycles. To illustrate the basic operation principles, the fundamental building block of a dynamic CMOS transmission gate shift register is shown in Fig. 9.24. It consists of a CMOS inverter, which is driven by a CMOS transmission gate. During the active clock phase (CK =1), the input voltage V_{in} is transferred onto the parasitic input capacitance C_x via the transmission gate. Note that the low on-resistance of the CMOS transmission gate usually results in a smaller transfer time compared to those for nMOS-only switches. Also, there is no threshold voltage drop across the CMOS transmission gate. When the clock signal becomes inactive, the CMOS transmission gate turns off and the voltage level across C_x can be preserved until the next cycle.

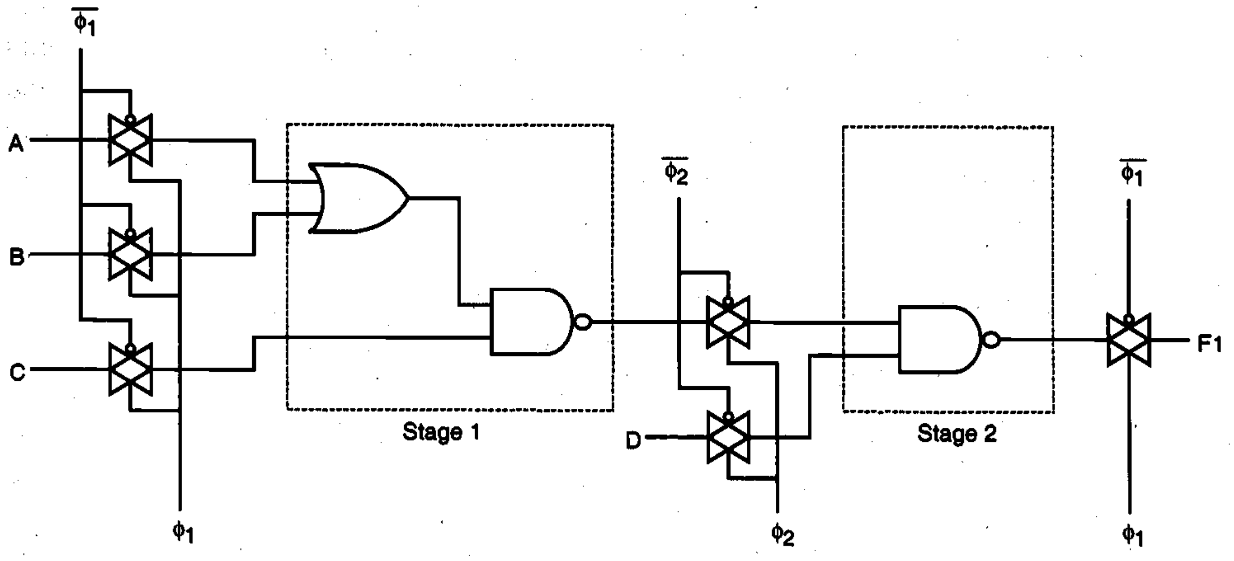


Figure 9.23. Typical example of dynamic CMOS transmission gate logic.

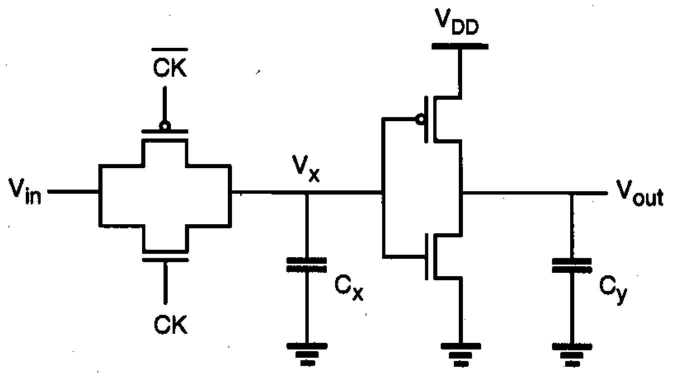


Figure 9.24. Basic building block of a CMOS transmission gate dynamic shift register.

Figure 9.25 shows a single-phase CMOS shift register, which is built by cascading identical units as in Fig. 9.24 and by driving each stage alternately with the clock signal and its complement.

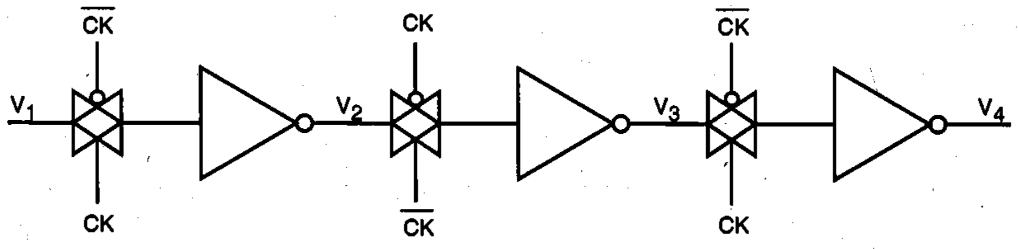


Figure 9.25. Single-phase CMOS transmission gate dynamic shift register.

Ideally, the transmission gates of the odd-numbered stages would conduct during the active clock phase (when CK = 1), while the transmission gates of the even-numbered stages are off, so that the cascaded inverter stages in the chain are alternately isolated. This would ensure that inputs are permitted in alternating half cycles. In practice, however, the clock signal and its complement do not constitute a truly nonoverlapping signal pair, since the clock voltage waveform has finite rise and fall times. Also, the clock skew between CK and $\overline{CK}$ may be unavoidable because one of the signals is generated by inverting the other. Therefore, true two-phase clocking with two nonoverlapping clock signals (ϕ_1 and ϕ_2) *and* their complements is usually preferred over single-phase clocking in dynamic CMOS transmission gate logic.

Dynamic CMOS Logic (Precharge-Evaluate Logic)

In the following, we will introduce a dynamic CMOS circuit technique which allows us to significantly reduce the number of transistors used to implement any logic function. The circuit operation is based on first *precharging* the output node capacitance and subsequently, *evaluating* the output level according to the applied inputs. Both of these operations are scheduled by a single clock signal, which drives one nMOS and one pMOS transistor in each dynamic stage. A dynamic CMOS logic gate which implements the function $F = \overline{(A_1A_2A_3 + B_1B_2)}$ is shown in Fig. 9.26.

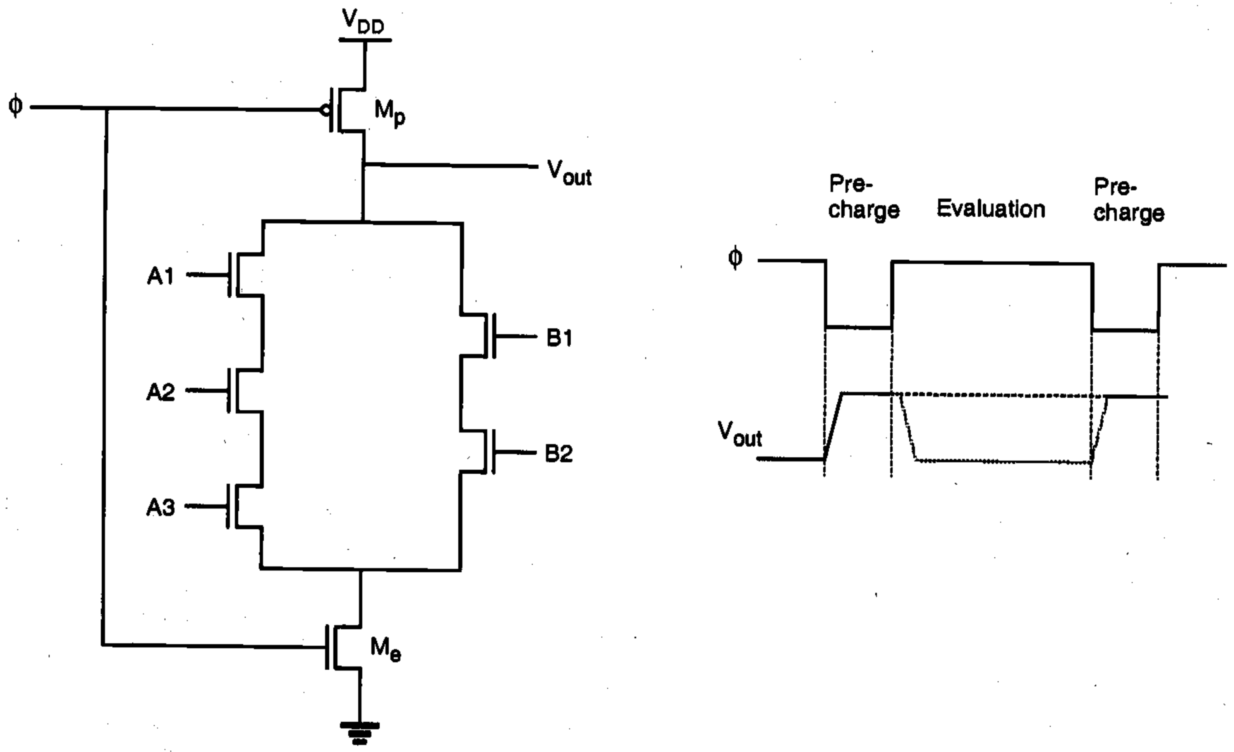


Figure 9.26. Dynamic CMOS logic gate implementing a complex Boolean function.

When the clock signal is low (precharge phase), the pMOS precharge transistor M_p is conducting, while the complementary nMOS transistor M_e is off. The parasitic output capacitance of the circuit is charged up through the conducting pMOS transistor to a logic-high level of $V_{out} = V_{DD}$. The input voltages are also applied during this phase, but they have no influence yet upon the output level since M_e is turned off.

When the clock signal becomes high (evaluate phase), the precharge transistor M_p turns off and M_e turns on. The output node voltage may now remain at the logic-high level or drop to a logic low, depending on the input voltage levels. If the input signals create a conducting path between the output node and the ground, the output capacitance will discharge toward $V_{OL} = 0$ V. The final discharged output level depends on the time span of the evaluation phase. Otherwise, V_{out} remains at V_{DD}.

The operation of the single-stage dynamic CMOS logic gate is quite straightforward. For practical multi-stage applications, however, the dynamic CMOS gate presents a significant problem. To examine this fundamental limitation, consider the two-stage cascaded structure shown in Fig. 9.27. Here, the output of the first dynamic CMOS stage drives one of the inputs of the second dynamic CMOS stage, which is assumed to be a two-input NAND gate for simplicity.

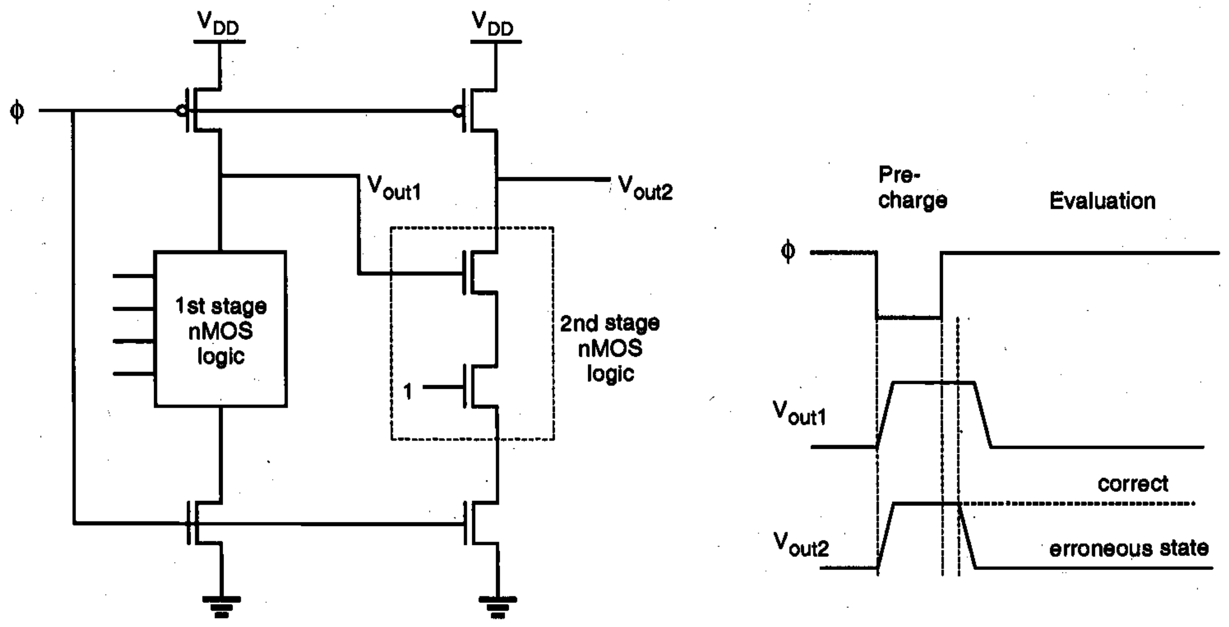


Figure 9.27. Illustration of the cascading problem in dynamic CMOS logic.

During the precharge phase, both output voltages V_{out1} and V_{out2} are pulled up by the respective pMOS precharge devices. Also, the external inputs are applied during this phase. The input variables of the first stage are assumed to be such that the output V_{out1} will drop to logic "0" during the evaluation phase. On the other hand, the external input of the second-stage NAND2 gate is assumed to be a logic "1," as shown in Fig. 9.27. When the evaluation phase begins, both output voltages V_{out1} and V_{out2} are logic-high. The output of the first stage (V_{out1}) eventually drops to its correct logic level after a certain time delay. However, since the evaluation in the second stage is done concurrently, starting with the high value of V_{out1} at the beginning of the evaluation phase, the output voltage V_{out2} at the end of the evaluation phase will be *erroneously* low. Although the first stage output subsequently assumes its correct output value once the stored charge is drained, the correction of the second-stage output is not possible.

This example illustrates that dynamic CMOS logic stages driven by the same clock signal cannot be cascaded directly. This severe limitation seems to undermine all the other advantages of dynamic CMOS logic, such as low power dissipation, large noise margins, and low transistor count. Alternative clocking schemes and circuit structures must be developed to overcome this problem. In fact, the search for viable circuit alternatives has spawned a large array of high-performance dynamic CMOS circuit techniques, some of which will be examined in the following section.

9.5. High-Performance Dynamic CMOS Circuits

The circuits presented here are variants of the basic dynamic CMOS logic gate structure. We will see that they are designed to take full advantage of the obvious benefits of dynamic operation and at the same time, to allow unrestricted cascading of multiple stages. The ultimate goal is to achieve reliable, high-speed, compact circuits using the least complicated clocking scheme possible.

Domino CMOS Logic

Consider the generalized circuit diagram of a domino CMOS logic gate shown in Fig. 9.28. A dynamic CMOS logic stage, such as the one shown in Fig. 9.26, is cascaded with a static CMOS inverter stage. The addition of the inverter allows us to operate a number of such structures in cascade, as explained in the following.

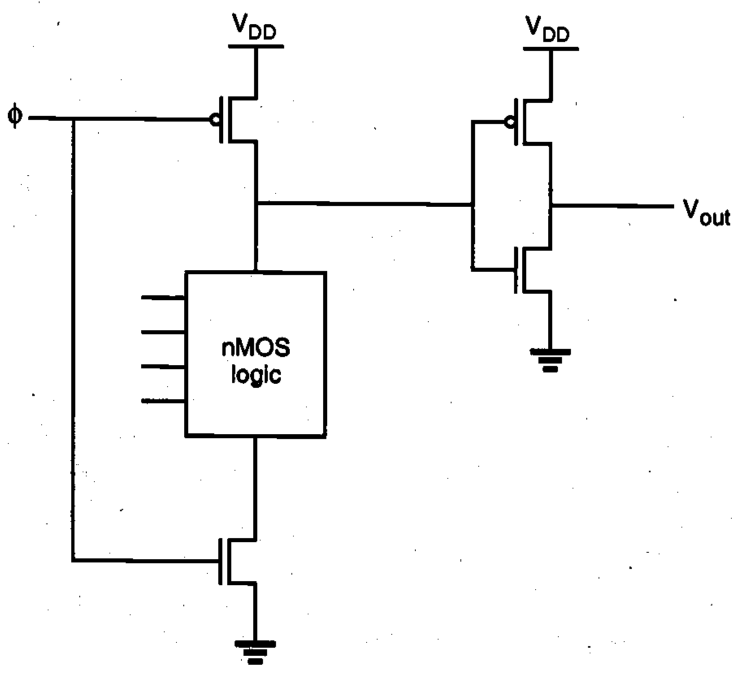


Figure 9.28. Generalized circuit diagram of a domino CMOS logic gate.

During the precharge phase (when CK = 0), the output node of the dynamic CMOS stage is precharged to a high logic level, and the output of the CMOS inverter (buffer) becomes low. When the clock signal rises at the beginning of the evaluation phase, there are two possibilities: The output node of the dynamic CMOS stage is either discharged to a low level through the nMOS circuitry (1 to 0 transition), or it remains high. Consequently, the inverter output voltage can also make at most one transition during the evaluation phase, from 0 to 1. Regardless of the input voltages applied to the dynamic CMOS stage, it is not possible for the buffer output to make a 1 to 0 transition during the evaluation phase.

Remember that the problem in cascading conventional dynamic CMOS stages occurs when one or more inputs of a stage make a 1 to 0 transition *during* the evaluation phase, as illustrated in Fig. 9.27. On the other hand, if we build a system by cascading domino CMOS logic gates as shown in Fig. 9.29, all input transistors in subsequent logic blocks will be turned off during the precharge phase, since all buffer outputs are equal to 0. During the evaluation phase, each buffer output can make at most one transition

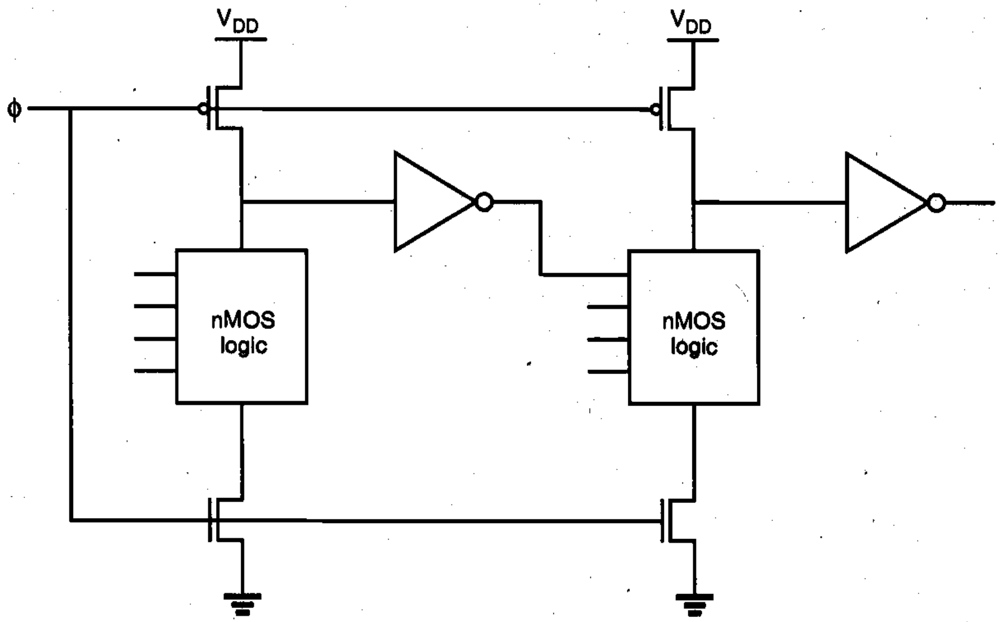
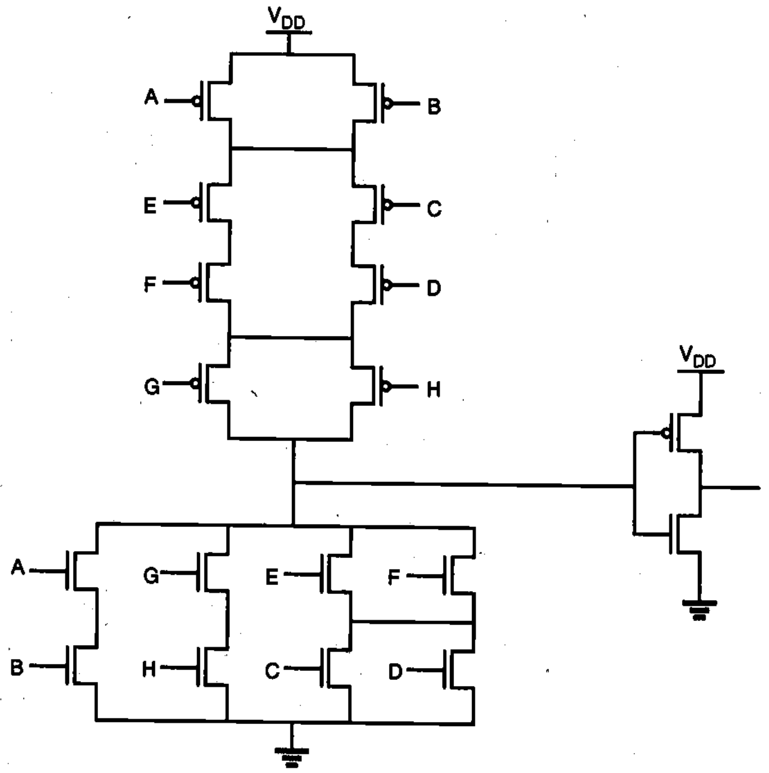


Figure 9.29. Cascaded domino CMOS logic gates.

(from 0 to 1), and thus each input of all subsequent logic stages can also make at most one (0 to 1) transition. In a cascade structure consisting of several such stages, the evaluation of each stage ripples the next stage evaluation, similar to a chain of dominos falling one after the other. The structure is hence called *domino CMOS logic.*

Domino CMOS logic gates allow a significant reduction in the number of transistors required to realize any complex Boolean function. The implementation of the 8-input Boolean function, $Z = AB + (C + D)(C + D) + GH$, using standard CMOS and domino CMOS, is shown in Fig. 9.30, where the reduction of circuit complexity is obvious. The distribution of the clock signal within the system is quite straightforward, since a single clock can be used to precharge and evaluate any number of cascaded stages, as long as the signal propagation delay from the first stage to the last stage does not exceed the time span of the evaluation phase. Also, conventional static CMOS logic gates can be used together with domino CMOS gates in a cascaded configuration (Fig. 9.31). The limitation is that the number of inverting static logic stages in cascade must be even, so that the inputs of the next domino CMOS stage experience only 0 to 1 transitions during the evaluation.

(a)



(b)

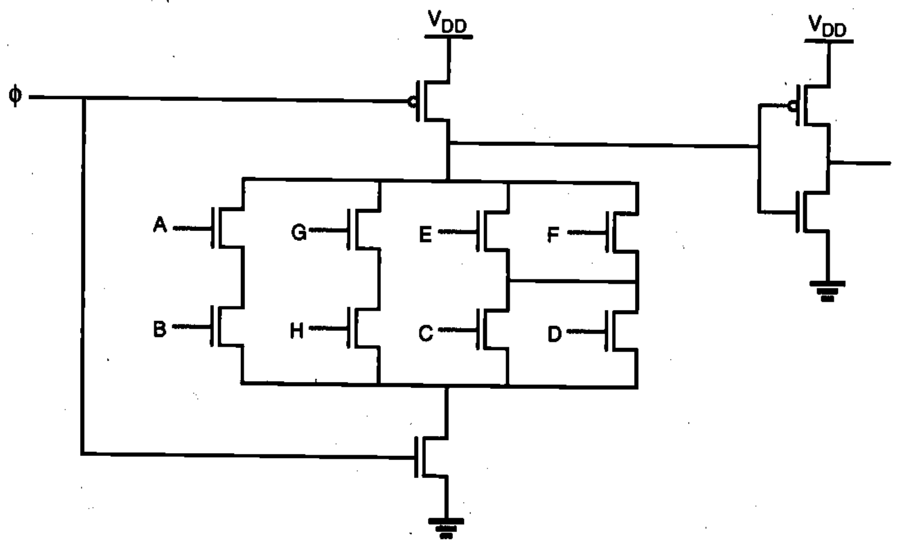


Figure 9.30. (a) An 8-input complex logic gate, realized using conventional CMOS logic and (b) domino CMOS logic.

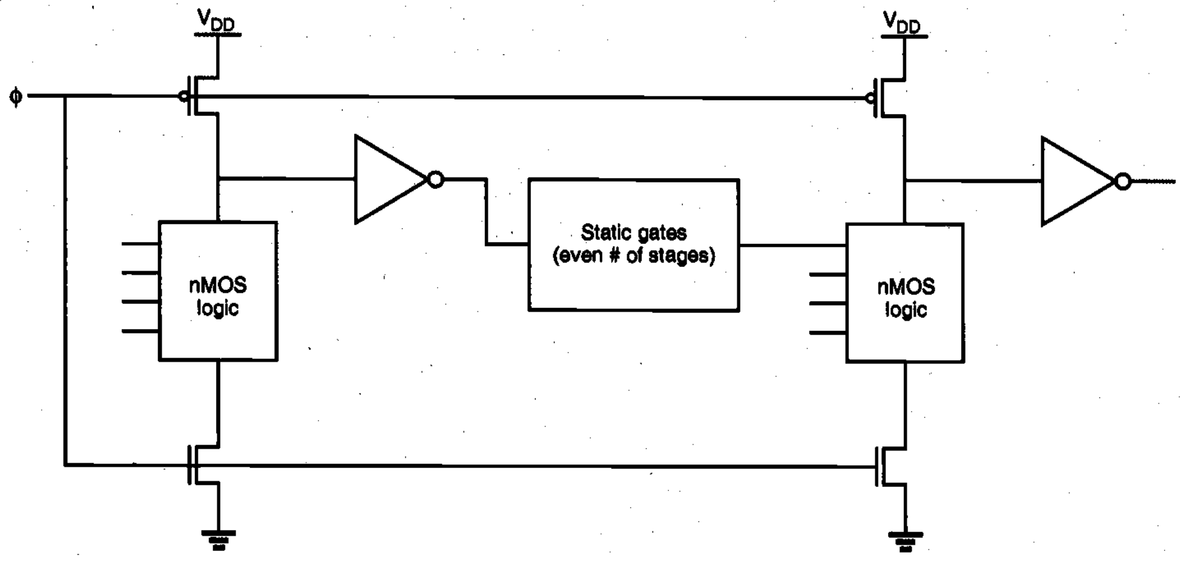


Figure 9.31. Cascading domino CMOS logic gates with static CMOS logic gates.

There are also some other limitations associated with domino CMOS logic gates. First, only non-inverting structures can be implemented using domino CMOS. If necessary, inversion must be carried out using conventional CMOS logic. Also, charge sharing between the dynamic stage output node and the intermediate nodes of the nMOS logic block during the evaluation phase may cause erroneous outputs, as will be explained in the following.

Consider the domino CMOS logic gate shown in Fig. 9.32, in which the intermediate node capacitance C_2 is comparable in size to the output node capacitance C_1. We will assume that all inputs are low initially, and that the intermediate node voltage across C_2 has an initial value of 0 V. During the precharge phase, the output node capacitance C_1 is charged up to its logic-high level of V_{DD} through the pMOS transistor. In the next phase, the clock signal becomes high and the evaluation begins. If the input signal of the uppermost nMOS transistor switches from low to high during this evaluation phase, as shown in Fig. 9.32, the charge initially stored in the output capacitance C_1 will now be shared by C_2, leading to the so-called *charge-sharing* phenomenon. The output node voltage after charge sharing becomes $V_{DD}/(1 + C_2/C_1)$. For example, if $C_1 = C_2$, the output voltage becomes $V_{DD}/2$ in the evaluation phase. Unless its logic threshold voltage is less than $V_{DD}/2$, the output voltage of the *following inverter* will then inadvertently switch high, which is a logic error. Thus, it is important to have C_2 much smaller than C_1.

Several measures can be taken in order to prevent erroneous output levels due to charge sharing in domino CMOS gates. One simple solution is to add a weak pMOS pull-up device (with a small (W/L) ratio) to the dynamic CMOS stage output, which essentially forces a high output level unless there is a strong pull-down path between the output and the ground (Fig. 9.33). It can be observed that the weak pMOS transistor will be turned on only when the precharge node voltage is kept high. Otherwise, it will be turned off as V_{out} becomes high.

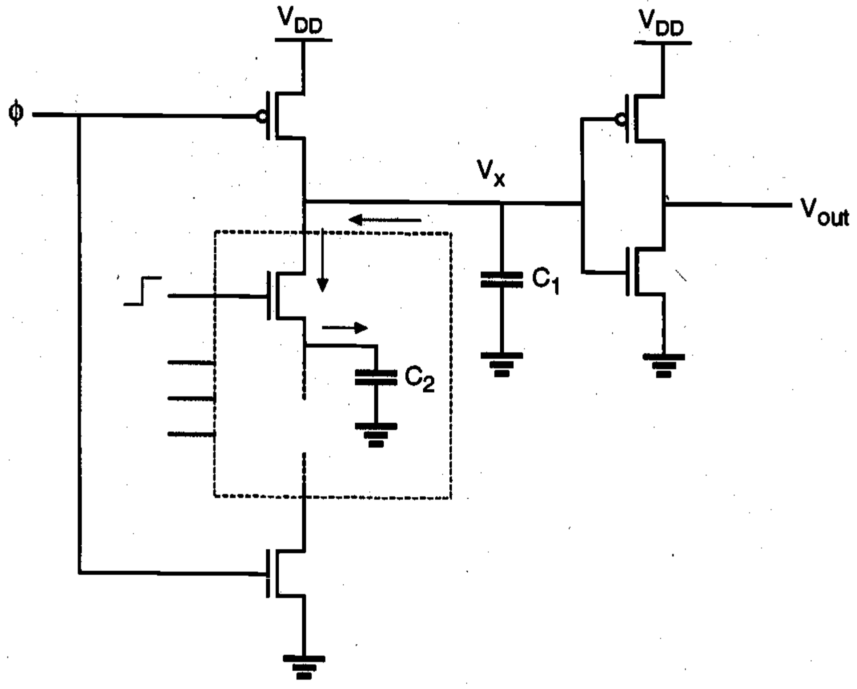


Figure 9.32. Charge sharing between the output capacitance C_1 and an intermediate node capacitance C_2 during the evaluation cycle may reduce the output voltage level.

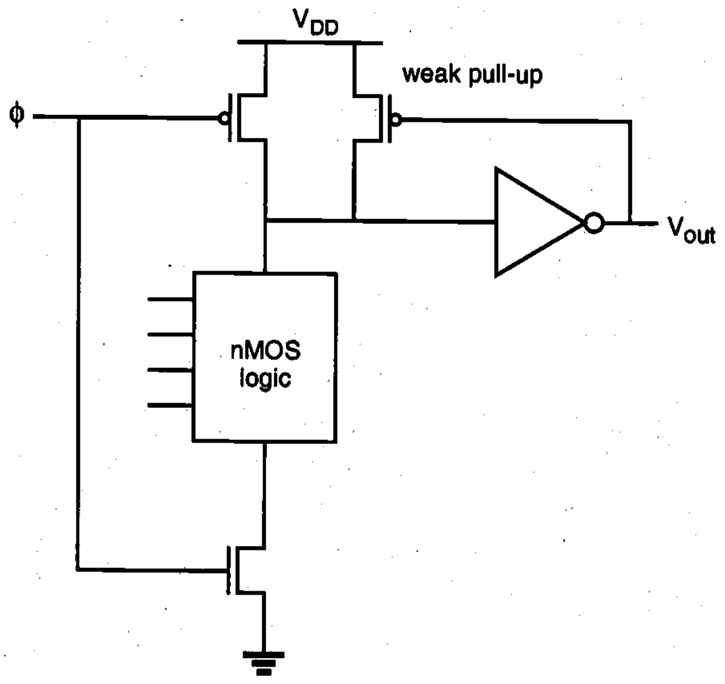


Figure 9.33. A weak pMOS pull-up device in a feedback loop can be used to prevent the loss of output voltage level due to charge sharing.

Another solution is to use separate pMOS transistors to precharge all intermediate nodes in the nMOS pull-down tree which have a large parasitic capacitance. The precharging of all high-capacitance nodes within the circuit effectively eliminates all potential charge-sharing problems during evaluation. However, it can also cause additional delay time since the nMOS logic tree now has to drain a larger charge in order to pull down the node voltage V_x. Another way of preventing logic errors due to charge sharing is to make the logic threshold voltage of the inverter smaller, such that the final stage output is not affected by lowering of V_x due to charge sharing. It should be noted that this design approach would trade off the pull-up speed (weaker pMOS transistor) for lower sensitivity to the charge-sharing problem.

The use of multiple precharge transistors also enables us to use the precharged intermediate nodes as resources for additional outputs. Thus, additional logic functions can be realized by tapping the internal nodes of the dynamic CMOS stage, as illustrated by two series-connected logic blocks in Fig. 9.34. The resulting multiple-output domino CMOS logic gate allows us to simultaneously realize several complex functions using a small number of transistors. Figure 9.35 shows the realization of four Boolean functions of nine variables, using a single domino CMOS logic gate. The four functions to be realized are listed in the following.

$$
\begin{aligned}
C_1 &= G_1 + P_1C_0 \\
C_2 &= G_2 + P_2G_1 + P_2P_1C_0 \\
C_3 &= G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1C_0 \\
C_4 &= G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1 + P_4P_3P_2P_1C_0
\end{aligned}
$$

It can be shown that the functions C_1 through C_4 are the four carry terms to be used in a four-stage carry-lookahead adder, where the variables G_i and P_i are defined as

$$
\begin{aligned}
G_i &= A_i \cdot B_i \\
P_i &= A_i \oplus B_i
\end{aligned}
$$

and A_i and B_i are the input bits associated with the i_{th} stage. Hence, this circuit is also known as the Manchester carry chain. The generation of the four carry terms using four separate standard CMOS logic gates or four separate single-output domino CMOS circuits, on the other hand, would require a large number of transistors and consequently a much larger silicon area. Variants of the multiple-output dynamic CMOS circuit shown in Fig. 9.35 are used widely in high-performance adder structures.

The transient performance of domino CMOS logic gates can be improved by adjusting the nMOS transistor sizes in the pull-down path, with the objective of reducing the discharge time. Shoji has shown that the best performance is obtained with a graded sizing of nMOS transistors in series structures, where the nMOS transistor closest to the output node also has the smallest (W/L) ratio. The domino CMOS circuit diagram and the corresponding stick-diagram layout of an optimized example are shown in Fig. 9.36. The fact that a graded reduction of transistor sizes from bottom to top ultimately leads to a better transient performance may seem counterintuitive. But, this effect can be explained by observing the RC delay of the combined pull-down path consisting of series-connected nMOS transistors.

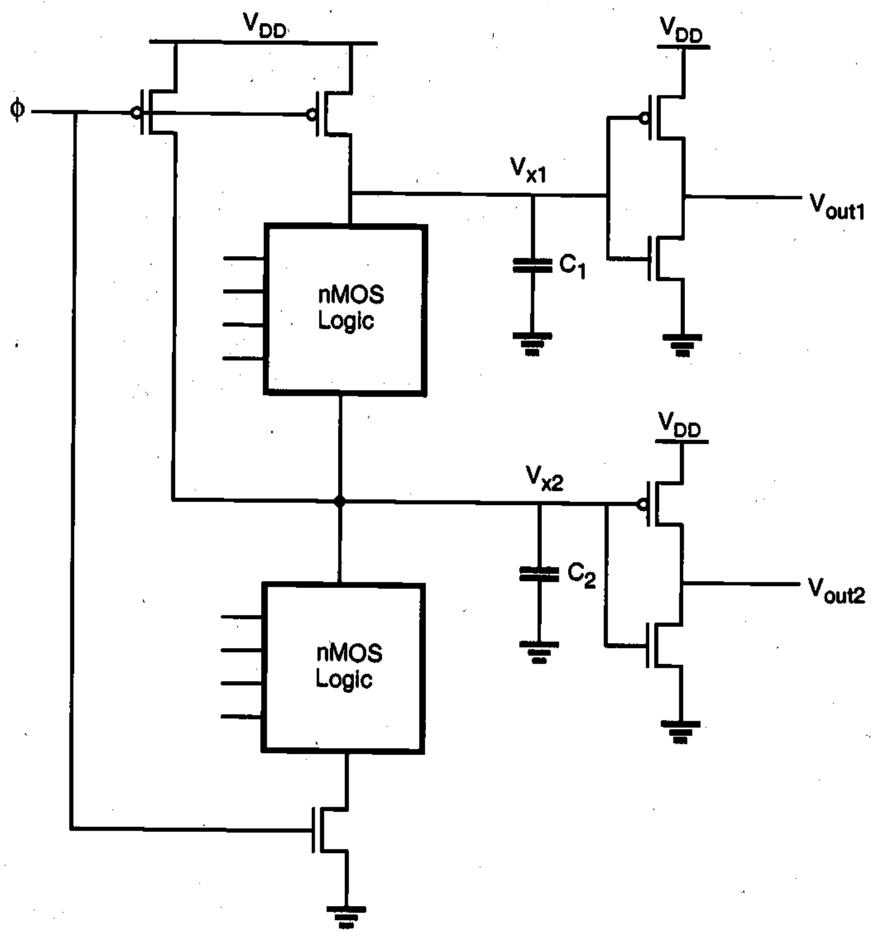


Figure 9.34. Precharging of internal nodes to prevent charge sharing also allows implementation of multiple-output domino CMOS structures.

Consider first the nMOS transistor closest to the output node. If the (W/L) ratio of this transistor is reduced by a certain factor, two effects occur. First, the current- driving capability will decrease, i.e., the equivalent resistance of the nMOS transistor will increase. Second, the parasitic drain capacitance associated with this transistor will decrease. If the length of the nMOS chain is sufficiently long, the increase in resistance has little influence upon the combined RC delay time, whereas a reduction of the capacitance significantly decreases the delay.

In fact, by applying Elmore's RC delay formula (see Chapter 6) to series-connected nMOS structures, one can determine if a reduction of nMOS transistor sizes will improve the transient performance. Let C_L represent the precharge-node load capacitance of the domino CMOS gate, and let C_1 represent the parasitic drain capacitance of the nMOS transistor closest to the precharge node. It is assumed that the inverter transistor sizes are fixed and that the pull-down chain contains N series-connected nMOS transistors. Shoji has shown that if the following condition

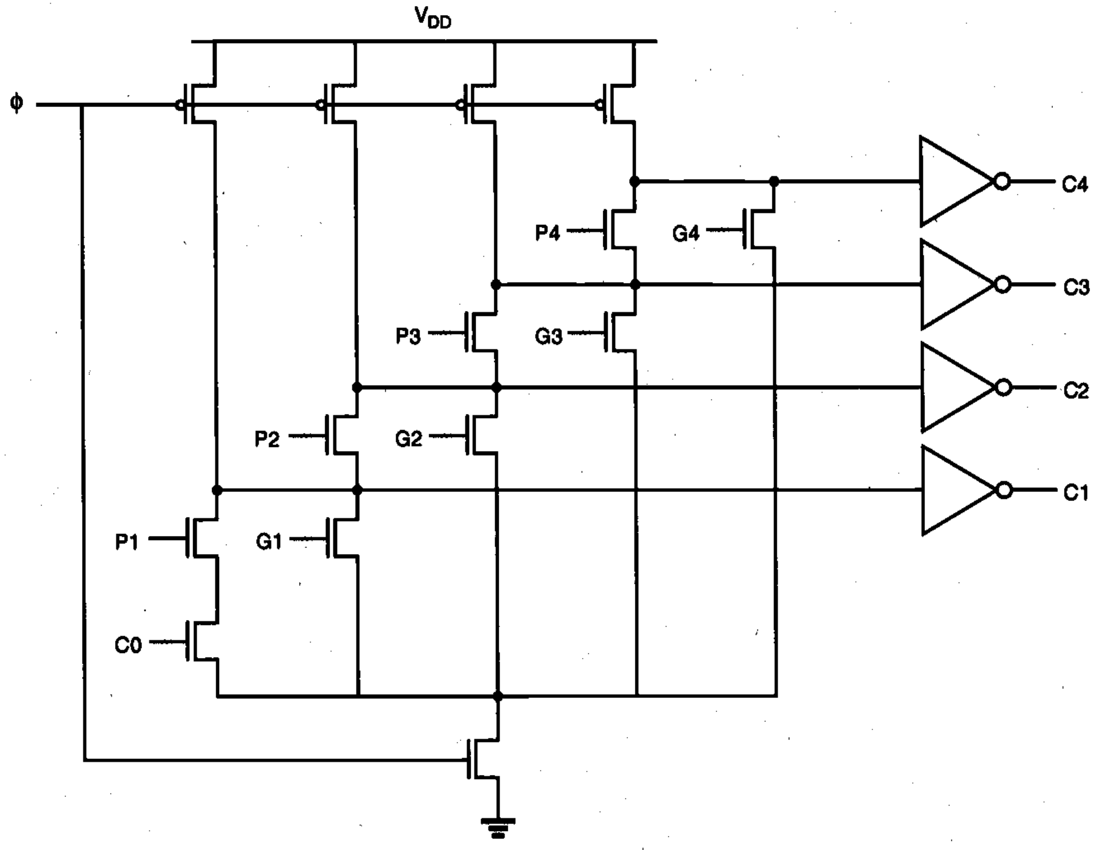


Figure 9.35. Example of a multiple-output domino CMOS gate realizing four functions.

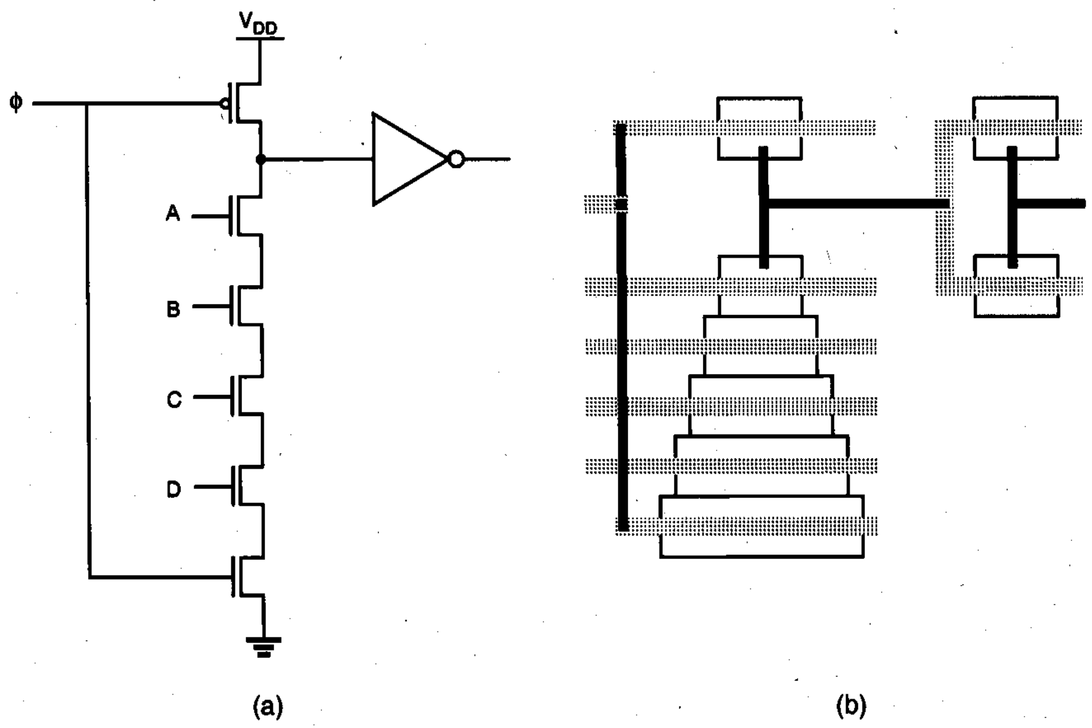


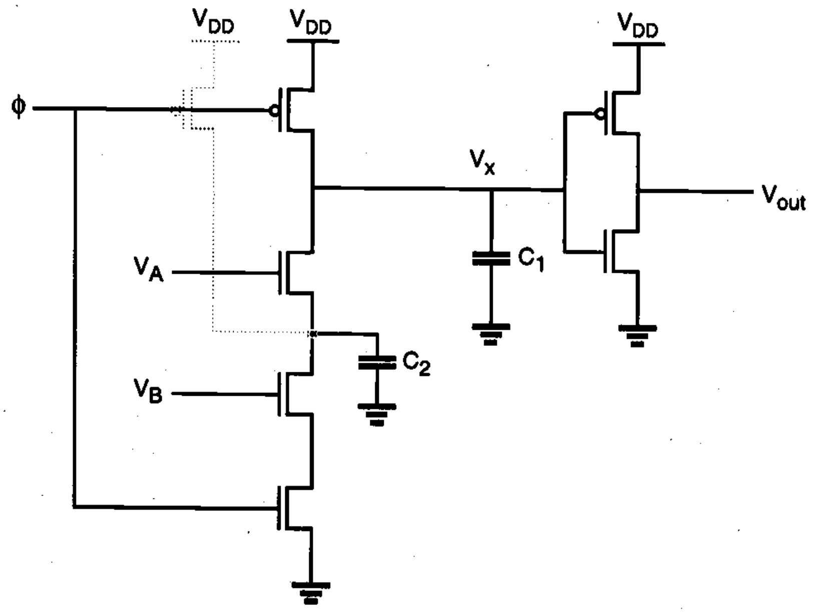
Figure 9.36. (a) Four-input domino CMOS NAND gate and (b) the corresponding stick-diagram layout to show the graded scaling of nMOS transistor sizes for improving the transient performance.

$$C_L < (N-1)\frac{C_1}{2} \tag{9.36}$$

is satisfied, the overall delay time can be reduced by decreasing the size of the nMOS transistor closest to the output node. This result can be iteratively applied to the other transistors in the pull-down chain, which leads to graded sizing of all nMOS devices. On the other hand, if the inverter gate transistors are allowed to be optimized along with the series-connected transistors, even shorter delays can be achieved with smaller chip area.

Example 9.4.

Consider the domino CMOS NAND2 gate shown below, where $C_1 = C_2 = 0.05$ pF. First, the operation of the circuit with only one pMOS precharge transistor will be examined. Since the two capacitances C_1 and C_2 are assumed to be equal to each other, we expect that the charge-sharing phenomenon will cause erroneous output values, as explained above, unless specific measures are taken to prevent it.

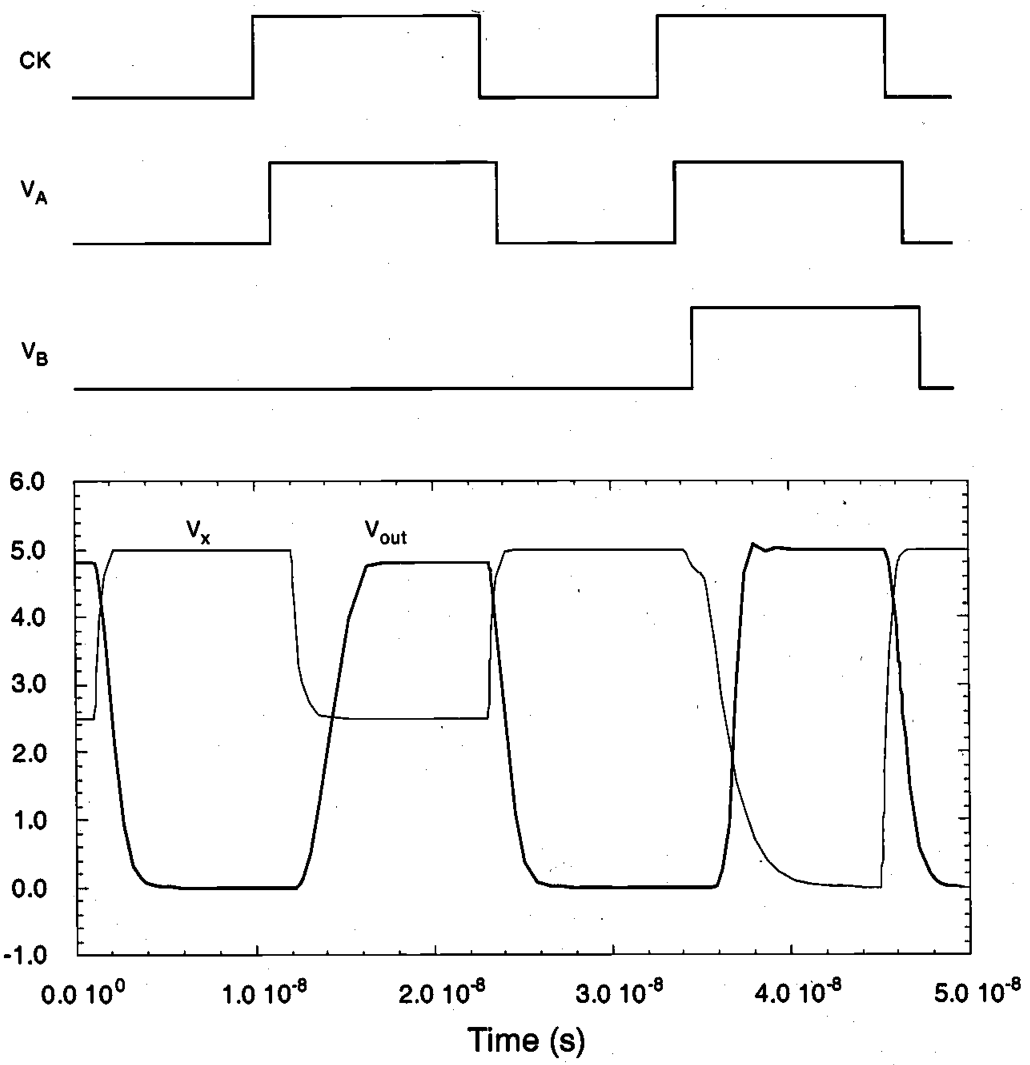


The SPICE-circuit input file of the domino CMOS NAND2 gate is listed below.

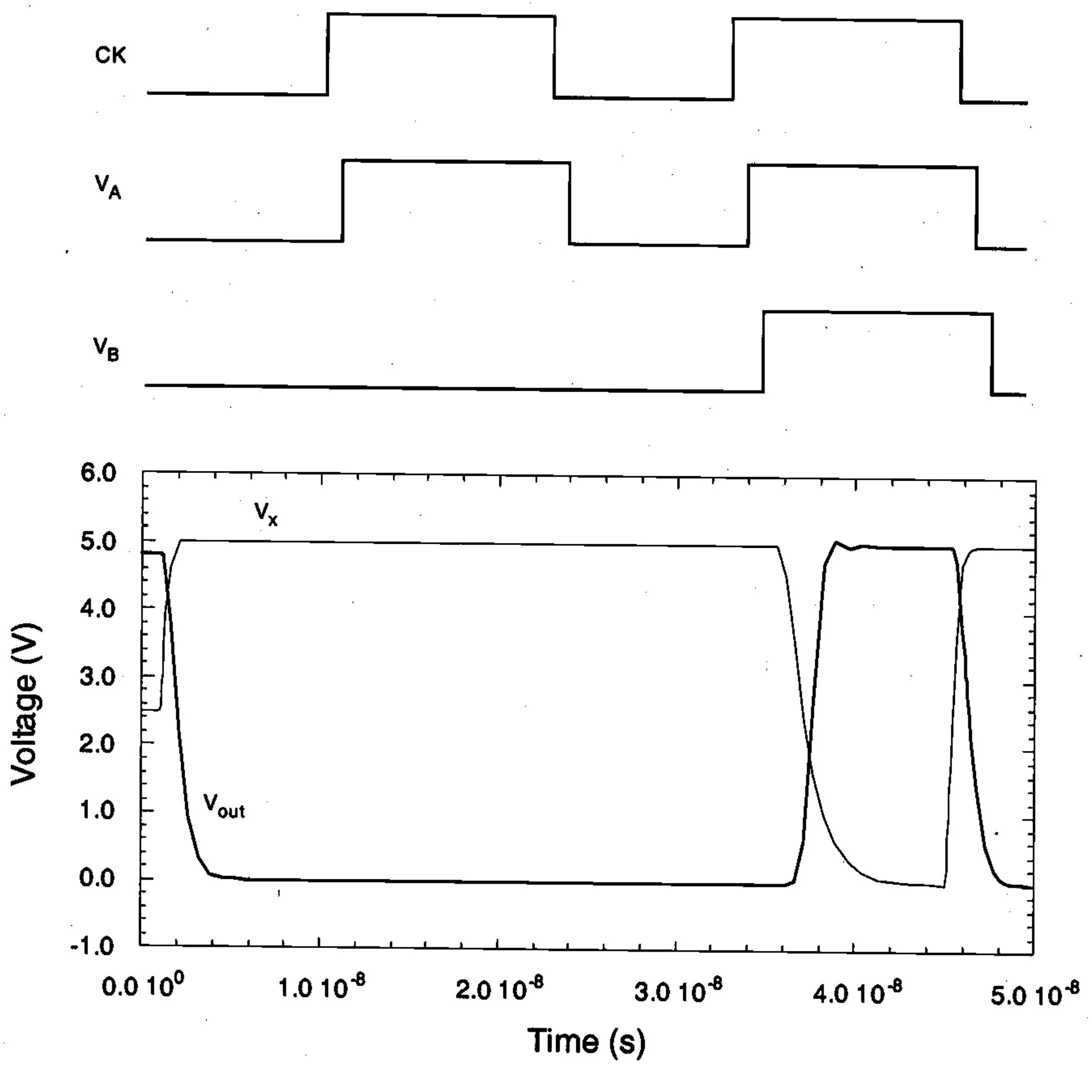
```
Domino CMOS with charge sharing
vdd 10 0 dc 5V
vin 1 0 dc pulse( 5 0 1ns 0.1ns 0.1ns 10ns 22ns)
vb 4 0 dc pulse ( 0 5 35ns 0.1ns 0.1ns 11ns 22n)
va 5 0 dc pulse( 0 5 12ns 0.1ns 0.1ns 11ns 22ns)
m1 2 1 0 0 mn l=5u w=10u
m2 3 4 2 0 mn l=5u w=10u
m3 6 5 3 0 mn l=5u w=10u
m4 6 1 10 10 mp l=5u w=25u
```

```
m6 7 6 0 0 mn l=5u w=10u
m7 7 6 10 10 mp l=5u w=100u
cload 6 0 0.05p
cs 3 0 0.05p
cout 7 0 0.1p
.model mn nmos vto=1 gamma=0.4 kp=2.5e-5
.model mp pmos vto=-1 gamma=0.4 kp=1.0e-5
.tran 0.1ns 50ns
.print tran v(1) v(6) v(7) v(3)
.end
```

The transient simulation of this circuit shows that the precharge node voltage V_x drops to about 2.5 V during the evaluation phase, due to charge sharing. As a result, the inverter output voltage erroneously switches to logic-high level during the first evaluation phase.



Now consider the case where an additional pMOS precharge transistor is connected between the power supply voltage V_{DD} and the intermediate node, as indicated in the figure above. Both pMOS transistors conduct during the precharge phase, and charge up the node capacitances to the same voltage level. Consequently, charge sharing can no longer cause a logic error at the output node. The simulation results *with* the additional pMOS precharge transistor are plotted below, showing that the output node voltage is pulled up to logic "1" only when both inputs are equal to logic "1."



It must be emphasized that there is a speed penalty for adding another pMOS precharge transistor to the circuit. Simulation results indicate that the pull-down delay of the node voltage V_x is actually increased by about 1 ns (approx. 25 percent) as a result of the additional parasitic capacitance which is due to the precharge device.

NORA CMOS Logic (NP-Domino Logic)

In domino CMOS logic gates, all logic operations are performed by the nMOS transistors acting as pull-down networks, while the role of pMOS transistors is limited to precharging the dynamic nodes. As an alternative and a complement to nMOS-based domino CMOS logic, we can construct dynamic logic stages using pMOS transistors as well. Consider the circuit shown in Fig. 9.37, with alternating nMOS and pMOS logic stages.

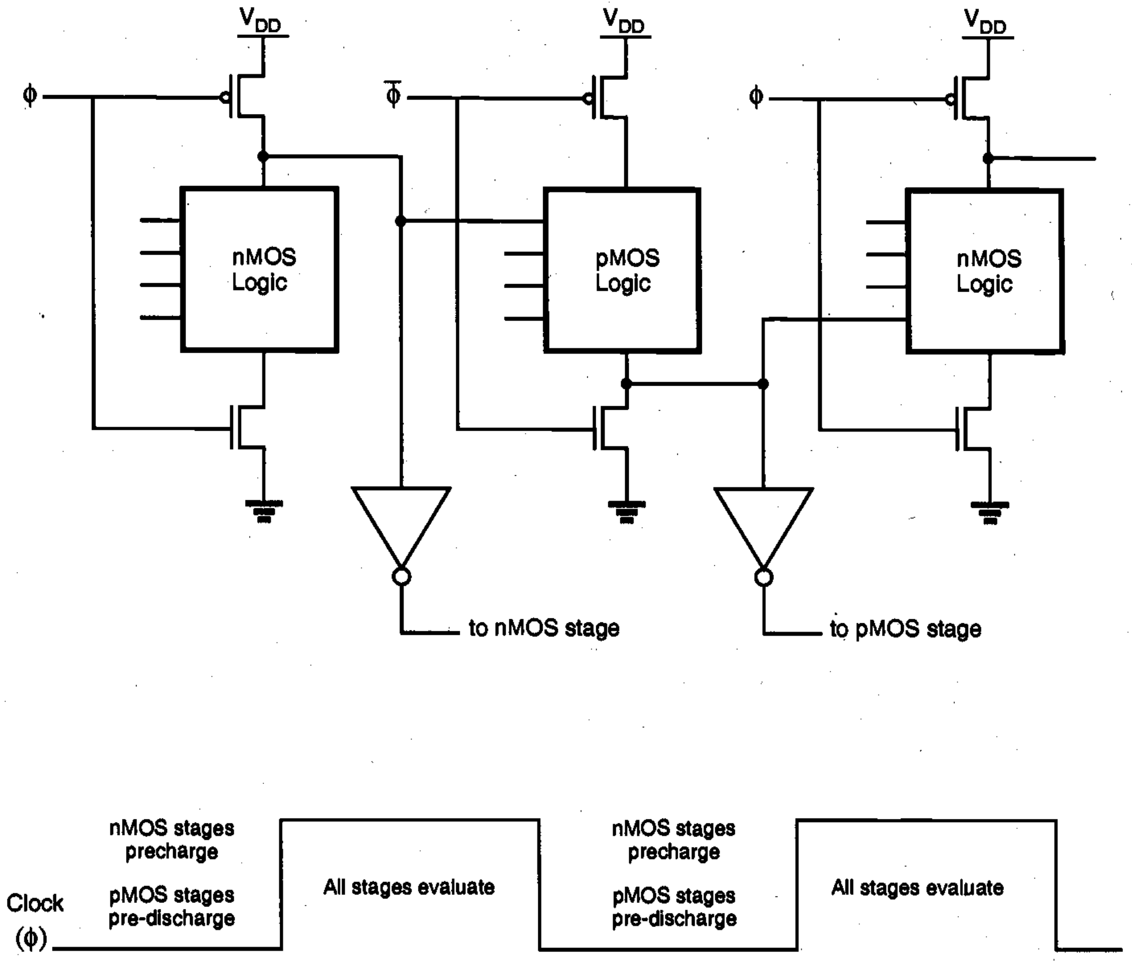


Figure 9.37. NORA CMOS logic consisting of alternating nMOS and pMOS stages, and the scheduling of precharge/evaluation phases.

Note that the precharge-and-evaluate timing of nMOS logic stages is accomplished by the clock signal ϕ, whereas the pMOS logic stages are controlled by the inverted clock signal, $\overline{\phi}$. The operation of the NORA CMOS circuit is as follows: When the clock signal is low, the output nodes of nMOS logic blocks are precharged to V_{DD} through the pMOS precharge transistors, whereas the output nodes of pMOS logic blocks are pre-discharged to 0 V through the nMOS discharge transistors, driven by $\overline{\phi}$. When the clock signal makes a low-to-high transition (note that the inverted clock signal $\overline{\phi}$ makes a high-to-low transition simultaneously), all cascaded nMOS and pMOS logic stages evaluate one after the other, much like the domino CMOS examined earlier. A simple NORA CMOS circuit example is shown in Fig. 9.38.

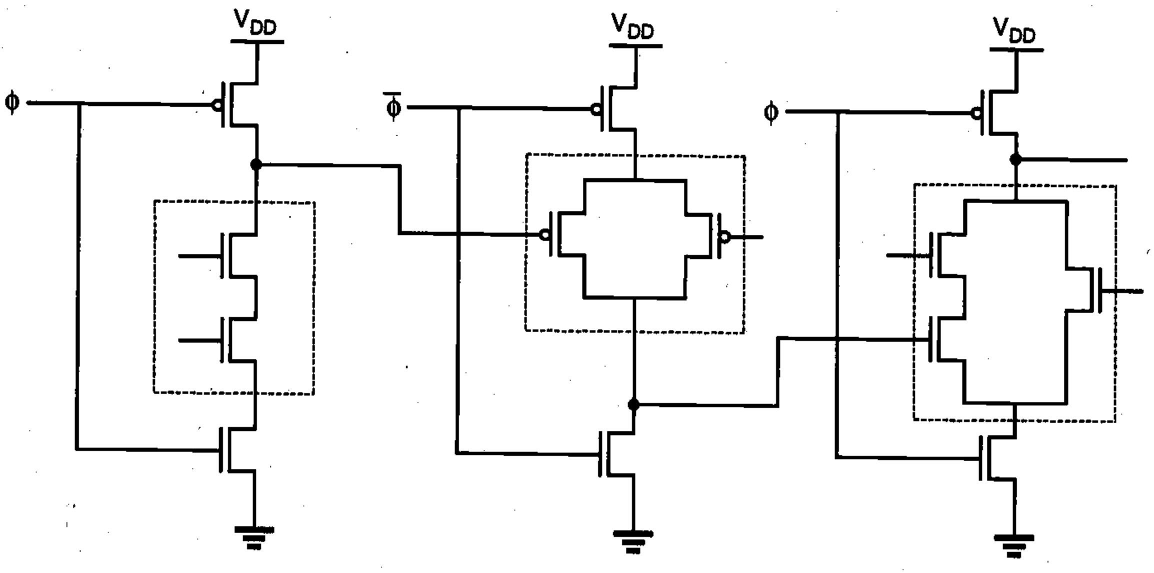


Figure 9.38. NORA CMOS logic circuit example.

The advantage of NORA CMOS logic is that a static CMOS inverter is not required at the output of every dynamic logic stage. Instead, direct coupling of logic blocks is feasible by alternating nMOS and pMOS logic blocks. NORA logic is also compatible with domino CMOS logic. Outputs of NORA nMOS logic blocks can be inverted, and then applied to the input of a domino CMOS block, which is also driven by the clock signal ϕ. Similarly, the buffered output of a domino CMOS stage can be applied directly to the input of a NORA nMOS stage.

The second important advantage of NORA CMOS logic is that it allows pipelined system architecture. Consider the circuit shown in Fig. 9.39(a), which consists of an nMOS-pMOS logic sequence similar to the one shown in Fig. 9.37, and a clocked CMOS (C^2MOS) output buffer. It can easily be seen that all stages of this circuit perform the precharge-discharge operation when the clock is low, and all stages of the circuit evaluate output levels when the clock is high. Therefore, we will call this circuit a ϕ-section, meaning that evaluation occurs during active ϕ.

Now consider the circuit shown in Fig. 9.39(b), which is essentially the same circuit shown in Fig. 9.39(a), with the only difference being that the signals ϕ and $\overline{\phi}$ have been interchanged. In this circuit, all logic stages perform the precharge-discharge operation when the clock is high, and all stages evaluate output levels when the clock is low. Therefore, we will call this circuit a $\overline{\phi}$-section, meaning that evaluation occurs during active $\overline{\phi}$.

A pipelined system can be constructed by simply cascading alternating ϕ- and $\overline{\phi}$-sections, as shown in Fig. 9.39(c). Note that each of the sections may consist of several logic stages, and that all logic stages in one section are evaluated during the same clock cycle. When the clock is low, the ϕ-sections in the pipelined system undergo the precharge cycle, while the $\overline{\phi}$-sections undergo evaluation. When the clock signal changes from low to high, the ϕ-sections start the evaluation cycle, while the $\overline{\phi}$-sections undergo precharge. Thus, consecutive sets of input data can be processed in alternating sections of the pipelined system.

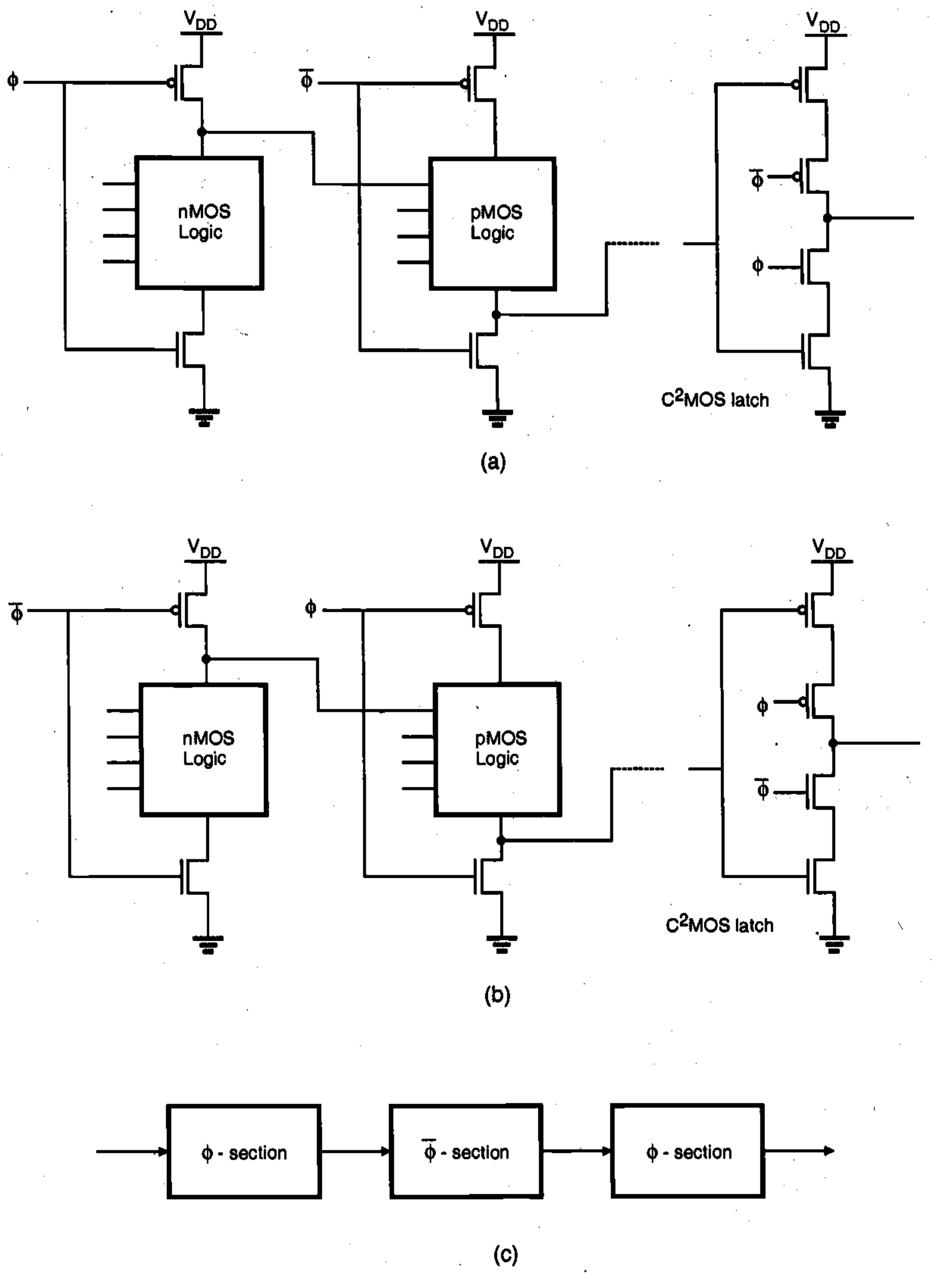


Figure 9.39. (a) NORA CMOS ϕ-section; evaluation occurs during $\phi = 1$. (b) NORA CMOS $\bar{\phi}$-section; evaluation occurs during $\phi = 0$. (c) A pipelined NORA CMOS system.

As in all dynamic CMOS structures, NORA CMOS logic gates also suffer from charge sharing and leakage. To overcome the dynamic charge sharing and soft- node leakage problems in NORA CMOS structures, a circuit technique called Zipper CMOS can be used.

The basic circuit architecture of Zipper CMOS is essentially identical to NORA CMOS, with the exception of the clock signals. The Zipper CMOS clock scheme requires the generation of slightly different clock signals for the precharge (discharge) transistors and for the pull-down (pull-up) transistors. In particular, the clock signals which drive the pMOS precharge and nMOS discharge transistors allow these transistors to remain in weak conduction or near cut-off during the evaluation phase, thus compensating for the charge leakage and charge-sharing problems. The generalized circuit diagram and the clock signals of the Zipper CMOS architecture are shown in Fig. 9.40.

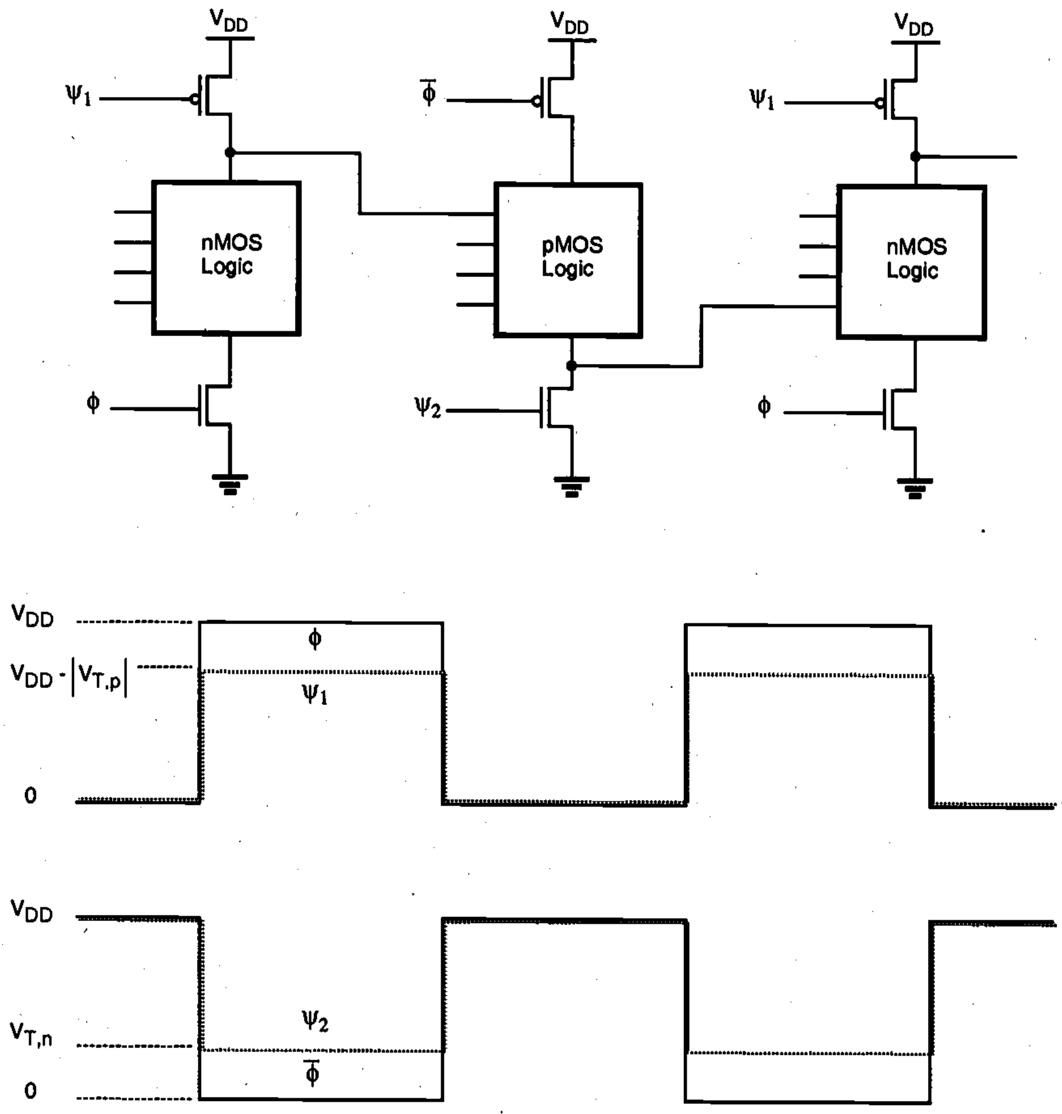


Figure 9.40. General circuit structure and the clock signals of Zipper CMOS.

The dynamic circuit technique to be presented in the following is distinctly different from the NORA CMOS circuit architecture in that it uses only one clock signal which is never inverted. Since the inverted clock signal ϕ is not used anywhere in the system, no clock skew problem exists. Consequently, higher clock frequencies can be reached for dynamic pipelined operation.

Consider the circuit diagram shown in Fig. 9.41. The circuit consists of alternating stages called n-blocks and p-blocks, and each block is being driven by the same clock signal ϕ. An n-block is constructed by cascading a dynamic nMOS stage and a dynamic latch, while a p-block is constructed by cascading a dynamic pMOS stage and a dynamic latch.

When the clock signal is low, the output node of the n-block is being precharged to V_{DD} by the pMOS precharge transistor. When the clock signal switches from low to high, the logic stage output is evaluated and the output latch generates a valid output level. On the other hand, we can see by inspection that the p-block pre-discharges when the clock is high, and evaluates when the clock is low. This means that a cascade-connection of alternating n-block and p-block circuits as shown in Fig. 9.41 will allow pipelined operation using a single clock signal. Compared to NORA CMOS, we need two extra transistors per stage, but the ability to operate with a true single-phase clock signal offers very attractive possibilities from the system-design point of view.

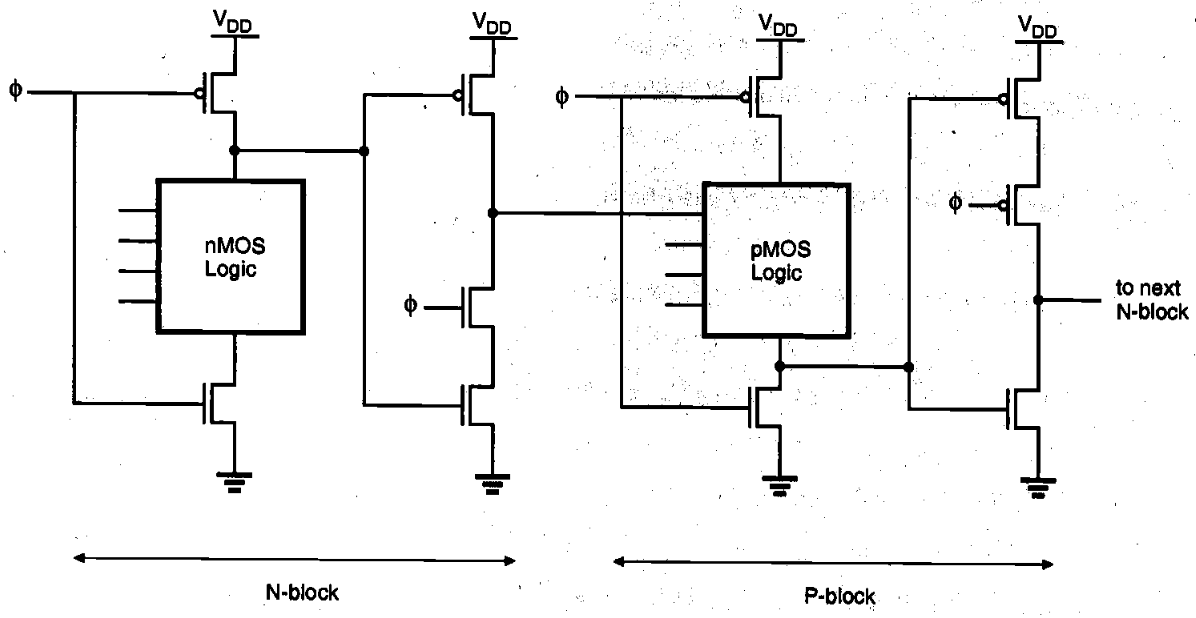


Figure 9.41. A pipelined true single-phase clock CMOS system.

Figure 9.42 shows the circuit diagram of a rising edge-triggered D-type flip-flop (DFF) which is constructed with the TSPC principle. The circuit consists of eleven transistors, in four simple stages. When the clock signal is low, the first stage acts as a transparent latch to receive the input signal, while the output node of the second stage is being precharged. During this cycle, the third and fourth stages simply keep the previous output state. When the clock signal switches from low to high, the first stage ceases to be transparent and the second stage starts evaluation. At the same time, the third stage becomes transparent and transmits the sampled value to the output. Note that the final stage (inverter) is only used to obtain the non-inverted output level.

The mask layout of this circuit (using 0.8 μm CMOS technology design rules) is shown in Fig. 9.43. The device dimensions are $(W/L)_n$ = (2 μm/0.8 μm) for nMOS transistors, and $(W/L)_p$ = (5.6 μm/0.8 μm) for pMOS transistors. The layout parasitics were determined through circuit extraction, and the extracted circuit file was simulated with SPICE to determine its performance. The simulated input and output waveforms of the TSPC-based DFF circuit are given in Fig. 9.44. It can be seen that the circuit is capable of operating with a clock frequency of 500 MHz. With their relatively simple design, low transistor count and high speed, TSPC-based circuits offer a very favorable alternative to conventional CMOS circuits, especially in high-performance designs.

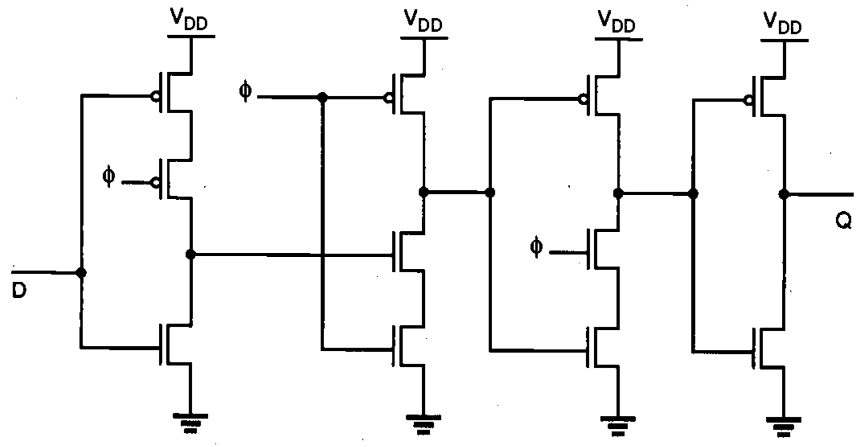


Figure 9.42. Circuit diagram of a TSPC-based rising edge-triggered DFF.

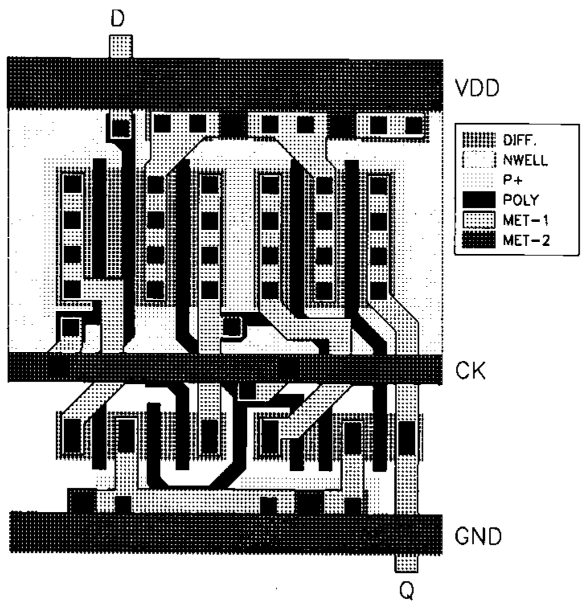


Figure 9.43. Mask layout of the TSPC-based DFF circuit.