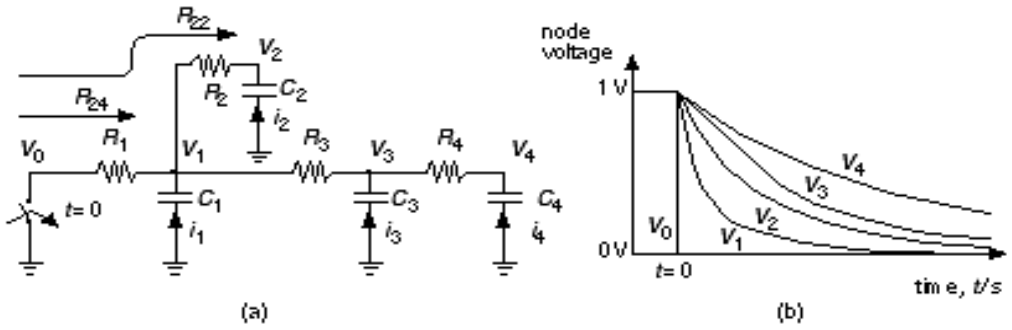
UNIT II COMBINATIONAL LOGIC CIRCUITS

Examples of Combinational Logic Design, Elmore"s constant, Pass transistor Logic, Transmission gates, static and dynamic CMOS design, Power dissipation – Low power design principles.

**Elmore's Constant:**

Aims at analysis of RC networks to examine the delays due to interconnects.




RC tree —representing a net with a fanout of two

The waveforms as a result of closing the switch at t = 0

$$V_i(t) = \exp(-t/\tau_{Di});\ \tau_{Di} = \sum_{k=1}^{n} R_{ki} C_k$$

$$V_i(t) = \exp(-t/\tau_{Di});\ \tau_{Di} = \sum_{k=1}^{n} R_{ki} C_k$$

$\tau_{Di}$ - Elmore delay different for each node

**2.1. Pass Transistors and Transmission Gates:**

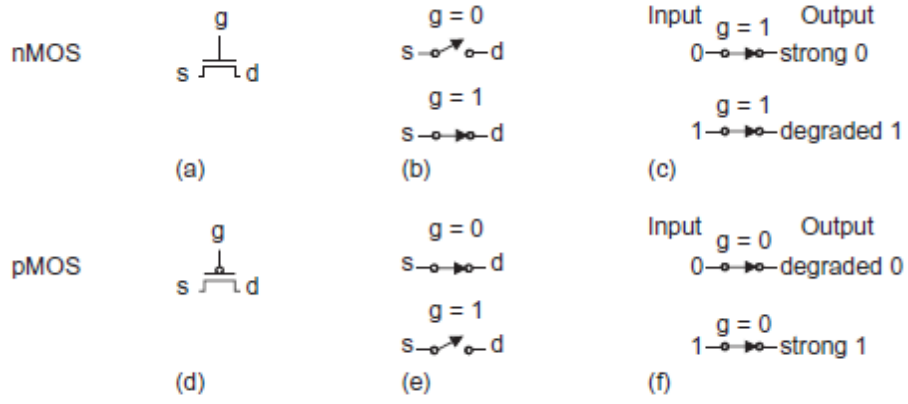



FIGURE 1.20 Pass transistor strong and degraded outputs

The strength of a signal is measured by how closely it approximates an ideal voltage source. In general, the stronger a signal, the more current it can source or sink. The power supplies ,or rails, (VDD and GND) are the source of the strongest 1s and 0s.

An nMOS transistor is an almost perfect switch when passing a 0 and thus we say it passes a strong 0. However, the nMOS transistor is imperfect at passing a 1. The high voltage level is somewhat less than VDD. We say it passes a degraded or weak 1. A pMOS transistor again has the opposite behavior, passing strong 1s but degraded 0s. The transistor symbols and behaviors are summarized in Figure 1.20 with g, s, and d indicating gate, source, and drain.

When an nMOS or pMOS is used alone as an imperfect switch, we sometimes call it a pass transistor. By combining an nMOS and a pMOS transistor in parallel (Figure 1.21(a)), we obtain a switch that turns on when a 1 is applied to g (Figure 1.21(b)) in which 0s and 1s are both passed in an acceptable fashion (Figure 1.21(c)). We term this a transmission gate or pass gate. In a circuit where only a 0 or a 1 has to be passed, the appropriate transistor (n or p) can be deleted, reverting to a single nMOS or pMOS device.

Note that both the control input and its complement are required by the transmission gate. This is called double rail logic. Some circuit symbols for the transmission gate are shown in Figure 1.21(d).1 None are easier to draw than the simple schematic, so we will use the schematic version to represent a transmission gate in this book.

In all of our examples so far, the inputs drive the gate terminals of nMOS transistors in the pull-down network and pMOS transistors in the complementary pull-up network. Thus, the nMOS transistors only need to pass 0s and the pMOS only pass 1s, so the output is always strongly driven and the levels are never degraded. This is called a fully restored logic gate and simplifies circuit design considerably. In contrast to other forms of logic, where the pull-up and pull-down switch networks have to be ratioed in some manner, static CMOS gates operate correctly independently of the physical sizes of the transistors. Moreover, there is never a path through 'ON' transistors from the 1 to the 0 supplies for any combination of inputs (in contrast to single-channel MOS, GaAs technologies, or bipolar). As we will find in subsequent chapters, this is the basis for the low static power dissipation in CMOS.

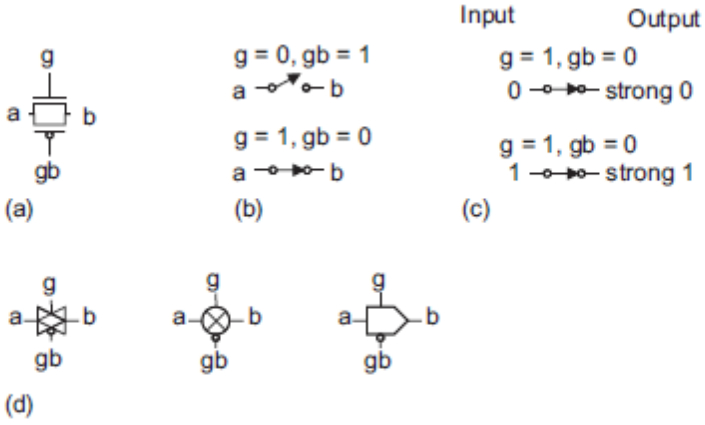



**FIGURE 1.21** Transmission gate

A consequence of the design of static CMOS gates is that they must be inverting. The nMOS pull-down network turns ON when inputs are 1, leading to 0 at the output. We might be tempted to turn the transistors upside down to build a noninverting gate. For example, Figure 1.22 shows a noninverting buffer. Unfortunately, now both the nMOS and pMOS transistors produce degraded outputs, so the technique should be avoided. Instead, we can build noninverting functions from multiple stages of inverting gates. Figure 1.23 shows several ways to build a 4-input AND gate from two levels of inverting static CMOS gates. Each design has different speed, size, and power trade-offs.

Similarly, the compound gate could be built with two AND gates, an OR gate, and an inverter. The AND and OR gates in turn could be constructed from NAND/NOR gates and inverters, as shown in Figure 1.24, using a total of 20 transistors. Good CMOS logic designers exploit the efficiencies of compound gates rather than using large numbers of AND/OR gates.

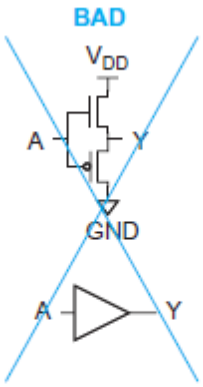



FIGURE 1.22
Bad noninverting buffer

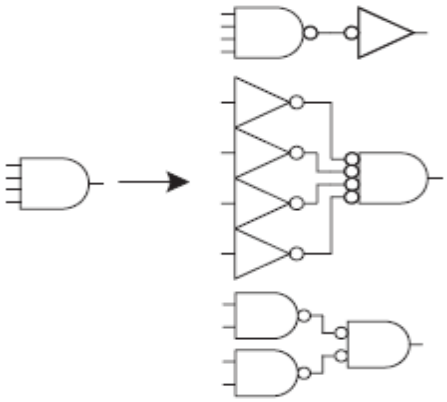


FIGURE 1.23 Various implementations of a CMOS 4-input AND gate

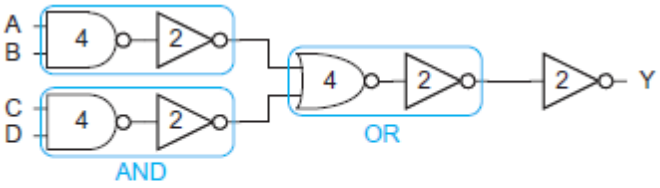



FIGURE 1.24 Inefficient discrete gate implementation of AOI22 with transistor counts indicated

**.2.2.Static CMOS:**

Static CMOS circuits with complementary nMOS pulldown and pMOS pullup networks are used for the vast majority of logic gates in integrated circuits. They have good noise margins, and are fast, low power, insensitive to device variations, easy to design, widely supported by CAD tools, and readily available in standard cell libraries.

Designers accustomed to AND and OR functions must learn to think in terms of NAND and NOR to take advantage of static CMOS. In manual circuit design, this is often done through bubble pushing. Compound gates are particularly useful to perform complex functions with relatively low logical efforts.

2.2.1.1 Bubble Pushing: CMOS stages are inherently inverting, So AND and OR functions must be built from NAND and NOR gates. DeMorgan's law helps with this conversion:

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

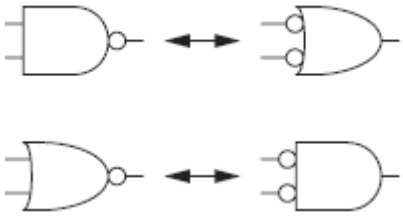
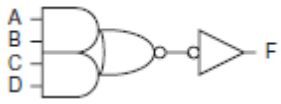


Fig 1.1Bubble pushing with DeMorgan's law

These relations are illustrated graphically in Figure 1.1 A NAND gate is equivalent to an OR of inverted inputs. A NOR gate is equivalent to an AND of inverted inputs. The same relationship applies to gates with more inputs. Switching between these representations is easy to do on a whiteboard and is often called bubble pushing.

2.2.1.2.Compound Gates: Satic CMOS also efficiently handles compound gates computing various inverting combinations of AND/OR functions in a single stage. The function F=AB+CD can be computed with an AND-ORINVERT-22 (AOI22) gate and an inverter, as shown in Figure 1.2 and 1.3



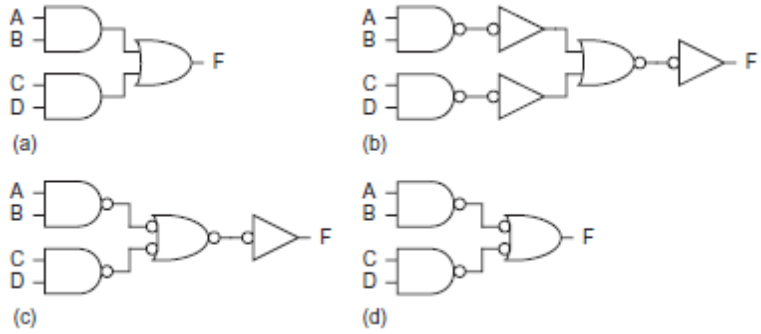




Fig 1.2 Logic using AOI22 gate

Fig 1.3 Bubble pushing to convert ANDs and ORs to NANDs and NORs

2.2.1.3. Input Ordering Delay Effect: The logical effort and parasitic delay of different gate inputs are often different. NANDs and NORs, are nominally symmetric but actually have slightly different logical effort and parasitic delays for the different inputs.Figure 1.4 shows a 2-input NAND gate annotated with diffusion parasitics. Consider the falling output transition occurring when one input held a stable 1 value and the other rises from 0 to 1. If input B rises last, node x will initially be at VDD – Vt ≈ VDD because it was pulled up through the nMOS transistor on input A. The Elmore delay is (R/2)(2C) +R(6C) =7RC=2.33 τ.On the other hand, if input A rises last, node x will initially be at 0 V because it was discharged through the nMOS transistor on input B. No charge must be delivered to node x, so the Elmore delay is simply R(6C)=6RC =2 τ.

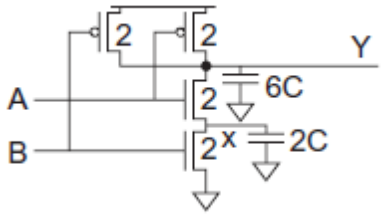



Fig 1.4 NAND gate delay estimation

2.2.1.4 Asymmetric Gates 9.2.1.4 Asymmetric Gates:

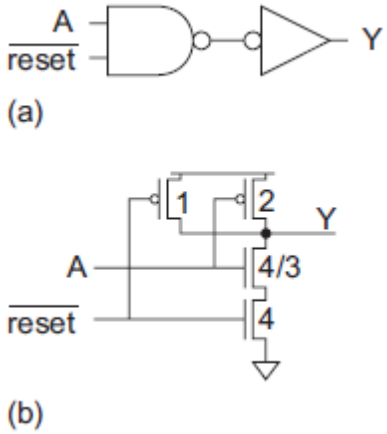



Fig 1.5 Resettable buffer optimized for data input

When one input is far less critical than another, even nominally symmetric gates can be made asymmetric to favor the late input at the expense of the early one. In a series network, this involves connecting the early input to the outer transistor and making the transistor wider so that it offers less series resistance when the critical input arrives. In a parallel network, the early input is connected to a narrower transistor to reduce the parasitic capacitance.

In Figure 1.5(a). Under ordinary conditions, the path acts as a buffer between A and Y. When reset is asserted, the path forces the output low. If reset only occurs under exceptional circumstances and can take place slowly, the circuit should be optimized for input-to-output delay at the expense of reset. This can be done with the asymmetric NAND gate in Figure 1.5(b).

2.2.1.5. Skewed Gates: One input transition is more important than the other skewed gates are used. We defined HI-skew gates to favor the rising output transition and LO-skew gates to favor the falling output transition. This favoring can be done by decreasing the size of the noncritical transistor. The logical efforts for the rising (up) and falling (down) transitions are called gu and gd, . Figure 1.6(a) shows HI-skew inverter. This maintains the same effective resistance for the critical transition while reducing the input capacitance relative to the unskewed inverter of Figure 1.6(b), thus reducing the logical effort on that critical transition to gu =2.5/3 =5/6. Of course, the improvement comes at the expense of the effort on the noncritical transition. The logical effort for the falling transition is estimated by comparing the inverter to a smaller unskewed inverter with equal pulldown current, shown in Figure 1.6(c), giving a logical effort of gd=2.5/1.5 =5/3.Figure 1.7 shows catalogs of HI skew and LO-skew gates with a skew factor of two.

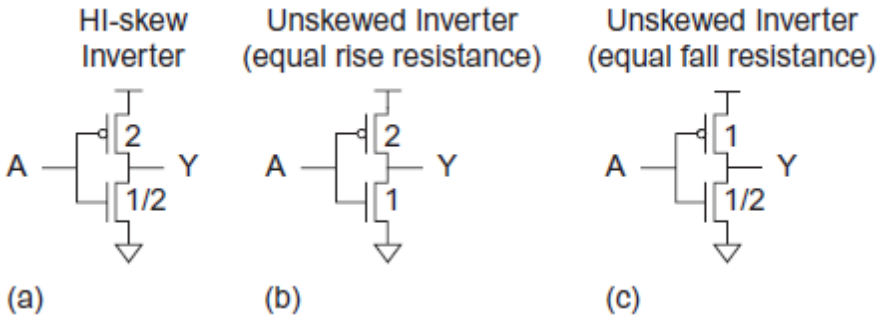



Fig 1.6 Logical effort calculation for HI-skew inverter

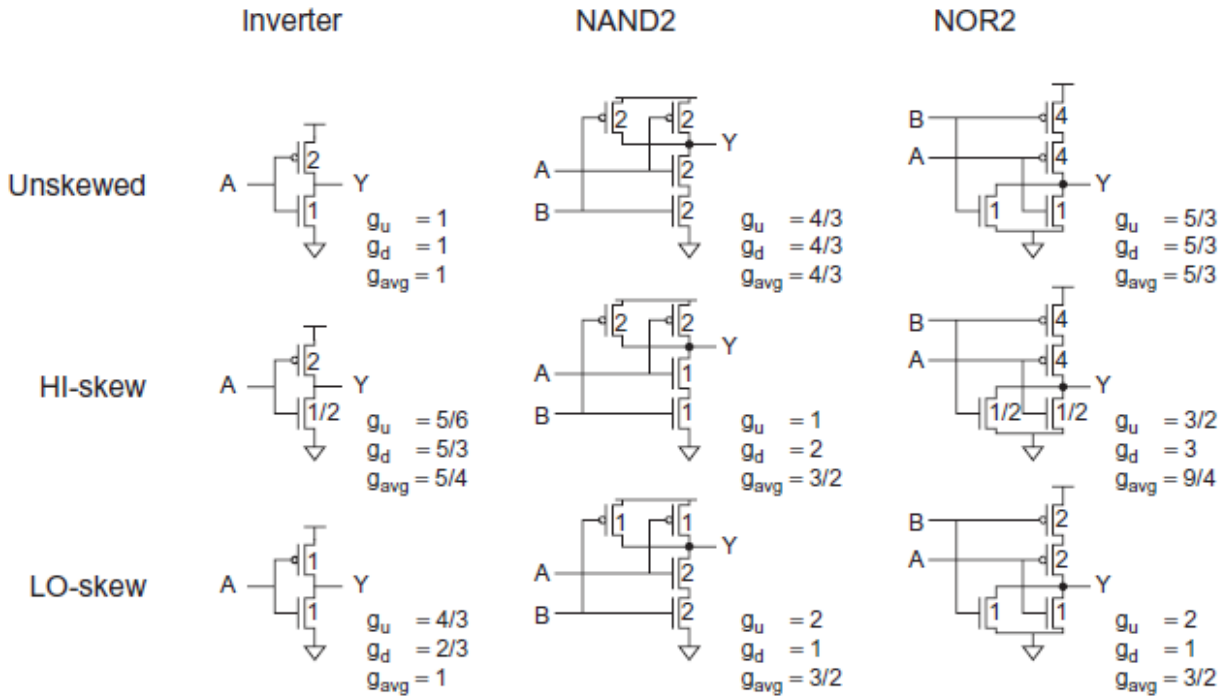



FIGURE 1.7 Catalog of skewed gates

2.2.1.6. P/N Ratios : In Figure 9.10 that the average logical effort of the LO-skew NOR2 is actually better than that of the unskewed gate. The pMOS transistors in the unskewed gate are enormous in order to provide equal rise delay. They contribute input capacitance for both transitions, while only helping the rising delay. By accepting a slower rise delay, the pMOS transistors can be downsized to reduce input capacitance and average delay significantly. In general, what is the best P/N ratio for logic gates (i.e., the ratio of pMOS to nMOS transistor width)? The ratio giving lowest average delay is the square root of the ratio that gives equal rise and fall delays. For processes with a mobility ratio of βn/βp =2 as we have generally been assuming, the best ratios are shown in Figure 1.8.

Reducing the pMOS size from 2 to $\sqrt{2}$ ≈1.4 for the inverter gives the theoretical fastest average delay, but this delay improvement is only 3%. However, this significantly reduces the pMOS transistor area. It also reduces input capacitance, which in turn reduces power consumption. Unfortunately, it leads to unequal delay between the outputs.

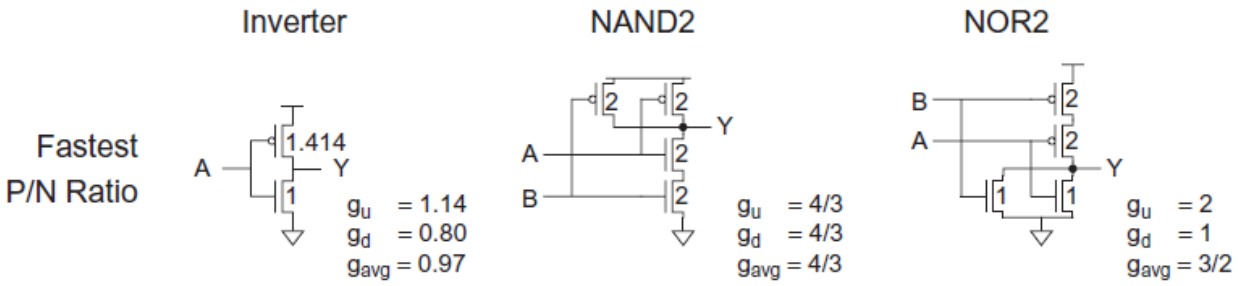



Fig 1.8 Gates with P/N ratios giving least delay

2.2.2. Ratioed Circuits: The ratioed gate consists of an nMOS pulldown network and some pullup device called the static load. When the pulldown network is OFF, the static load pulls the output to 1. When the pulldown network turns ON, the output pulls down to 0.

2.2.2.1 Pseudo-nMOS : Figure 1.9 shows several pseudo-nMOS logic gates. The pulldown network is like that of an ordinary static gate, but the pullup network has been replaced with a single pMOS transistor that is grounded so it is always ON. The pMOS transistor widths are selected to be about 1/4 the strength (i.e., 1/2 the effective width) of the nMOS pulldown network as a compromise between noise margin and speed.

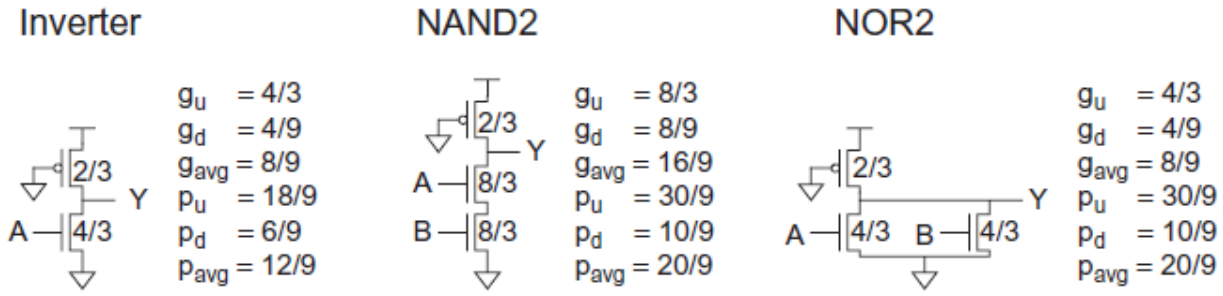



Fig 1.9 Pseudo-nMOS logic gates

2.2.2.2 Ganged CMOS : Figure 1.10 illustrates pairs of CMOS inverters ganged together. The truth table is given in Table 1.1, showing that the pair compute the NOR function. Such a circuit is sometimes called a symmetric 2 NOR , or more generally, ganged CMOS. When one input is 0 and the other 1, the gate can be viewed as a pseudo-nMOS circuit with appropriate ratio constraints. When both inputs are 0, both pMOS transistors turn on in parallel, pulling the output high faster than they would in an ordinary pseudo Nmos gate. Moreover, when both inputs are 1, both pMOS transistors turn OFF, saving static power dissipation. As in pseudo-nMOS, the transistors are sized so the pMOS are about 1/4 the strength of the nMOS and the pulldown current matches that of a unit inverter. Hence, the symmetric NOR achieves both better performance and lower power dissipation than a 2-input pseudo-nMOS NOR.

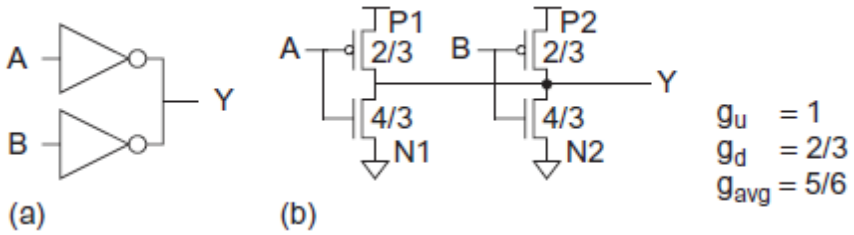



Fig 1.10 Symmetric 2-input NOR gate

TABLE 1.1 Operation of symmetric NOR

| A | B | N1 | P1 | N2 | P2 | Y |
|---|---|---|---|---|---|---|
| 0 | 0 | OFF | ON | OFF | ON | 1 |
| 0 | 1 | OFF | ON | ON | OFF | ~ 0 |
| 1 | 0 | ON | OFF | OFF | ON | ~ 0 |
| 1 | 1 | ON | OFF | ON | OFF | 0 |

Disadvantages of Ratioed Circuits:

1.Reduced Capacitance

2.Slow Rising Transitions

3.Contention on the falling Transition

4.Static Power Dissipation

5.Non-Zero $V_{0L}$

2.2.2.3.Cascode Voltage Switch Logic:

Cascode Voltage Switch Logic (CVSL3) seeks the benefits of ratioed circuits without the static power consumption. It uses both true and complementary input signals and computes both true and complementary outputs using a pair of nMOS pulldown networks, as shown in Figure 1.11. The pulldown network f implements the logic function as in a static CMOS gate, while $\hat{f}$ uses inverted inputs feeding transistors arranged in the conduction complement. For any given input pattern, one of the pulldown networks will be ON and the other OFF. The pulldown network that is ON will pull that output low. This low output turns ON the pMOS transistor to pull the opposite output high. When the opposite output rises, the other pMOS transistor turns OFF so no static power dissipation occurs. CVSL has a potential speed advantage because all of the logic is performed with nMOS transistors, thus reducing the input capacitance. the CVSL gate requires both the low- and high-going transitions, adding more delay. Contention current during the switching period also increases power consumption.

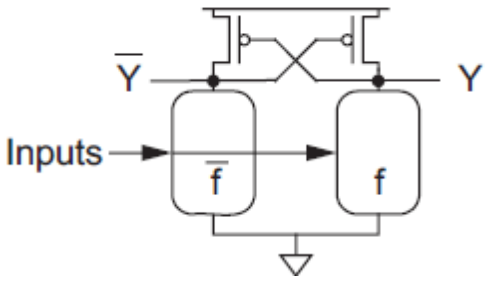



Fig 1.11 CVSL gates

Advantages of Static CMOS Design:
1.Low sensitivity to noise
2.Good Performance
3.Low Power Consumption
4.Large fan-in
5.Similar circuit Topology

**2.3. Dynamic Circuits:**

The drawbacks of ratioed circuits include slow rising transitions, contention on the falling transitions, static power dissipation, and a nonzero VOL. Dynamic circuits is used to overcome these drawbacks by using a clocked pullup transistor rather than a pMOS that is always ON. Figure 1.12 compares (a) static CMOS, (b) pseudo-nMOS, and (c) dynamic inverters.

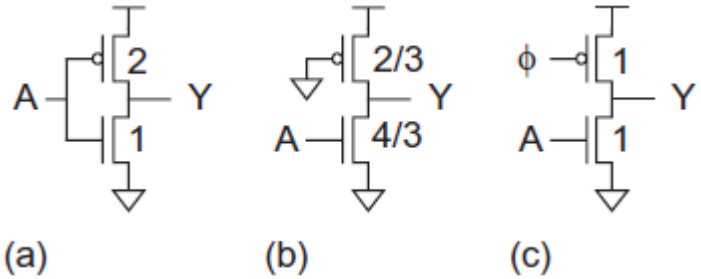



Fig 1.12 Comparison of (a) static CMOS, (b) pseudo-nMOS, and (c) dynamic inverters

Dynamic circuit operation is divided into two modes, as shown in Figure 1.13.During precharge, the clock Φ is 0, so the clocked pMOS is ON and initializes the output Y high. During evaluation, the clock is 1 and the clocked pMOS turns OFF. The output may remain high or may be discharged low through the pulldown network. Dynamic circuits are the fastest commonly used circuit family because they have lower input capacitance and no contention during switching. They also have zero static power dissipation. However, they require careful clocking, consume significant dynamic power, and are sensitive to noise during evaluation.

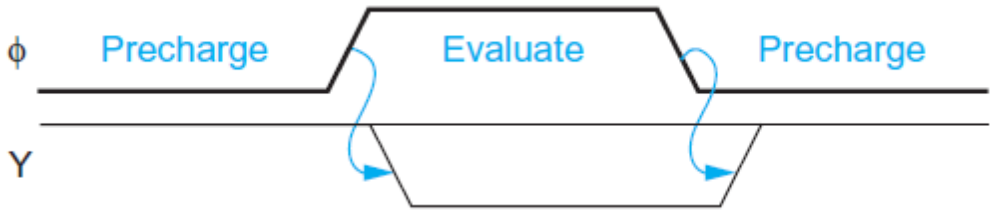



Fig 1.13 Precharge and evaluation of dynamic gates

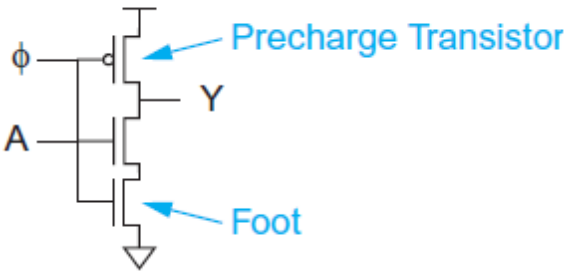



Fig 1.14 Footed dynamic inverter

In Figure 1.14(c), if the input A is 1 during precharge, contention will take place because both the pMOS and nMOS transistors will be ON. When the input cannot be guaranteed to be 0 during precharge, an extra clocked evaluation transistor can be added to the bottom of the nMOS stack to avoid contention as shown in Figure 1.14. The extra transistor is sometimes called a foot.

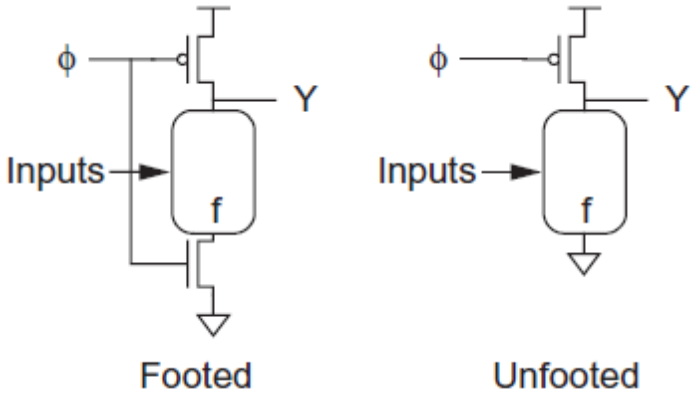



Fig 1.15 Generalized footed and unfooted dynamic gates

Figure 1.15 shows generic footed and unfooted gates. Figure 1.16 estimates the falling logical effort of both footed and unfooted dynamic gates. Precharge occurs while the gate is idle and often may take place more slowly. Therefore, the precharge transistor width is chosen for twice unit resistance. This reduces the capacitive load on the clock and the parasitic capacitance at the expense of greater rising delays. We see that the logical efforts are very low. Footed gates have higher logical effort than their unfooted counterparts but are still an improvement over static logic. In practice, the logical effort of footed gates is better than predicted because velocity saturation means series nMOS transistors have less resistance than we have estimated. Moreover, logical efforts are also slightly better than predicted because there is no contention between nMOS and pMOS transistors during the input transition. The size of the foot can be increased relative to the other nMOS transistors to reduce logical effort of the other inputs at the expense of greater clock loading. Like pseudo-nMOS gates, dynamic gates are particularly well suited to wide NOR functions or multiplexers because the logical effort is independent of the number of inputs. Of course, the parasitic delay does increase with the number of inputs because there is more diffusion capacitance on the output node. Characterizing the logical effort and parasitic delay of dynamic gates is tricky because the output tends to fall much faster than the input rises, leading to potentially misleading dependence of propagation delay on fanout.

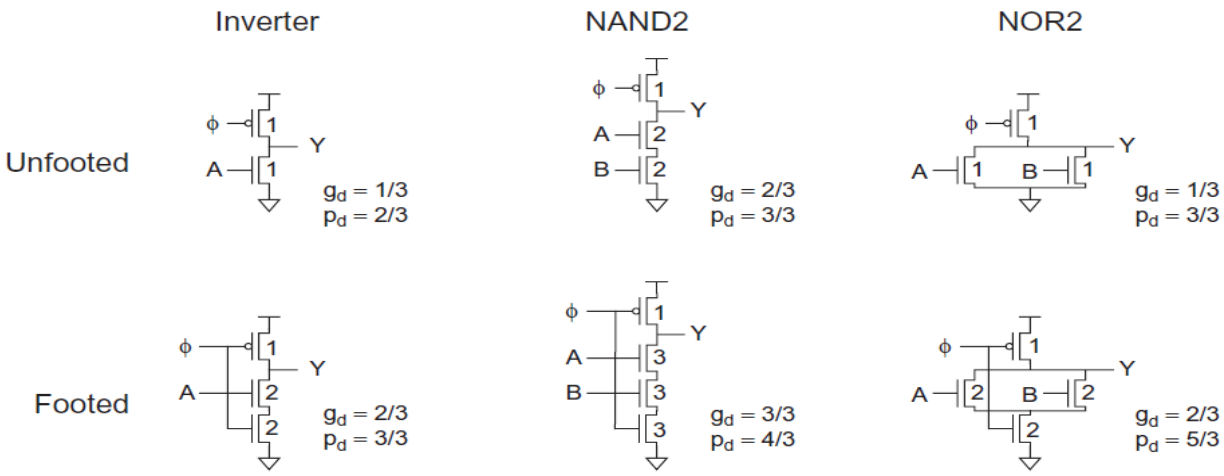



Fig 1.16 Catalog of dynamic gates

A fundamental difficulty with dynamic circuits is the monotonicity requirement. While a dynamic gate is in evaluation, the inputs must be monotonically rising. That is, the input can start LOW and remain LOW, start LOW and rise HIGH, start HIGH and remain HIGH, but not start HIGH and fall LOW. Figure 1.17 shows waveforms for a footed dynamic inverter in which the input violates monotonicity. During precharge, the output is pulled HIGH. When the clock rises, the input is HIGH so the output is discharged LOW through the pulldown network, as you would want to have happen in an inverter. The input later falls LOW, turning off the pulldown network. However, the precharge transistor is also OFF so the output floats, staying LOW rather than rising as it would in a normal inverter. The output will remain low until the next precharge step. In summary, the inputs must be monotonically rising for the dynamic gate to compute the correct function.

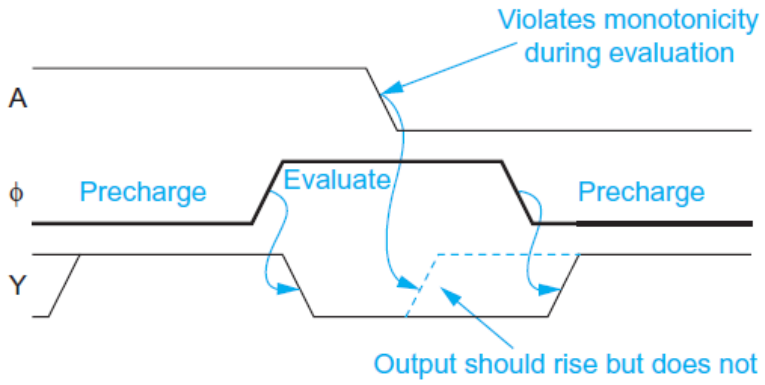



Fig 1.17 Monotonicity problem

2.3.1 Domino Logic: The monotonicity problem can be solved by placing a static CMOS inverter between dynamic gates, as shown in Figure 1.18(a). This converts the monotonically falling output into a monotonically rising signal suitable for the next gate, as shown in Figure 1.18(b). The dynamic-static pair together is called a domino gate A single clock can be used to precharge and evaluate all the logic gates within the chain. The dynamic output is monotonically falling during evaluation, so the static inverter output is monotonically rising.

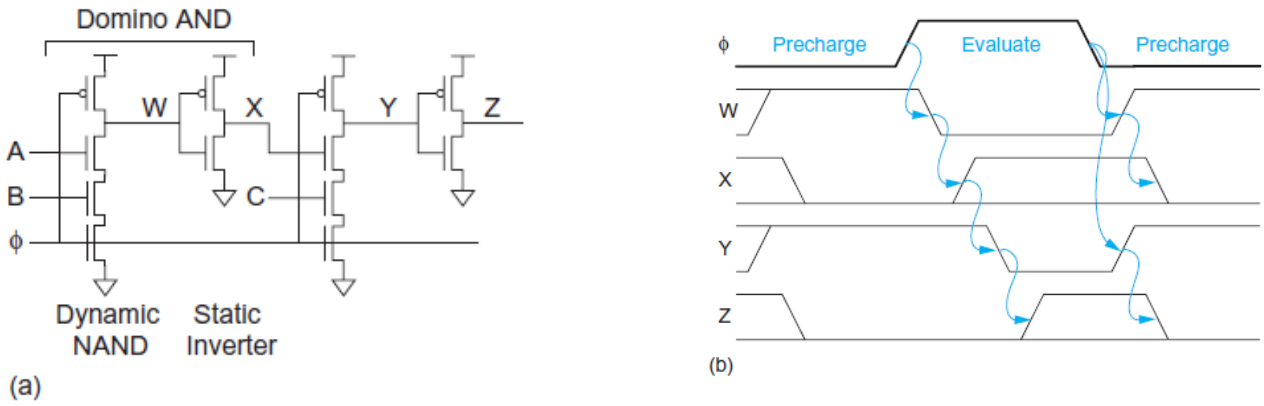



Fig 1.18 Domino gates

2.3.2 Dual-Rail Domino Logic: Dual-rail domino gates encode each signal with a pair of wires. The input and output signal pairs are denoted with _h and _l, respectively. Table 1.2 summarizes the encoding. The _h wire is asserted to indicate that the output of the gate is “high” or 1. The _l wire is asserted to indicate that the output of the gate is “low” or 0.When the gate is precharged, neither _h nor _l is asserted. The pair of lines should never be both asserted simultaneously during correct operation.

TABLE 1.2 Dual-rail domino signal encoding

| sig_h | sig_l | Meaning |
|---|---|---|
| 0 | 0 | Precharged |
| 0 | 1 | '0' |
| 1 | 0 | '1' |
| 1 | 1 | Invalid |

Dual-rail domino gates accept both true and complementary inputs and compute both true and complementary outputs, as shown in Figure 1.19(a). Observe that this is identical to static CVSL circuits.

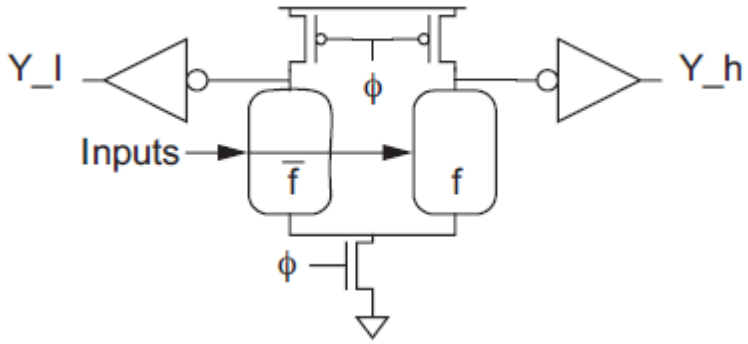



Fig 1.19 Dual-rail domino gates

Disadvantages of Duel rail domino logic circuits:

1.Require more area
2.Require more wiring
3.Require more power
4.Has less efficiency

2.3.3 Keepers : Dynamic circuits also suffer from charge leakage on the dynamic node. If a dynamic node is precharged high and then left floating, the voltage on the dynamic node will drift over time due to subthreshold, gate, and junction leakage. The time constants tend to be in the millisecond to nanosecond range, depending on process and temperature. This problem is analogous to leakage in dynamic RAMs. Moreover, dynamic circuits have poor input noise margins. If the input rises above $V_t$ while the gate is in evaluation, the input transistors will turn on weakly and can incorrectly discharge the output. Both leakage and noise margin problems
can be addressed by adding a keeper circuit.

Figure 1.20 shows a conventional keeper on a domino buffer. The keeper is a weak transistor that holds, or staticizes, the output at the correct level when it would otherwise float. When the dynamic node X is high, the output Y is low and the keeper is ON to prevent X from floating. When X falls, the keeper initially opposes the transition so it must be much weaker than the pulldown network. Eventually Y rises, turning the keeper OFF and avoiding static power dissipation.

The keeper must be strong (i.e., wide) enough to compensate for any leakage current drawn when the output is floating and the pulldown stack is OFF. Strong keepers also improve the noise margin because when the inputs are slightly above $V_t$ the keeper can supply enough current to hold the output high.

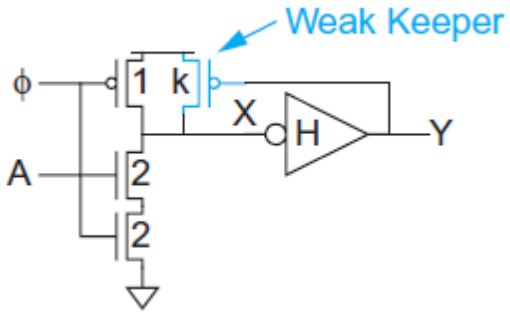



Fig 1.20 Conventional keeper

For small dynamic gates, the keeper must be weaker than a minimum-sized transistor. This is achieved by increasing the keeper length, as shown in Figure 1.21(a).

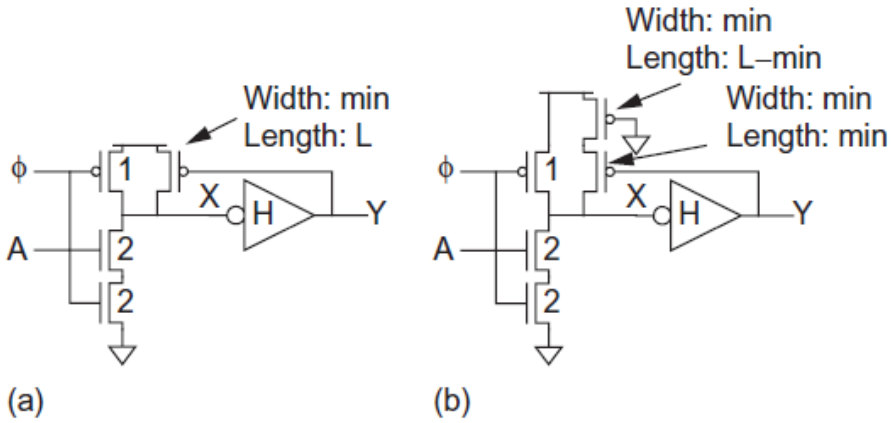



Fig 1.21 Weak keeper implementations

2.3.4 Multiple-Output Domino Logic (MODL) It is often necessary to compute multiple functions where one is a subfunction of another or shares a subfunction. Multiple-output domino logic (MODL) saves area by combining all of the computations into a multiple-output gate.

A popular application is in addition, where the carry-out $c_i$ of each bit of a 4-bit block must be computed, Each bit position i in the block can either propagate the carry ($p_i$) or generate a carry ($g_i$). The carry-out logic is

$$c_1 = g_1 + p_1 c_0$$
$$c_2 = g_2 + p_2(g_1 + p_1 c_0)$$
$$c_3 = g_3 + p_3(g_2 + p_2(g_1 + p_1 c_0))$$
$$c_4 = g_4 + p_4(g_3 + p_3(g_2 + p_2(g_1 + p_1 c_0)))$$

This can be implemented in four compound AOI gates, as shown in Figure 1.22.

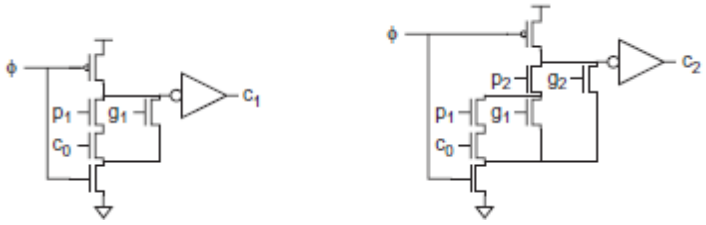



Fig 1.22 Conventional and MODL carry chains

2.3.5 NP and Zipper Domino: Another variation on domino is shown in Figure 1.23(a).The HI-skew inverting static gates are replaced with predischarged dynamic gates using pMOS logic. For example, a footed dynamic p-logic NAND gate is shown in Figure 1.23(b). When $\phi$ is 0, the first and third stages precharge high while the second stage predischarges low. When $\phi$ rises, all the stages evaluate. Domino connections are possible, as shown in Figure 1.23(c). The design style is called NP Domino or NORA Domino (NO RAce)

NORA has two major drawbacks. The logical effort of footed p-logic gates is generally worse than that of HI-skew gates. Secondly, NORA is extremely susceptible to noise.

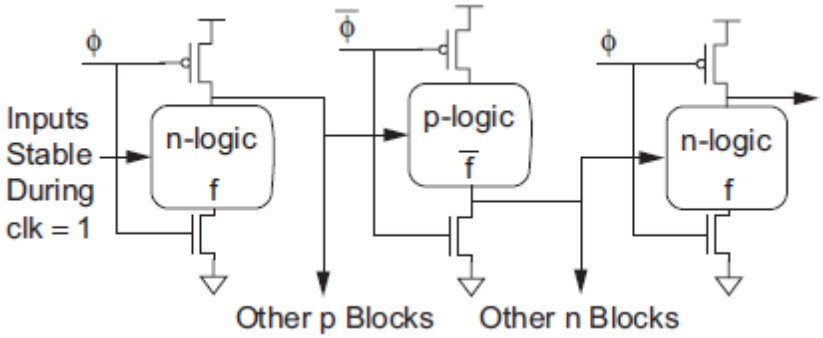



Fig 1.23 NP Domino

Disadvantagrs of NORA:
1.Logical effort is the worst
2.Susceptible to noise

2.3.5.1 Pass-Transistor Circuits:

Pass transistor is a single MOS that pass the signal between the drain and source terminal instead of a fixed power supply. These circuits build switches using either nMOS pass transistors or parallel pairs of nMOS and pMOS transistors called transmission gates. Full adders and other circuits rich in XORs also can be efficiently constructed with pass transistors. Pass-transistor circuits are essentially equivalent ways to draw the fundamental logic structures. An independent evaluation finds that for most general purpose logic, static CMOS is superior in speed, power, and area. In comparison, static CMOS NAND and NOR gates are relatively efficient and benefit less from pass transistors.

1.5.1 CMOS with Transmission Gates: Structures such as tristates, latches, and multiplexers are often drawn as transmission gates in conjunction with simple static CMOS logic.If multiple stages of logic are cascaded, they can be viewed as alternating transmission gates and inverters. Figure 1.24(a) redraws the multiplexer to include the inverters from the previous stage that drive the diffusion inputs but to exclude the output inverter. Figure 1.24(b) shows this multiplexer drawn at the transistor level.

The shorting of the intermediate nodes has two effects on delay. The effective resistance decreases somewhat (especially for rising outputs) because the output is pulled up or down through the parallel combination of both pass transistors rather than through a single transistor. However, the effective capacitance increases slightly because of the extra diffusion and wire capacitance required for this shorting.

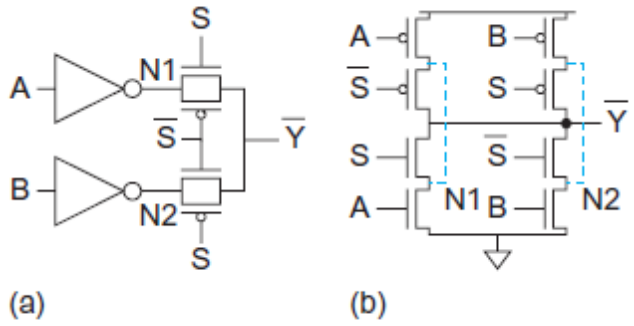



Fig 1.24 Alternate representations of CMOSTG in a 2-input inverting multiplexer

2.3.5.2. Complementary Pass Transistor Logic (CPL): CPL can be understood as an improvement on CVSL. CVSL is slow because one side of the gate pulls down, and then the cross-coupled pMOS transistor pulls the other side up. The size of the cross coupled device is an inherent compromise between a large transistor that fights the pulldown excessively and a small transistor that is slow pulling up. CPL resolves this problem by making one half of the gate pull up while the other half pulls down.

CPL is an inverting approach because it provides compact logic gates and the cell layout is reusable. The main drawbacks are the threshold loss and the fact that an input variable may have to drive more than one terminal.

Pass Transistor -Advantages:
1.Low Capacitance
Disadvantages:
1.nMOS is strong at passing 0
2.Has body effect

Advantages of dynamic Circuits:
1.Lower input capacitance
2.No contention during switching
3.Zero static power dissipation

Disadvantages of dynamic circuits:
1.Require careful clocking
2.Consume significant dynamic power
Sensitive to noise
Applications of dynamic circuits
1.Used in wide NOR functions
2.Used in multiplexer

**2.4. Dynamic CMOS Design:**

It was noted earlier that static CMOS logic with a fan-in of N requires 2N devices. A variety of approaches were presented to reduce the number of transistors required to implement a given logic function including pseudo-NMOS, pass transistor logic, etc. The pseudo-NMOS logic style requires only N + 1 transistors to implement an N input logic gate, but unfortunately it has static power dissipation. In this section, an alternate logic style called dynamic logic is presented that obtains a similar result, while avoiding static power consumption. With the addition of a clock input, it uses a sequence of precharge and conditional evaluation phases.

2.4.1 Dynamic Logic: Basic Principles

The basic construction of an (n-type) dynamic logic gate is shown in Figure 6.52a. The PDN (pull-down network) is constructed exactly as in complementary CMOS. The operation of this circuit is divided into two major phases: precharge and evaluation, with the mode of operation determined by the clock signal CLK.

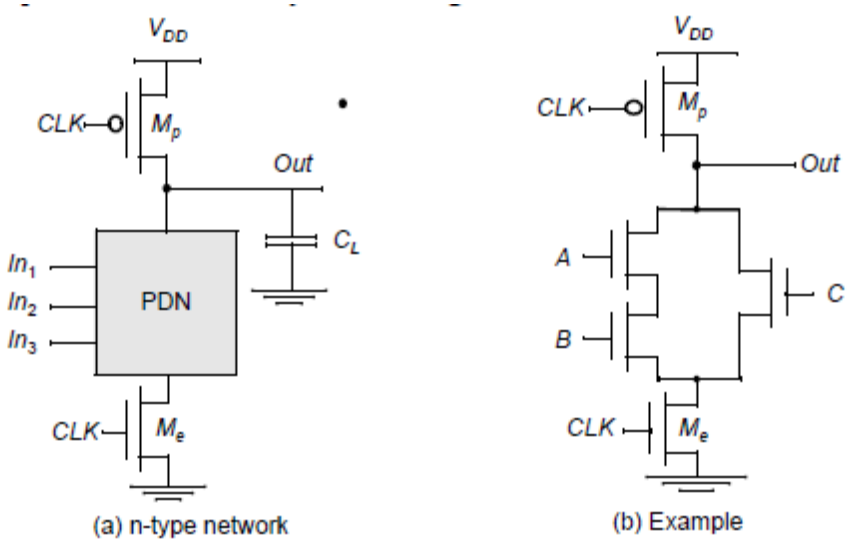



**Figure 6.52** Basic concepts of a dynamic gate.

**Precharge**

When CLK = 0, the output node Out is precharged to VDD by the PMOS transistor Mp.During that time, the evaluate NMOS transistor Me is off, so that the pull-down path is disabled. The evaluation FET eliminates any static power that would be consumed during theprecharge period (this is, static current would flow between the supplies if both the pulldown and the precharge device were turned on simultaneously).

**Evaluation**

For CLK = 1, the precharge transistor Mp is off, and the evaluation transistor Me is turne on. The output is conditionally discharged based on the input values and the pull-down topology. If the inputs are such that the PDN conducts, then a low resistance path exists between Out and GND and the output is discharged to GND. If the PDN is turned off, the precharged value remains stored on the output capacitance CL, which is a combination of junction capacitances, the wiring capacitance, and the input capacitance of the fan-out gates. During the evaluation phase, the only possible path between the output node and a supply rail is to GND. Consequently, once Out is discharged, it cannot be charged again till then next precharge operation. The inputs to the gate can therefore make at most one transition during evaluation. Notice that the output can be in the high-impedance state during the evaluation period if the pull-down network is turned off. This behavior is fundamentally different from the static counterpart that always has a low resistance path between the output and one of the power rails.

As as an example, consider the circuit shown in Figure 6.52b. During the precharge phase (CLK=0), the output is precharged to VDD regardless of the input values since the evaluation device is turned off. During evaluation (CLK=1), a conducting path is created between Out and GND if (and only if) A·B+C is TRUE. Otherwise, the output remains at the precharged state of VDD. The following function is thus realized,

A number of important properties can be derived for the dynamic logic gate:

• The logic function is implemented by the NMOS pull-down network. The construction of the PDN proceeds just as it does for static CMOS.

• The number of transistors (for complex gates) is substantially lower than in the static case: N + 2 versus 2N.

• It is non-ratioed. The sizing of the PMOS precharge device is not important for realizing proper functionality of the gate. The size of the precharge device can be made large to improve the low-to-high transition time (of course, at a cost to the high-tolow transition time). There is however, a trade-off with power dissipation since a larger precharge device directly increases clock-power dissipation.

• It only consumes dynamic power. Ideally, no static current path ever exists between VDD and GND. The overall power dissipation, however, can be significantly higher compared to a static logic gate.

• The logic gates have faster switching speeds. There are two main reasons for this. The first (obvious) reason is due to the reduced load capacitance attributed to the lower number of transistors per gate and the single-transistor load per

fan-in. Second, the dynamic gate does not have short circuit current, and all the current provided by the pull-down devices goes towards discharging the load capacitance.

The low and high output levels VOL and VOH are easily identified as GND and VDD and are not dependent upon the transistor sizes. The other VTC parameters are dramatically different from static gates. Noise margins and switching thresholds have been defined as static quantities that are not a function of time. To be functional, a dynamic gate requires a periodic sequence of precharges and evaluations. Pure static analysis, therefore, does not apply. During the evaluate period, the pull-down network of a dynamic inverter starts to conduct when the input signal exceeds the threshold voltage (VTn) of the NMOS pull-down transistor. Therefore, it is reasonable to set the switching threshold (VM) as well as VIH and VIL of the gate equal to VTn. This translates to a low value for the NML.

**2.4.2 Speed and Power Dissipation of Dynamic Logic**

The main advantages of dynamic logic are increased speed and reduced implementation area. Fewer devices to implement a given logic function implies that the overall load capacitance is much smaller. The analysis of the switching behavior of the gate has some interesting peculiarities to it. After the precharge phase, the output is high. For a low input signal, no additional switching occurs. As a result, tpLH = 0! The high-to-low transition, on the other hand, requires the discharging of the output capacitance through the pull-down network. Therefore tpHL is proportional to CL and the current-sinking capabilities of the pull-down network. The presence of the evaluation transistor slows the gate somewhat, as it presents an extra series resistance. Omitting this transistor, while functionally not forbidden, may result in static power dissipation and potentially a performance loss.

The above analysis is somewhat unfair, because it ignores the influence of the precharge time on the switching speed of the gate. The precharge time is determined by the time it takes to charge CL through the PMOS precharge transistor. During this time, the logic in the gate cannot be utilized. However, very often, the overall digital system can be designed in such a way that the precharge time coincides with other system functions. For instance, the precharge of the arithmetic unit in a microprocessor can coincide with the instruction decode. The designer has to be aware of this "dead zone" in the use of dynamic logic, and should carefully consider the pros and cons of its usage, taking the overall system requirements into account.

When evaluating the power dissipation of a dynamic gate, it would appear that dynamic logic presents a significant advantage. There are three reasons for this. First, the physical capacitance is lower since dynamic logic uses fewer transistors to implement a given function. Also, the load seen for each fanout is one transistor instead of two. Second, dynamic logic gates by construction can at most have one transition per clock cycle. Glitching (or dynamic hazards) does not occur in dynamic logic. Finally, dynamic gates do not exhibit short circuit power since the pull-up path is not turned on when the gate is evaluating.

While these arguments are generally true, they are offset by other considerations: (i) the clock power of dynamic logic can be significant, particularly since the clock node has a guaranteed transition on every single clock cycle; (ii) the number of transistors is higher than the minimal set required for implementing the logic; (iii) short-circuit power may exist when leakage-combatting devices are added (as will be discussed further); (iv) and, most importantly, dynamic logic generally displays a higher switching activity due to the periodic precharge and discharge operations. Earlier, the transition probability for a static gate was shown to be p0 p1= p0 (1-p0). For dynamic logic, the output transition probability does not depend on the state (history) of the inputs, but rather on the signal probabilities only. For an n-tree dynamic gate, the output makes a 0Õ1 transition during the precharge phase only if the output was discharged during the preceding evaluate phase. The 0Õ1 transition probability for an n-type dynamic gate hence equals

$$\alpha_{0 \rightarrow 1} = p_0$$

where p0 is the probability that the output is zero. This number is always larger or equal to p0 p1. For uniformly distributed inputs, the transition probability for an N-input gate is:

$$\alpha_{0 \to 1} = \frac{N_0}{2^N}$$

where N0 is the number of zero entries in the truth table of the logic function.

### 2.4.3 Issues in Dynamic Design

Dynamic logic clearly can result in high performance solutions compared to static circuits. However, there are several important considerations that must be taken into account if one wants dynamic circuits to function properly. This include charge leakage, charge sharing, backgate (and in general capacitive) coupling, and clock feedthrough. Some of these issues are highlighted in this section.

**Charge Leakage**

The operation of a dynamic gate relies on the dynamic storage of the output value on a capacitor. If the pull-down network is off, the output should ideally remain at the precharged state of VDD during the evaluation phase. However, this charge gradually leaks away due to leakage currents, eventually resulting in a malfunctioning of the gate. Figure 6.56a shows the sources of leakage for the basic dynamic inverter circuit.

Source 1 and 2 are the reverse-biased diode and sub-threshold leakage of the NMOS pull-down device M1, respectively. The charge stored on CL will slowly leak away due these leakage sources, assuming that the input is at zero during evaluation. Charge leakage causes a degradation in the high level (Figure 6.56b). Dynamic circuits therefore require a minimal clock rate, which is typically on the order of a few kHz. This makes the usage of dynamic techniques unattractive for low performance products such as watches, or processors that use conditional clocks (where there are no guarantees on minimum clock rates). Note that the PMOS precharge device also contributes some leakage current due to the reverse bias diode (source 3) and the subthreshold conduction (source 4). To some extent, the leakage current of the PMOS counteracts the leakage of the pull-down path. As a result the output voltage is going to be set by the resistive divider composed of the pull-down and pull-up paths.

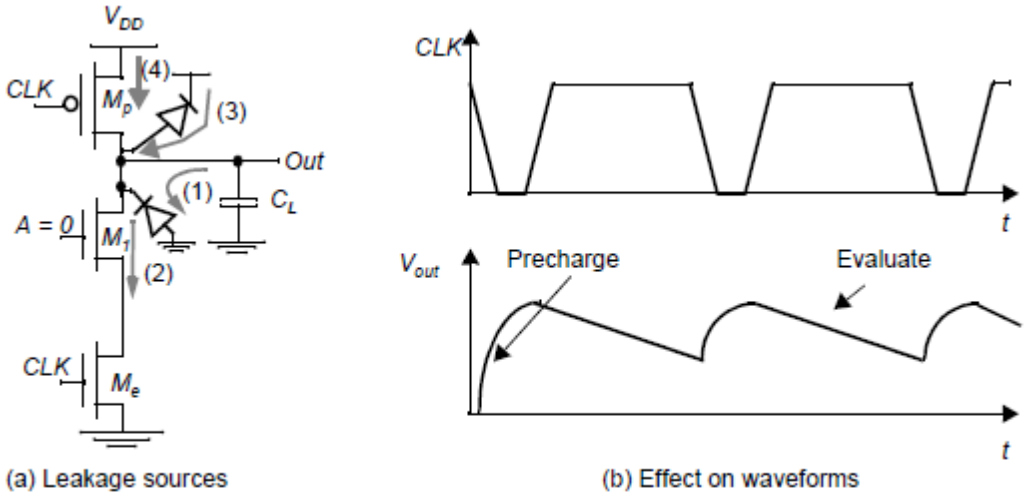



**Figure 6.56** Leakage issues in dynamic circuits.

Leakage is caused by the high impedance state of the output node during the evaluate mode, when the pull down path is turned off. The leakage problem van be counteracted by reducing the output impedance on the output node during evaluation. This is often done by adding a bleeder transistor as shown in Figure 6.58a. The only function of the bleeder—a pseudo-NMOS-like pull-up device—is to compensate for the charge lost due to the pull-down leakage paths. To avoid the ratio problems associated with this style of circuit and the associated static power

consumption, the bleeder resistance is made high, or, in other words, the device is kept small. This allows the (strong) pull-down devices to lower the Out node substantially below the switching threshold of the inverter. Often, the bleeder is implemented in a feedback configuration to eliminate the static power dissipation (Figure 6.58b).

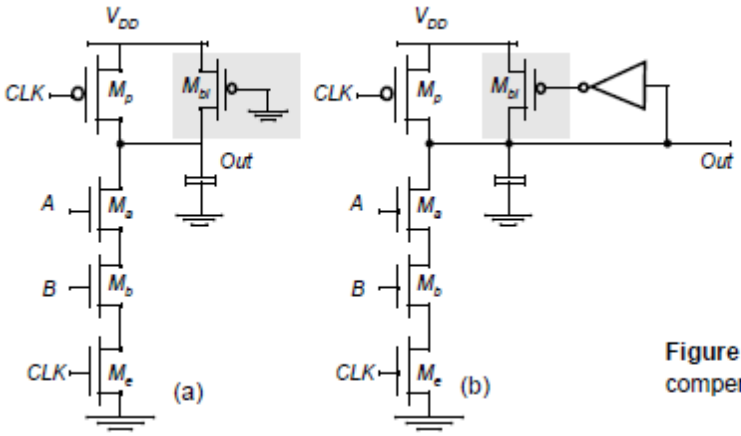



**Figure 6.58** Static bleeders compensates for the charge-leakage.

**Charge Sharing**

Another important concern in dynamic logic is the impact of charge sharing. Consider the circuit of Figure 6.59. During the precharge phase, the output node is precharged to VDD. Assume that all inputs are set to 0 during precharge, and that the capacitance Ca is discharged. Assume further that input B remains at 0 during evaluation, while input A makes a 0 --1 transition, turning transistor Ma on. The charge stored originally on capacitor CL is redistributed over CL and Ca. This causes a drop in the output voltage, which cannot be recovered due to the dynamic nature of the circuit.

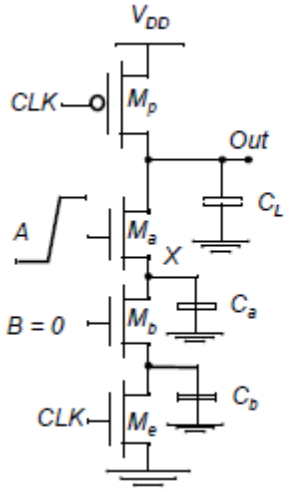



**Figure 6.59** Charge sharing in dynamic networks.

The influence on the output voltage is readily calculated. Under the above assumptions, the following initial conditions are valid: Vout(t = 0) = VDD and VX(t = 0) = 0. Two possible scenarios must be considered:

1. $\Delta V_{out} < V_{Tn}$ — In this case, the final value of $V_X$ equals $V_{DD} - V_{Tn}(V_X)$. Charge conservation yields

$$C_L V_{DD} = C_L V_{out}(t) + C_a[V_{DD} - V_{Tn}(V_X)]$$

or

$$\Delta V_{out} = V_{out}(t) - V_{DD} = -\frac{C_a}{C_L}[V_{DD} - V_{Tn}(V_X)] \quad (6.37)$$

2. $\Delta V_{out} > V_{Tn}$ — $V_{out}$ and $V_X$ reach the same value:

$$\Delta V_{out} = -V_{DD}\left(\frac{C_a}{C_a + C_L}\right) \quad (6.38)$$

Which of the above scenarios is valid is determined by the capacitance ratio. The boundary condition between the two cases can be determined by setting ∆Vout equal to VTn in Eq. (6.38), yielding

$$\frac{C_a}{C_L} = \frac{V_{Tn}}{V_{DD} - V_{Tn}}$$

Overall, it is desirable to keep the value of ∆Vout below |VTp|. The output of the dynamic gate might be connected to a static inverter, in which case the low level of Vout would cause static power consumption. One major concern is circuit malfunction if the output voltage is brought below the switching threshold of the gate it drives. The most common and effective approach to deal with the charge redistribution is to also precharge critical internal nodes, as is shown in Figure 6.61. Since the internal nodes are charged to VDD during precharge, charge sharing does not occur. This solution obviously comes at the cost of increased area and capacitance.

**Capacitive Coupling**

The high impedance of the output node makes the circuit very sensitive to crosstalk effects. A wire routed over a dynamic node may couple capacitively and destroy the state of the floating node. Another equally important form of capacitive coupling is the backgate (or output-to-input) coupling. Consider the circuit shown in Figure 6.62 in which a dynamic two-input NAND gate drives a static NAND gate. A transition in the input In of the static gate may cause the output of the gate (Out2) to go low. This output transition couples capacitively to the other input of the gate, the dynamic node Out1, through the gate-source and gate-drain capacitances of transistor M4. A simulation of this effect is shown in Figure 6.63, and demonstrates that the output of the dynamic gate can drop significantly.

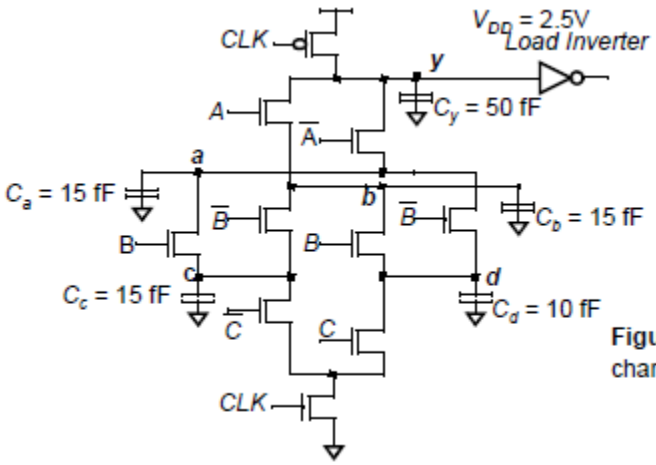



**Figure 6.60** Example illustrating the charge sharing effect in dynamic logic.

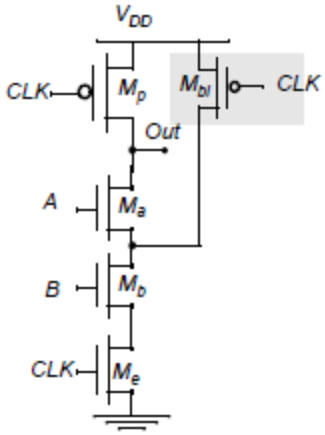



**Figure 6.61** Dealing with charge-sharing by precharging internal nodes. An NMOS precharge transistor may also be used, but this requires an inverted clock.

This further causes the output of the static NAND gate not to drop all the way down to 0V, and a small amount of static power is dissipated. If the voltage drop is large enough, the circuit can evaluate incorrectly, and the NAND output may not go low. When designing and laying out dynamic circuits, special care is needed to minimize capacitive coupling.

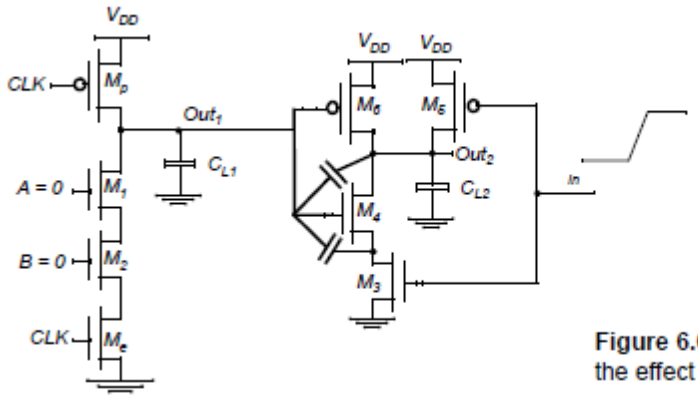



**Figure 6.62** Example demonstrating the effect of backgate coupling.

**Clock-Feedthrough**

A special case of capacitive coupling is clock-feedthrough, an effect caused by the capacitive coupling between the clock input of the precharge device and the dynamic output node. The coupling capacitance consists of the gate-to-drain capacitance of the precharge device, and includes both the overlap and the channel capacitances. This capacitive coupling causes the output of the dynamic node to rise above VDD on the low-to-high transition of the clock, assuming that the pull-down network is turned off. Subsequently, the fast rising and falling edges of the clock couple onto the signal node, as is quite apparent in the simulation of Figure 6.63.

The danger of clock feedthrough is that it may cause the (normally reverse-biased) junction diodes of the precharge transistor to become forward-biased. This causes electron injection into the substrate, which can be collected by a nearby high impedance node in the 1 state, eventually resulting in faulty operation. CMOS latchup might be another result of this injection. For all purposes, high-speed dynamic circuits should be carefully simulated to ensure that clock-feedthrough effects stay within bounds.All the above considerations demonstrate that the design of dynamic circuits is rather tricky and requires extreme care. It should therefore only be attempted when high performance is required.

**2.4.4 Cascading Dynamic Gates**

Besides the signal integrity issues, there is one major catch that complicates the design of dynamic circuits: straightforward cascading of dynamic gates to create more complex structures does not work. The problem is best illustrated with the two cascaded n-type dynamic inverters, shown in Figure 6.64a. During the precharge phase (i.e., CLK =0), the outputs of both inverters are precharged to VDD. Assume that the primary input In makes a 0--1 transition (Figure 6.64b). On the rising edge of the clock, output Out1 starts to discharge. The second output should

remain in the precharged state of VDD as its expected value is 1 (Out1 transitions to 0 during evaluation). However, there is a finite propagation delay for the input to discharge Out1 to GND. Therefore, the second output also starts to discharge. As long as Out1 exceeds the switching threshold of the second gate, which approximately equals VTn, a conducting path exists between Out2 and GND, and precious charge is lost at Out2. The conducting path is only disabled once Out1 reaches VTn, and turns off the NMOS pull-down transistor. This leaves Out2 at an intermediate voltage level. The correct level will not be recovered, as dynamic gates rely on capacitive storage in contrast to static gates, which have dc restoration. The charge loss leads to reduced noise margins and potential malfunctioning.

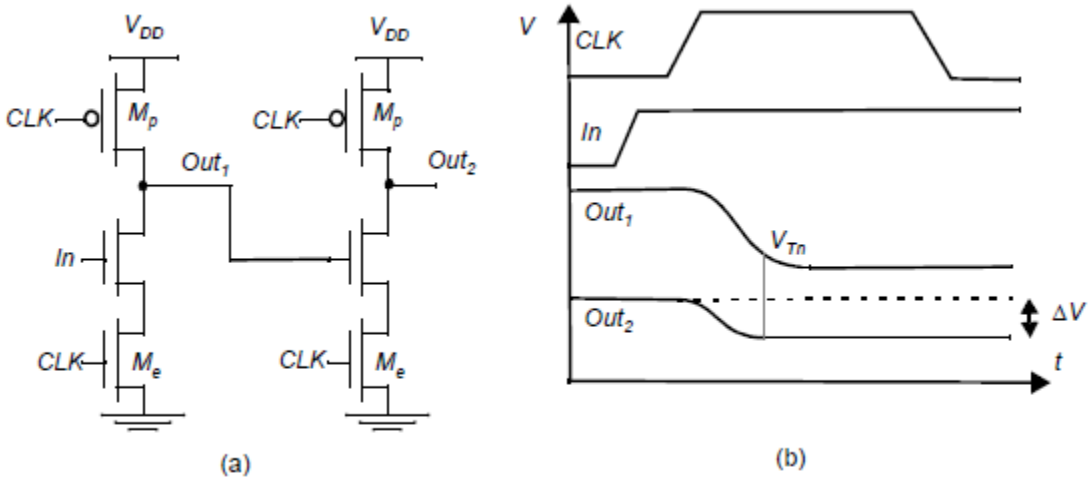



**Figure 6.64** Cascade of dynamic *n*–type blocks.

The cascading problem arises because the outputs of each gate—and hence the inputs to the next stages—are precharged to 1. This may cause inadvertent discharge in the beginning of the evaluation cycle. Setting all the inputs to 0 during precharge addresses that concern. When doing so, all transistors in the pull-down network are turned off after precharge, and no inadvertent discharging of the storage capacitors can occur during evaluation. In other words, correct operation is guaranteed as long as the inputs can only make a single 0 --1 transition during the evaluation period2. Transistors are only be turned on when needed, and at most once per cycle. A number of design styles complying with this rule have been conceived. The two most important ones are discussed below.

**Domino Logic**

Concept. A Domino logic module [Krambeck82] consists of an n-type dynamic logic block followed by a static inverter (Figure 6.65). During precharge, the output of the ntype dynamic gate is charged up to VDD, and the output of the inverter is set to 0. During evaluation, the dynamic gate conditionally discharges, and the output of the inverter makes a conditional transition from 0 --1. If one assumes that all the inputs of a Domino gate are outputs of other Domino gates3, then it is ensured that all inputs are set to 0 at the end of the precharge phase, and that the only transitions during evaluation are 0 --1 transitions.The formulated rule is hence obeyed. The introduction of the static inverter has the additional advantage that the fan-out of the gate is driven by a static inverter with a lowimpedance output, which increases noise immunity. The buffer furthermore reduces the capacitance of the dynamic output node by separating internal and load capacitances.

Consider now the operation of a chain of Domino gates. During precharge, all inputs are set to 0. During evaluation, the output of the first Domino block either stays at 0 or makes a 0--1 transition, affecting the second gate. This effect might ripple through the whole chain, one after the other, similar to a line of falling dominoes hence the name. Domino CMOS has the following properties:

- Since each dynamic gate has a static inverter, only non-inverting logic can be implemented. Although there are ways to deal with this, as is discussed in a subsequent section, this is major limiting factor, and pure Domino design has become rare.

- Very high speeds can be achieved: only a rising edge delay exists, while tpHL equals zero. The inverter can be sized to match the fan-out, which is already much smaller than in the complimentary static CMOS case, as only a single gate capacitance has to be accounted for per fan-out gate.

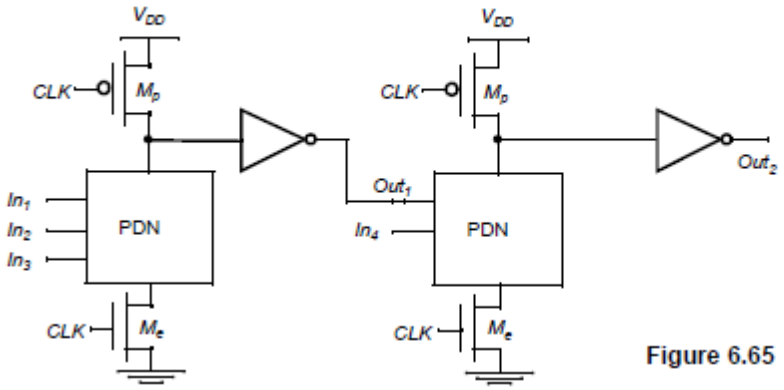



**Figure 6.65** DOMINO CMOS logic.

Since the inputs to a Domino gate are low during precharge, it is tempting to eliminate the evaluation transistor as this would reduce clock load and increase pull-down drive. However, eliminating the evaluation device extends the precharge cycle: the precharge now has to ripple through the logic network as well. Consider the logic network shown in Figure 6.66, where the evaluation devices have been eliminated. If the primary input In1 is 1 during evaluation, the output of each dynamic gate is 0 and the output of each static inverter is 1. On the falling edge of the clock, the precharge operation is started. Assume further that In1 makes a high-to-low transition. The input to the second gate is initially high, and it takes two gate delays before In2 is driven low. During that time, the second gate cannot precharge its output, as the pull-down network is fighting the precharge device. Similarly, the third gate has to wait till the second gate precharges before it can start precharging, etc. Therefore the time taken to precharge the logic circuit is equal to its critical path. Another important negative is the extra power dissipation when both pull-up and pull-down devices are on. It is therefore good practice to always utilize evaluation devices.

**Dealing with the Non-inverting Property of Domino Logic**

A major limitation in Domino logic is that only non-inverting logic can be implemented. This requirement has limited the widespread use of pure Domino logic. There are several ways to deal with the non-inverting logic requirement. Figure 6.67 shows one approach to the problem—reorganizing the logic using simple boolean transforms such as De Morgan's Law. Unfortunately, this sort of optimization is not always possible, and more general schemes may have to be used.

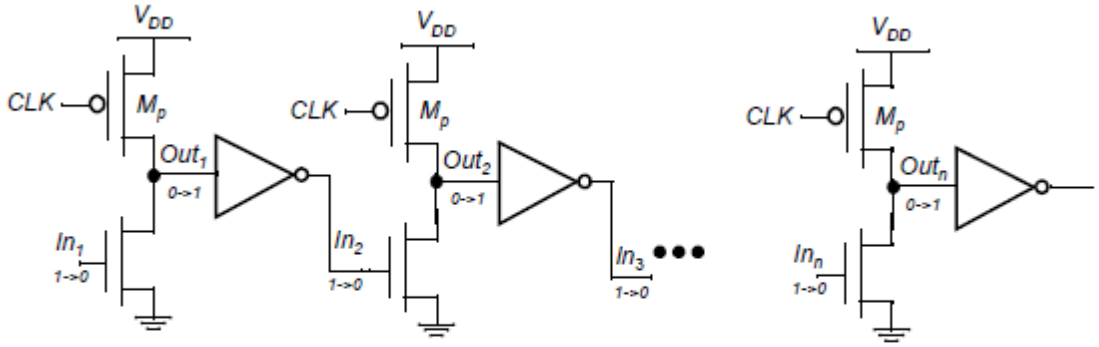



**Figure 6.66** Effect of ripple precharge when the evaluation transistor is removed. The circuit also exhibits static power dissipation.

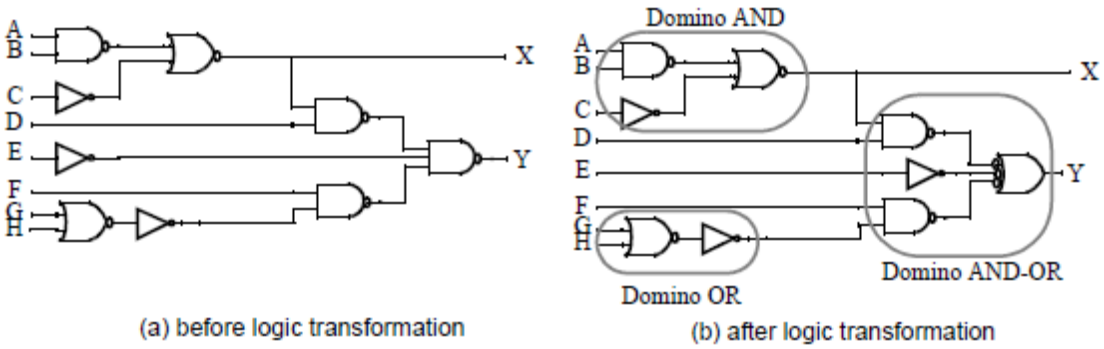



**Figure 6.67** Restructuring logic to enable implementation using non-inverting Domino Logic.

A general but expensive approach to solving the problem is the use of differential logic. Dual-rail Domino is similar in concept to the DCVSL structure discussed earlier, but uses a precharged load instead of a static cross-coupled PMOS load. Figure 6.68 shows the circuit schematic of an AND/NAND differential logic gate. Note that all inputs come from other differential Domino gates, and are low during the precharge phase, while making a conditional 0--1 transition during evaluation. Using differential Domino, it is possible to implement any arbitrary function. This comes at the expense of an increased power dissipation, since a transition is guaranteed every single clock cycle regardless of the input values—either O or $\bar{O}$ must make a 0--1 transition. The function of transistors Mf1 and Mf2 is to keep the circuit static when the clock is high for extended periods of time (bleeder). Notice that this circuit is not ratioed, even in the presence of the PMOS pull-up devices! Due to its high-performance, this differential approach is very popular, and is used in several commercial microprocessors.

**Optimization of Domino Logic Gates**.

Several optimizations can be performed on Domino logic gates. The most obvious performance optimization involves the sizing of the transistors in the static inverter. With the inclusion of the evaluation devices in Domino circuits, all gates precharge in parallel, and the precharge operation takes only two gate delays—charging the output of the dynamic gate to VDD, and driving the inverter output low. The critical path during evaluation goes through the pull-down path of the dynamic gate, and the PMOS pull-up transistor of the static inverter. Therefore, to speed up the circuit during evaluation, the beta ratio of the static inverter should be made high so that its switching threshold is close to VDD. This can be accomplished by using a small (minimum) sized NMOS and a large PMOS device. The minimum-sized NMOS only impacts the precharge time, which is limited in general due to the parallel precharging of all gates. The only disadvantage of using a large beta ratio is a reduction in noise margin. A designer should therefore simultaneously consider the reduced noise margin and performance during the device sizing.

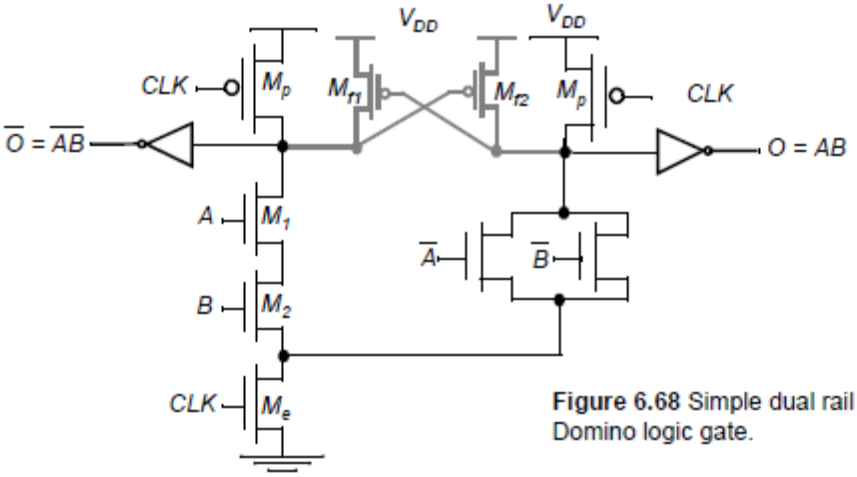



**Figure 6.68** Simple dual rail (differential) Domino logic gate.

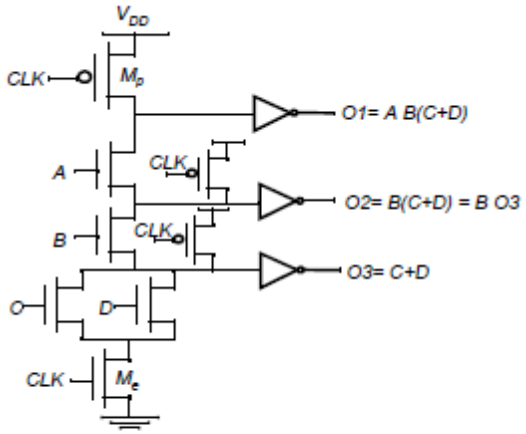



**Figure 6.69** Multiple output Domino

Numerous variations of Domino logic have been proposed [Bernstein98]. One optimization that reduces area is Multiple Output Domino Logic. The basic concept is illustrated is Figure 6.69. It exploits the fact that certain outputs are subsets of other outputs to generate a number of logical functions in a single gate. In this example, O3 = C+D is used in all three outputs, and hence it is implemented at the bottom of the pull-down network. Since O2 equals B ·O3, it can reuse the logic for O3. Notice that the internal nodes have to be precharged to VDD to produce the correct results. Given that the internal nodes precharge to VDD, the number of devices driving precharge devices is not reduced. However, the number of evaluation transistors is drastically reduced as they are amortized over multiple outputs. Additionally, this approach results in a reduction of the fan-out factor, once again due to the reuse of transistors over multiple functions.

Compound Domino (Figure 6.70) represents another optimization of the generic Domino gate, once again minimizing the number of transistors. Instead of each dynamic gate driving a static inverter, it is possible to combine the outputs of multiple dynamic gates with the aid of a complex static CMOS gate, as shown in Figure 6.70. The outputs of three dynamic structures, implementing O1 = A B C, O2 = D E F and O3 = G H, are combined using a single complex CMOS static gate that implements O = (o1+o2) o3. The total logic function realized this way equals O = A B C D E F + GH.

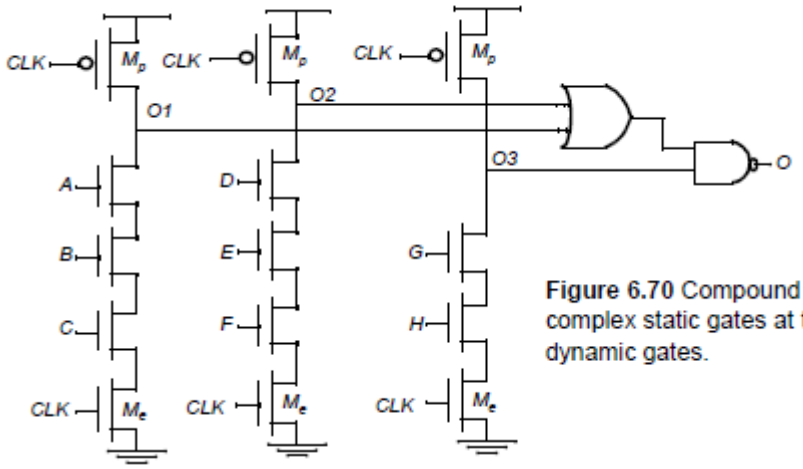



**Figure 6.70** Compound Domino logic uses complex static gates at the output of the dynamic gates.

Compound Domino is a useful tool for constructing complex dynamic logic gates.Large dynamic stacks are replaced using parallel small fan-in structures and complex CMOS gates. For example, a large fan-in Domino AND can be implemented as parallel dynamic NAND structures with lower fan-in that are combined using a static NOR

gate. One important consideration in Compound Domino is the problem associated with back gate coupling. Care must be taken to ensure that the dynamic nodes are not affected by the coupling between the output of the static gates and the output of dynamic nodes.

**np-CMOS**

The Domino logic presented in the previous section has the disadvantage that each dynamic gate requires an extra static inverter in the critical path to make the circuit functional.np-CMOS, provides an alternate approach to cascading dynamic logic by using two flavors (n-tree and p-tree) of dynamic logic. In a p-tree logic gate, PMOS devices are used to build a pull-up logic network, including a PMOS evaluation transistor (Figure 6.71) ([Goncalvez83, Friedman84, Lee86]). The NMOS predischarge transistor drives the output low during precharge. The output conditionally makes a 0 --1 transition during evaluation depending on its inputs.

np-CMOS logic exploits the duality between n-tree and p-tree logic gates to eliminate the cascading problem. If the n-tree gates are controlled by CLK, and p-tree gates are controlled using CLK, n-tree gates can directly drive p-tree gates, and vice-versa. Similar to Domino, n-tree outputs must go through an inverter when connecting to another n-tree gate. During the precharge phase (CLK = 0), the output of the n-tree gate, Out1, is charged to VDD, while the output of the p-tree gate, Out2, is pre-discharged to 0V. Since the n-tree gate connects PMOS pull-up devices, the PUN of the p-tree is turned off at that time. During evaluation, the output of the n-tree gate can only make a 1□ 0 transition, conditionally turning on some transistors in the p-tree. This ensures that no accidental discharge of Out2 can occur. Similarly, n-tree blocks can follow p-tree gates without any problems, as the inputs to the n-gate are precharged to 0. A disadvantage of the np-CMOS logic style is that the p-tree blocks are slower than the n-tree modules, due to the lower current drive of the PMOS transistors in the logic network. Equalizing the propagation delays requires extra area.

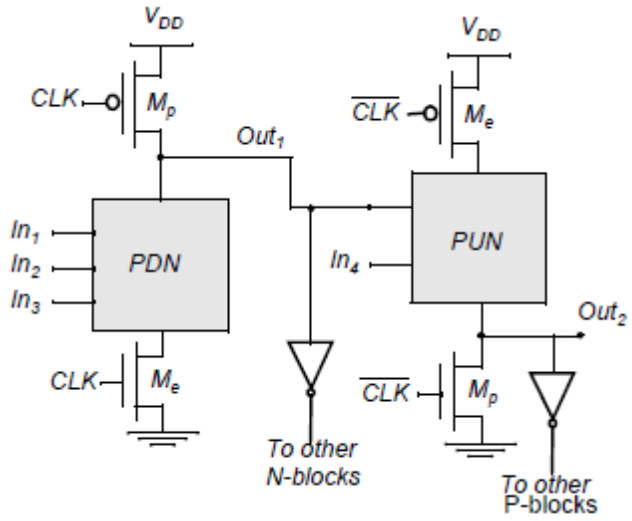



**Figure 6.71** The *np*-CMOS logic circuit style.

**2.5.Power Dissipation:**

The instantaneous power P(t) drawn from the power supply is proportional to the supply Current $I_{DD}(t)$ and the supply voltage $V_{DD}$.

$$P(t) = i_{DD}(t)V_{DD}$$

The energy consumed over the time interval T is the integral of P(t)

$$E=\int_0^T i_{DD}(t)V_{DD}\,dt$$

The average power over this interval is,

$$P_{avg}=E/T=1/T\int_0^T i_{DD}(t)V_{DD}\,dt$$

Power dissipation in CMOS circuits comes from two components

1.Static dissipation due to

- subthreshold conduction through OFF transistors
- tunneling current through gate oxide
- leakage through reverse-biased diodes
- contention current in ratioed circuits
- Dynamic dissipation due to

2.Dynamic Power Dissipation due to

- charging and discharging of load capacitances
- *short circuit* current while both PMOS and NMOS networks are partially ON

$$P_{total} = P_{static} + P_{dynamic}$$

Static Dissipation:

Considering the static CMOS inverter ,if the input ='0' the associated Nmos transistor is'OFF' and the pMOS transistor is 'ON'. This output voltage is VDD or logic '1'.When the input ='1',the associated nMOS transistor is 'ON' and the pMOS transistor is 'OFF'. The output voltage is '0' volts(GND).Note that one of the transistors is always OFF when the gate is in either of these logic states. Ideally,no current flows through the OFF transistor so the power dissipation is zero when the circuit is quiescent, i.e.,when no transistors are switching. Zero quisent power dissipation is a principle advantage of CMOS over competing transistor technologies. The static power dissipation is the product of total leakage current and the supply voltage.

$$P_{static}=I_{static}V_{DD}.$$

Dynamic Power Dissipation:

The Primary dynamic dissipation is charging the load capacitance. Suppose a load C is switched between $V_{DD}$ and GND at an average frequency of $f_{sw}$. Over any given time T, load will be charged and discharged $Tf_{sw}$ times. Current flows from $V_{DD}$ to the load to charge it. Current then flows from the load to GND during discharge. In one complete charge/discharge cycle, a total charge of $Q=CV_{DD}$ is transferred between $V_{DD}$ and GND. The average dynamic power dissipation is

$$P_{dynamic}=1/T\int_0^T i_{DD}(t)V_{DD}\,dt$$

$$=V_{DD}/T\int_0^T i_{DD}(t)dt$$

Taking the integral of the current over some interval T as the total charge delivered during that time ,we simplify to

$$P_{dynamic}=V_{DD}/T[T\,f_{sw}\,CV_{DD}]\;CV_{DD}^2f_{sw}$$

Because most gates do not switch every clock cycle, it is often more convenient to express switching frequency fsw as an activity factor α times the clock frequency. Now the dynamic power dissipation may be rewritten as:

$$P_{dynamic}=\alpha CV_{DD}^2f$$

A clock frequency has an activity factor of α=1 because it rises and falls every cycle. Most data has a maximum activity factor of 0.5 because it transitions only once each cycle. Static CMOS logic has been empirically determined to have activity factor closer to 0.1 because some gates maintain one output state more often than another and because real data inputs to some portions of a system often remain constant from one cycle to the next.

Low Power Design:

Total power dissipation is the sum of the static and dynamic dissipation components. Dynamic dissipation has historically been far greater than static power when systems are active, and hence, static power is often ignored, although this will change as gate and subthreshold leakage increase.

Power dissipation has become extremely important to VLSI designers. For high performance systems such as workstations and servers, dynamic power consumption per chip is often limited to about 150W by the amount of heat that can be managed with air-cooled systems and cost effective heat sinks.

For battery-based systems such as laptops, cell phones, and PDAs, power consumption sets the battery life of the product.

Dynamic Power Reduction:

If a process is selected with sufficiently high threshold voltages and oxide thickness, static dissipation is small and dynamic dissipation usually dominates while the chip is active. Dynamic power is reduced by reducing the activity factor, the switching capacitance, the power supply, or the operating frequency.

Activity factor is very important. Static logic has an inherently low activity factor of 1 and a very power hungry. The energy-delay product (EDP) is a popular metric that balances the importance of energy and delay. the power-delay product (PDP) is simply the energy. The minimum energy point is the least energy that an operation could consume if delay were unimportant. The minimum energy point typically consumes an order of magnitude less energy than the conventional operating point, but runs at least three orders of magnitude more slowly.

Static power Reduction: Static power reduction involves minimizing $I_{static}$. Some circuit techniques such as analog current sources and pseudo-nMOS gates intentionally draw static power. They are turned off when they are not needed.

$$I_{ds} = I_{ds0} e^{\frac{V_{gs} - V_t}{nV_T}} [1 - e^{\frac{-V_{ds}}{V_T}}]$$

Good design practice starts with high-$V_t$ devices everywhere and selectively introduces low-*Vt* devices where necessary. Using multiple thresholds requires additional implant masks that add to the cost of a CMOS process. Alternatively, designers can increase the channel length, which tends to raise the threshold voltage via the short channel effect. Most nanometer processes offer a thin oxide for logic transistors and a much thicker oxide for I/O transistors that can withstand higher voltages. The oxide thickness is controlled by another mask step. Gate leakage is negligible in the thick oxide devices, but their performance is inadequate for high speed logic applications. Some processes offer another intermediate oxide thickness to reduce gate leakage.

To achieve high $I_{on}$ in active mode and low $I_{off}$ in sleep mode is to dynamically adjust the threshold voltage of the transistor by applying a body bias. This technique is sometimes called *variable threshold CMOS* (VTCMOS).For example, low-*Vt* devices can be used and a *reverse body bias* (RBB) can be applied during sleep mode to reduce leakage. Alternatively, higher-*Vt* devices can be used, and then a *forward body bias* (FBB) can be applied during active mode to increase performance. Body bias can be applied to the power gating transistors to turn them off more effectively during sleep. Too much reverse body bias leads to greater junction leakage. , while too much forward body bias leads to substantial current through the body to source diodes.

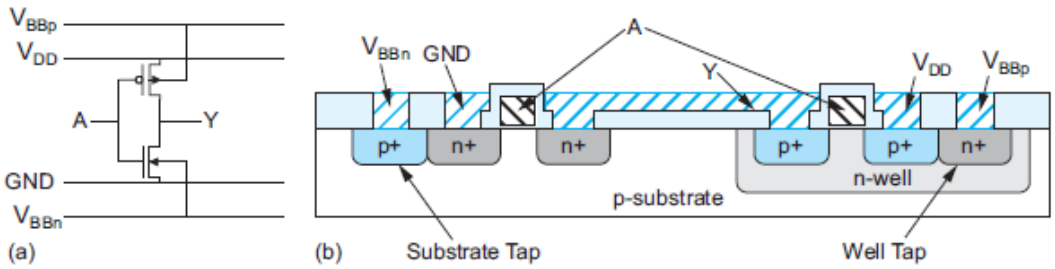



Fig 3.1 Body bias

Figure 3.1 shows a schematic and cross-section of an inverter using body bias. In an n-well process, all nMOS transistors share the same p substrate and must use the same $V_{BBn}$. In a triple-well process, groups of transistors can use different p-wells isolated from the substrate and thus can use different body biases. The well and substrate carry little current, so the bias voltages are relatively easy to generate using a charge pump.

**LOW POWER LOGIC DESIGN:**

Various techniques can be followed to achiveve low power dissipation. These are given below.

1.Suitable process to be selected. Dynamic power is low in the advanced manufacturing proces. Static power increases in the advanced process because subthreshold leakage is increased when the threshold decreased. Trade offs between the power and cost is very important.

2.Supply voltage is also very important parameter. Depend upon the process, the supply voltage is to be scaled. Depend upon the workload, the voltage and clock frequency are to be adjusted when there is no need for maximum performance.

3.The method of adjusting the power supply is to divide the logic into high speed and low power groups. Fast logic is connected to the high supply and slow logic is connected to the low supply.

4.The capacitance is also very important parameter which affect low power operation. It consists of gate capacitance, diffusion capacitance and wire capacitance. Gate and diffusion capacitance can be reduced by using less number of small transistors.Maximum sized, complementary CMOS with a low P/N ratio is the best for low power operation.

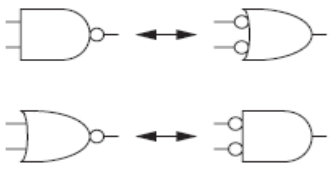
**TWO MARKS**:

1.What is Bubble Pushing.

CMOS stages are inherently inverting, So AND and OR functions must be built from NAND and NOR gates. DeMorgan's law helps with this conversion. A NAND gate is equivalent to an OR of inverted inputs. A NOR gate is equivalent to an AND of inverted inputs. The same relationship applies to gates with more inputs. Switching between these representations is easy to do on a whiteboard and is often called bubble pushing.

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$



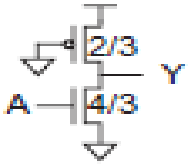
2.Define Skewed Gates.

One input transition is more important than the other skewed gates are used. We defined HI-skew gates to favor the rising output transition and LO-skew gates to favor the falling output transition. Inverters with different beta ratios r =βp / βn are called skewed inverters. If r >1, the inverter is HI-skewed. If r <1, the inverter is LO-skewed. If r =1, the inverter has normal skew or is unskewed.

Combinational Logic Circuits

3.Draw the basic structure of pseudo-Nmos structure.

In Pseudo-nMOS the pulldown network is like that of an ordinary static gate, but the pullup network has been replaced with a single pMOS transistor that is grounded so it is always ON.



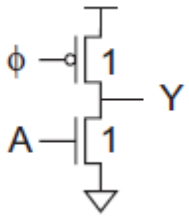

4.Drawbaks of Pseudo-nMOS

The drawbacks of ratioed circuits include

- slow rising transitions
- contention on the falling transitions
- static power dissipation
- a nonzero VOL.

5..What is meant by dynamic circuits.

Dynamic circuit operation is divided into two modes, During precharge, the clock Φ is 0, so the clocked pMOS is ON and initializes the output Y high. During evaluation, the clock is 1 and the clocked pMOS turns OFF. The output may remain high or may be discharged low through the pulldown network.




6.Advantages and Disadvantage of dynamic circuits.

Advantages:

1.Dynamic circuits are the fastest
2.They have lower input capacitance
3.No contention during switching
4. They also have zero static power dissipation

Disadvantages:

1.They require careful clocking
2.sensitive to noise during evaluation.
3.Monotonicity problem

22.How can you avoid monotonicity problem.

Monotonicity problem can be avoided by Domino Logic circuit. This problem can be solved by placing a static CMOS inverter between dynamic gates.

7..Define Pass-Transistor.

Pass transistor is a single MOS that pass the signal between the drain and source terminal instead of a fixed power supply. These circuits build switches using either nMOS pass transistors or parallel pairs of nMOS and pMOS transistors called transmission gates.

8..What is meant by Latch and flip-flop

Latches and flip-flops are the two most commonly used sequencing elements. Both have three terminals: data input (D), clock (clk), and data output (Q). The latch is transparent when the clock is high and opaque when the clock is low;in other words, when the clock is high, D flows through to Q as if the latch were just a buffer, but when the clock is low, the latch holds its present Q output even if D changes. The flip-flop is an edge-triggered device that copies D to Q on the rising edge of the clock and ignores D at all other times.

11.What is synchronizer

A synchronizer is a circuit that accepts an input that can change at arbitrary times and produces an output aligned to the synchronizer's clock. Because the input can change during the synchronizer's aperture, the synchronizer has a nonzero probability of producing a metastable output.

12.What is Arbiter

The arbiter is closely related to the synchronizer. It determines which of two inputs arrived first. If the spacing between the inputs exceeds some aperture time, the first input should be acknowledged.