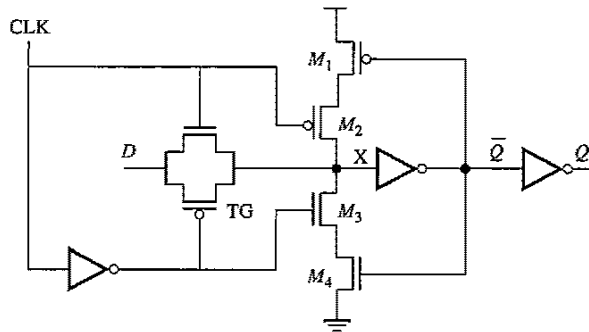


**Figure 7.23**

Typical D-latch implementation in CMOS.

of the INV2, TG2, TG1, and the gate that is driving D . The basic idea is to ensure that the forward path is “stronger” than the feedback path. Therefore, the driver and TG1 must be large enough such that the devices driving the forward path can swing the voltage at Q past the switching threshold V_s when competing with INV2 and TG2. If they are not sized properly, the old value will remain latched until TG2 turns off.

A more popular version of the D-latch is shown in Figure 7.23. This circuit employs the same principles as the one shown in Figure 7.22 but avoids any conflicts at internal nodes. The operation of this latch is as follows. When the CLK is high, the latch operates in transparent mode. Any changes at the input D are reflected at the output after a delay of $TG + 2\text{ INV}$. If the input data is high, then the value of $\bar{Q}$ is low. Therefore, M_1 is on and M_4 is off. When CLK goes low, then M_2 is turned on while M_3 is turned off. The internal node X is pulled high through M_1 and M_2 . Note that there is no conflict situation in this circuit, and node X is not a dynamic node. A similar analysis can be done with a low input data. This time M_3 and M_4 would act to pull the internal node low after the clock goes low. Again, there are no conflicts in this configuration.

The circuit for a master-slave D flip-flop can be constructed from two D-latches, as shown in Figure 7.24. The flip-flop is simply a cascade of two D-latches. The connection of the clock signals to the master and slave are important to note. The positive edge of the clock shuts off the first latch and enables the second latch. The operation of the flip-flop is as follows: when the clock is low, the first latch is transparent, while the second latch is disabled. The data will flow up to TG2, but no further since TG1 is on and TG2 is off. When CLK goes high, TG1 shuts off and holds D at the internal node Q . Either the pull-up or pull-down path for the internal Q is enabled depending on the data value. Meanwhile, TG2 turns on and allows Q to flow to the outputs. Since D is only transmitted to the output on the rising edge of CLK, this is a positive edge-triggered flip-flop.

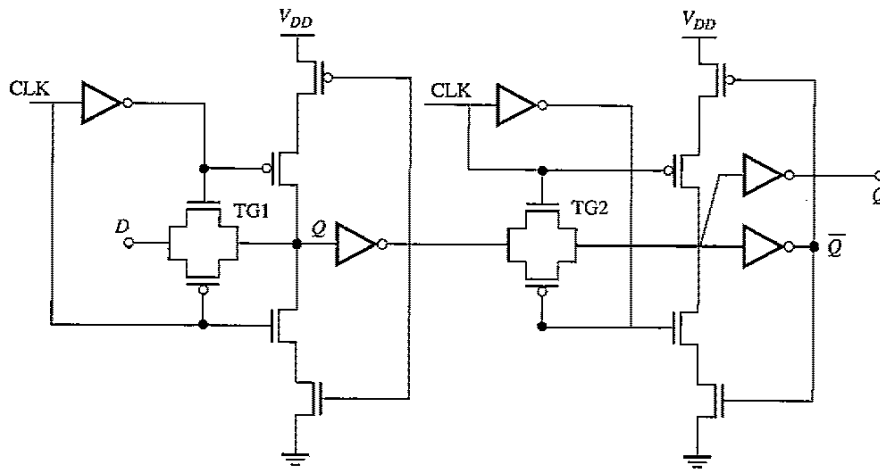
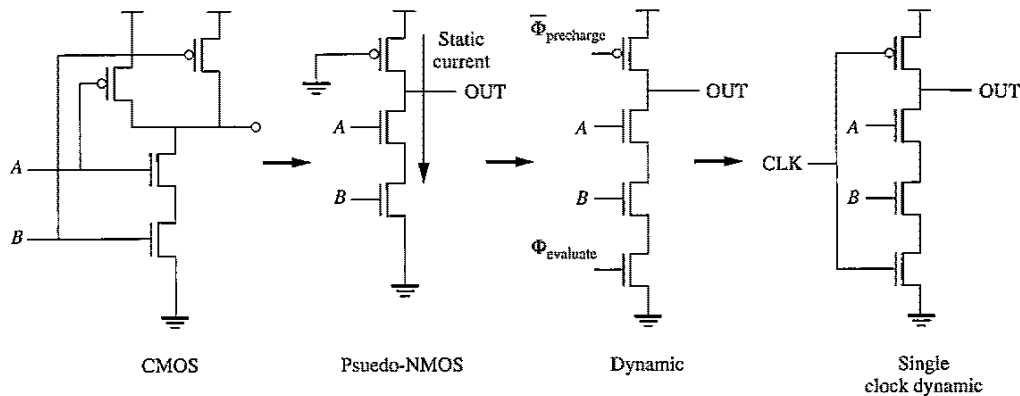


Figure 7.24
Positive edge-triggered D-type flip-flop.

7.5 Domino Logic

We have already seen the numerous difficulties that may arise with dynamic logic: charge sharing, feedthrough, charge leakage, single-event upsets, etc. These problems have limited the use of dynamic logic in industry. Furthermore, the CAD tool support needed to verify these types of designs are not readily available. Nevertheless, it is worthwhile to understand this approach for two reasons. First, static gates require numerous PMOS devices with relatively large sizes. Trying to implement a high fanin gate (i.e., a gate with more than four inputs) is difficult because either the N -complex or P -complex will require a large series stack. Pseudo-NMOS gates help to reduce the high fanin problem at the expense of static power when the output is low. The dynamic approach allows us to design circuits that are faster than static gates, with lower power than pseudo-NMOS, so they are worth exploring. We limit the treatment here to the most prominent dynamic configurations used in industry.

The structure of a dynamic logic gate can be derived from a static logic gate, as shown in the diagram of Figure 7.25. The conversion from static CMOS to pseudo-NMOS is straightforward, as shown in the first and second diagrams. We know that the second gate suffers from excessive power dissipation when the output is low. We need a way of shutting off the PMOS device when the output is low so as to reduce the power. What if we turned on the PMOS device to precharge the output to a high value, and then shut off the PMOS device during the period when the output would go low? For this purpose, we could introduce a new clock signal, $\Phi_{\text{precharge}}$, to turn the PMOS device *on* and *off* as needed. However, if we tried to charge up the output using the clocked PMOS device while a path to Gnd existed through the NMOS devices, we would have contention between the pull-up and pull-down paths.

**Figure 7.25**

Evolution from static gate to dynamic gate.

Therefore, we must add another NMOS transistor, called a *foot* transistor or *evaluation* transistor, in the pull-down path. This transistor is turned on only after the PMOS load is shut off using another clock signal, Φ_{evaluate} .

The configuration described above is depicted in the third circuit of Figure 7.25 where $\Phi_{\text{precharge}}$ is used to turn on the p -channel device and pull the output high (precharge phase), while Φ_{evaluate} is used to turn on the n -channel device to potentially pull the output node low (evaluation phase). The precharge signal is active low since it is connected to the PMOS device. The evaluate signal is active high since it turns on the NMOS device. Since the precharge and evaluate clocks are in phase, we can simply use one signal called CLK to drive the precharge and foot transistors. This is shown in the fourth circuit for the dynamic 2-input NAND gate.

The operation of the 2-input dynamic NAND gate is as follows. Initially, the precharged signal goes low and the output is precharged high. It is as if we are postulating a "1" at the output of all nodes during this phase. The precharge phase is intended to be a small portion of the clock cycle. Then, the CLK goes high and we enter the evaluation phase when the dynamic gates either remain high or are pulled low. For this NAND gate, if both A and B are high, then a path exists to Gnd. Otherwise, the output remains high and we preserve the value on the capacitance at the output node. This charge storage at the output node makes it a dynamic gate.

Dynamic Gate Implementations

Example 7.6

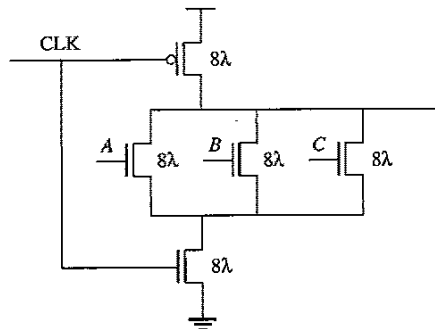
Problem:

Implement a 3-input NOR gate in dynamic logic and explain its operation. Size the transistors to deliver the same delays as a conventional CMOS inverter (PMOS $8\lambda:2\lambda$, NMOS $4\lambda:2\lambda$).

Solution:

The dynamic 3-input NOR gate requires three parallel NMOS transistors, plus two transistors connected to the CLK input: one PMOS pull-up and one NMOS foot transistor. When CLK is low, the precharge transistor pulls the output to V_{DD} . When the clock goes high, the evaluate transistor turns on. If A or B or C is high, the output will be discharged to Gnd. If all three inputs are low, the output will remain high.

The pull-up requires a width of 8λ to deliver the same rise delay as the CMOS inverter. This is somewhat irrelevant since all gates are precharged simultaneously. The pull-down tree should also be sized with 8λ transistors to produce the equivalent 4λ device to deliver the same fall delay as the CMOS inverter. The final sizes are shown below:



In reality, the CLK signal arrives first so A, B, and C can be considered as late arriving signals. In that case, the transistors can be reduced to half the size to deliver the same delay as the inverter.

Most logic functions can be implemented as dynamic circuits using the structure shown in Figure 7.26. The desired function is implemented in the NMOS complex. Transistors M_p and M_N are the precharge and evaluate transistors, respectively. The foot transistor is the only extra device relative to pseudo-NMOS gates. Note that all dynamic gates require a clock signal for proper operation. Furthermore, the clock must be routed to all dynamic gates, as opposed to static gates which do not require a clock. This places an additional burden on the physical layout tools.

There is a notable complication when one dynamic gate feeds another, as shown in Figure 7.27a. Since all output nodes are precharged high, all inputs to subsequent gates are high, implying that there will be an active path to Gnd from each output node as soon as the foot transistor is turned on. Once an output node has been discharged, it cannot go high until the next precharge phase. Therefore, a direct connection is not suitable.

A simple solution exists for this problem: add a static inverter to the output of all dynamic gates. Then define a "stage" as the dynamic gate plus the inverter. The

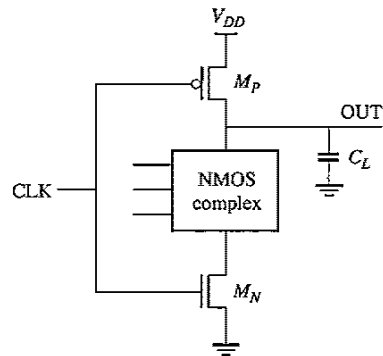
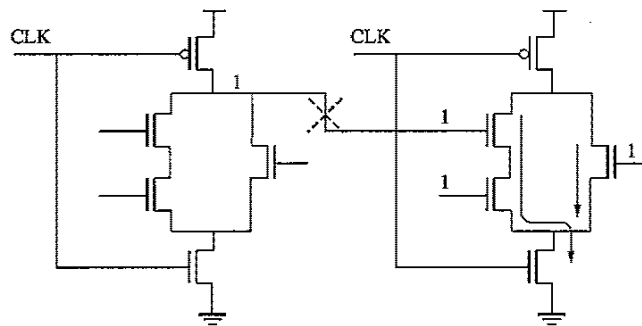
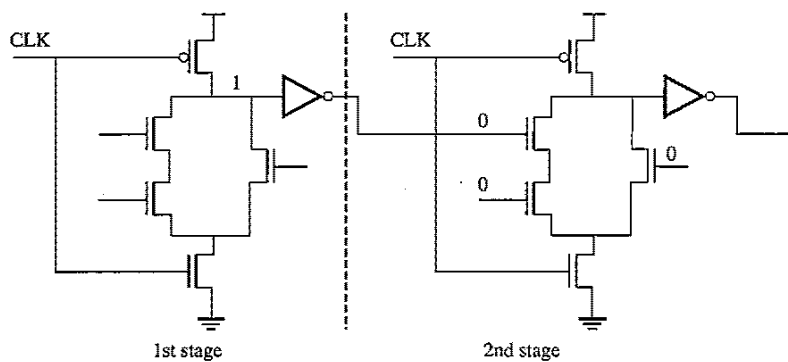


Figure 7.26
General structure of a dynamic gate.



(a) Direct connection not possible



(b) Insert inverter between dynamic gates

Figure 7.27
Connecting dynamic gates.

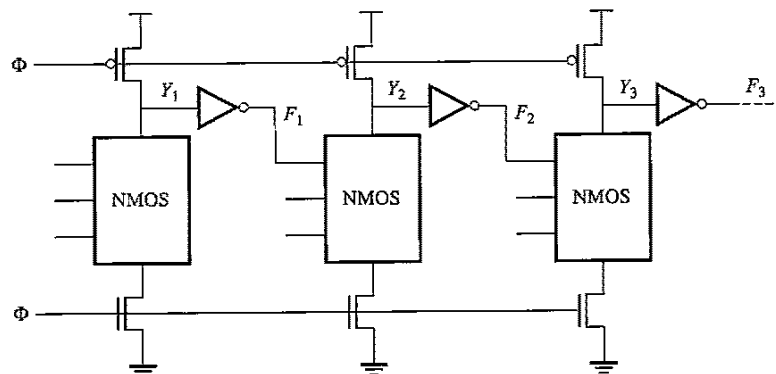


Figure 7.28
Domino cascaded gates.

output of each stage is now low during precharge. Therefore, all NMOS transistors are off during precharge and can only be turned on during the evaluate phase as shown in Figure 7.27b. The combination of a dynamic stage and an inverter is referred to as a *domino stage*, and circuits using this form of dynamic logic are referred to as *domino logic* circuits.

A cascade of three domino logic stages is shown in Figure 7.28. All three stages are precharged high when Φ is low. The clock, Φ , need not be low for very long since all stages are precharged simultaneously. In fact, the clock should only be low until all inverter outputs are low. When Φ goes high, we enter the evaluation phase where the internal nodes (labeled Y_1, Y_2, Y_3) will fall like dominos in order from left to right, assuming there is a path to the Gnd through their respective n -complexes. This is why the name *domino* is used to describe logic circuits implemented in this fashion.

Meanwhile, the true outputs, F_1, F_2 , and F_3 , would rise to V_{DD} , depending on the logic in the n -complexes. The clock must remain high long enough for logic to propagate through the entire chain. Therefore, an asymmetric clock with a relatively high duty cycle (percentage of time that the clock is high) is used for domino logic. The propagation delay is the sum of each dynamic block and inverter. However, we care most about the falling edge of the dynamic block and the rising edge of the inverter output. When we optimize the speed of this type of logic, we try to speed up these transitions because they are the only ones that control the cycle time.

In fact, we should design a domino stage with a stronger pull-down in the dynamic gate and a stronger pull-up in the static inverter. For the dynamic gate, we should increase the sizes of the NMOS devices in the n -complex. Since we are only interested in the rising output of the inverter, we could increase the size of its PMOS device. This type of inverter is a *skewed* inverter since its switching threshold is skewed relative to the conventional inverter. As a side-effect of the sizing operations, the V_S is higher for the static inverter. Furthermore, there is no pull-up fighting the pull-down in the first gate so its V_S is actually lower than expected (roughly equal

to the V_{TN} of the NMOS device). Therefore, a domino stage will actually switch earlier than a regular gate.

There is also a power savings in domino logic. Only those gates that discharge to Gnd will dissipate power. There will be no power dissipation due to crowbar current since a direct path between V_{DD} and Gnd is not allowed in this type of logic. In addition, glitches can be effectively removed since all inputs switch from low to high.

One disadvantage of domino logic is that it can only be used to create noninverting functions. This may not seem to be a problem at first glance, but one quickly realizes that *an inverter cannot be implemented* in domino logic. This is not a serious problem since we can push the inversions around the circuit as required until they appear at a primary input or output. We can also generate the true and complement values if necessary using two precharged gates. In the worst case, we will require twice as many gates but the overall speed will still be faster.

Adder Function in Domino Logic

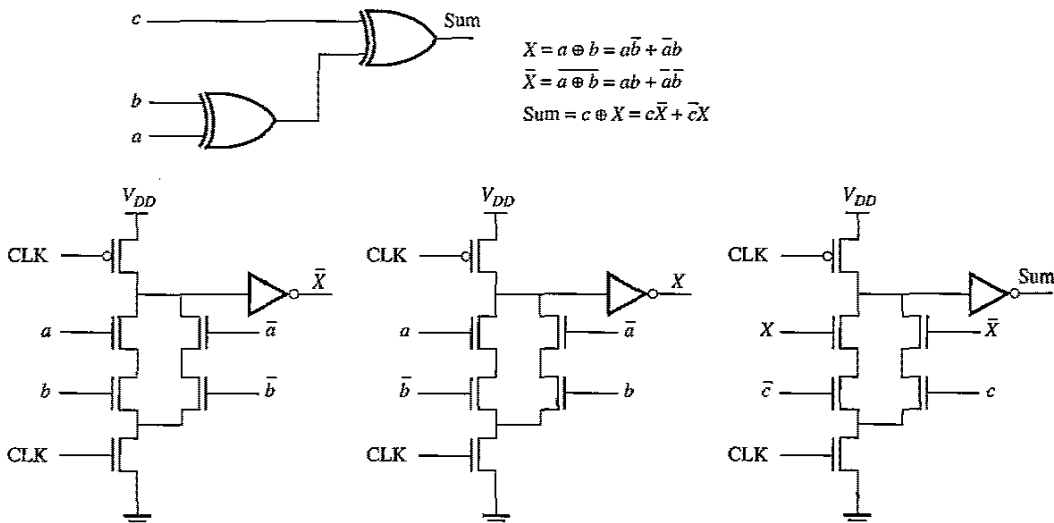
Example 7.7

Problem:

Implement the function $sum = a \oplus b \oplus c$ in domino logic. Assume that the literals $a, \bar{a}, b, \bar{b}$ are available as stable inputs to the gates.

Solution:

We need to create the XOR and XNOR of a and b using domino gates and then apply them to a third XOR gate. Note that we need the true and complement of X to implement the function.



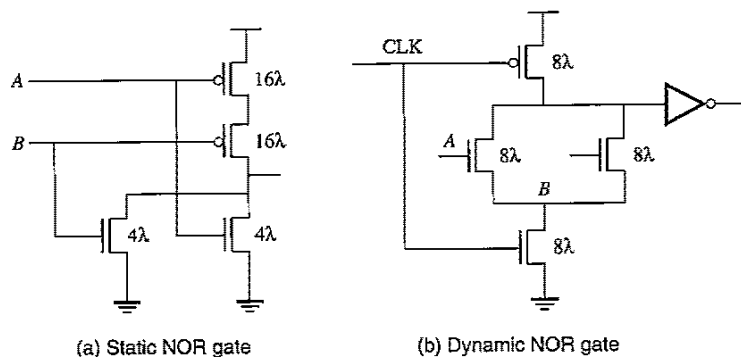


Figure 7.29
Comparison of static and dynamic logical effort.

7.5.1 Logical Effort for Domino Gates

One question to pose at this stage is whether or not domino logic is actually faster than the static CMOS gates, and by how much. One way to evaluate its timing performance is to use logical effort. For comparison, consider a static NOR gate and a dynamic NOR gate as shown in Figure 7.29. The pull-down strengths of both gates have been made equal and we assume that they are driving the same output load to conduct a fair comparison. The length of all devices is 2λ .

For the static gate, we already determined its logical effort to be $5/3$. This can be derived by comparing the input capacitance of the static NOR to a corresponding inverter. The inverter size of 8λ for the pull-up and 4λ for the pull-down produces a ratio of

$$LE_{\text{NOR}} = \frac{(\text{NOR input cap.})}{(\text{Inverter input cap.})} = \frac{16\lambda + 4\lambda}{8\lambda + 4\lambda} = 5/3$$

For the domino circuit, the input capacitance is simply 8λ since it is only connected to NMOS devices. Then,

$$LE_{\text{dyn_NOR}} = \frac{(\text{Dynamic input cap.})}{(\text{Inverter input cap.})} = \frac{8\lambda}{12\lambda} = 2/3$$

Since the extra inverter in domino logic has an $LE = 1$, the average LE for each of the two gates is the geometric mean of the product of their logical efforts (i.e., $LE_{\text{avg}} = \sqrt{(2/3)(1)} \approx 0.8$). Therefore, the domino stage is better in terms of its overall drive capability and input capacitive loading.

The results above may seem counterintuitive at first since there are two stages in the domino circuit and only one in the static NOR gate. However, the fact that only one NMOS device is driven in the domino case gives it a decided advantage in terms of input capacitance. This advantage translates into a faster overall circuit.

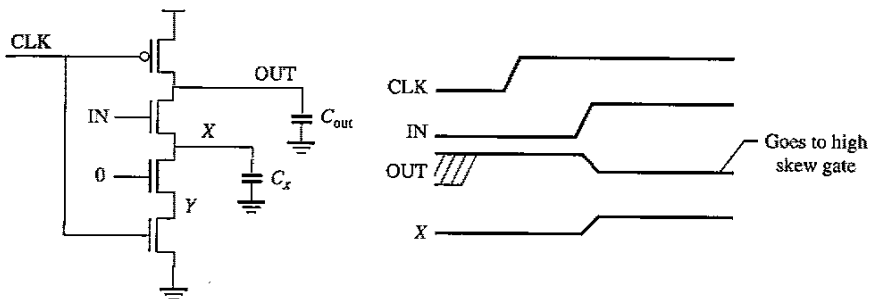


Figure 7.30
Charge-sharing example in domino logic.

Furthermore, the pull-down transistors in the domino case do not fight with the pull-up device as in the case of the static NOR gate (albeit, only for a brief period during switching). This makes the switching threshold V_s smaller for domino inputs than for static gates.

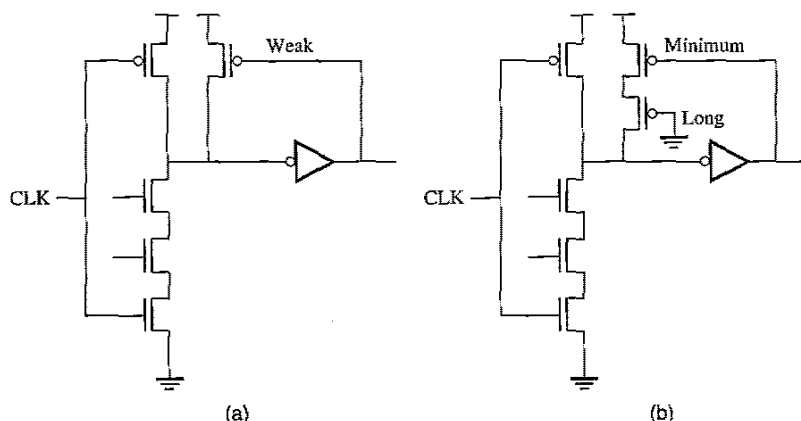
7.5.2 Limitations of Domino Logic

While the previous sections highlighted the merits of dynamic logic, there are notable disadvantages that have prevented its widespread use in industry. The main problem is the potential for logic upset due to charge loss on a capacitor, and that is not acceptable to most designers. Charge may be lost via charge sharing, noise injection due to capacitive coupling, charge leakage, or α -particle hits. Once lost, it cannot be recovered and the circuit ceases to function correctly. We address each of these issues below.

Charge sharing was described earlier in this chapter. We now apply it to the domino stage shown in Figure 7.30. Initially, CLK is low so the circuit is in the precharge phase. The OUT signal is unknown until the precharge is complete. We assume that signal X is low initially. When CLK goes high, we enter the evaluation phase. At some point, IN goes high and turns on one of the transistors connected to the output, while the other transistor in the stack remains off. As a result, we have charge sharing between the output node and the internal node X. The degree of charge sharing depends on the two capacitances C_{out} and C_x :

$$V^* = \frac{C_{out}}{C_x + C_{out}} V_{DD}$$

If they are equal, then the output value will be reduced to half its original value (i.e., $1/2 V_{DD}$). Since this value is being fed to an inverter, the implications of this degree of charge sharing is that the inverter may flip its value and produce an erroneous result. Furthermore, the fact that the inverter is skewed for timing reasons means that a smaller drop in the output voltage will cause the inverter to switch. There is

**Figure 7.31**

Minimizing the effects of charge sharing using keepers.

obviously a design tradeoff between timing and noise tolerance when sizing the inverter.

There are a number of ways to minimize charge sharing in these types of circuits. One way is to increase the output capacitance so that the final output will not vary significantly from its initial value. The input capacitance of the static inverter helps in this regard. Another approach is to precharge the internal node X to V_{DD} during the precharge phase with an additional transistor. This will require additional power and area. It will also increase the delay since there will be additional charge to remove from the internal nodes.

A third approach is to introduce “keepers” or “baby-sitters” to hold the output value high in the presence of charge sharing. In Figure 7.31a, a weak PMOS device is placed in a feedback path from the output to the internal node. After precharge, the PMOS device is on and it continues to pull up the node even after the precharge is complete. If there is any charge sharing, the output is pulled back up to V_{DD} by this device. On the other hand, if there is a path to Gnd from the output of the dynamic stage, the NMOS path must overpower the PMOS device so that the output actually switches. For this reason, we need to make the PMOS device very weak. That is, the length L of the device is made large which makes the W/L small.

The problem with this requirement is that it places a larger drive requirement on the output inverter. To circumvent this issue, two transistors can be used in the feedback path: one transistor is driven by the inverter output but it is a minimum size device so it does not place a significant load on the inverter. The second one is always on but it is a very long (and hence weak) device. The effective pull-up strength is controlled by the long device, but the capacitance seen by the inverter output is due to the small device. There is additional power dissipation, some ratioing that is required and some additional delay. Clearly, as you begin to make the

dynamic circuit more tolerant to noise, it begins to take on the characteristics of a static device.

A second issue in dynamic circuits is leakage. Both subthreshold current and reverse-bias diode leakage act to remove stored charge from any *high-impedance* nodes. However, every output node is precharged on every cycle. Perhaps a more serious issue is that of α -particle hits on dynamic nodes. These are cosmic particles that may enter the substrate and discharge capacitive nodes. Depending on the size of the capacitance, and hence the total charge on a dynamic node, it could reduce the value to a point where the stored information is lost. One way to overcome this issue is to increase the size of the capacitances, but this increases the area and power of the circuits. The keeper circuit is also useful in mitigating the effects of leakage and α -particles.

A third area of concern is noise injection from the clock or capacitive coupling to neighboring nodes. Clock feedthrough was described earlier in this chapter and this effect can be observed at the dynamic output node through the C_{GD} capacitance of the PMOS precharge device. The example below illustrates that this effect does not pose a problem. The issue of capacitive coupling to other nodes is described in more detail in the chapter on interconnect. The design approach to reducing this problem is to minimize the coupling between any dynamic node and its neighbor using either spacing or shielding techniques that are described in Chapter 10.

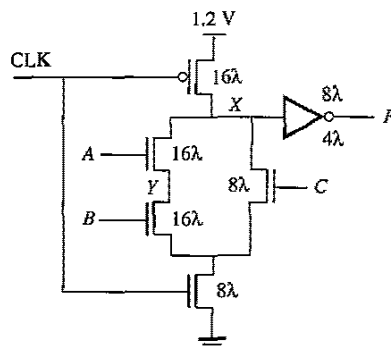
Charge Sharing and Feedthrough in Domino Logic

Example 7.8

Problem:

For the domino function below, assume 0.13 μm technology parameters and answer the following questions:

- What function does the gate perform at the output F ?
- How much clock feedthrough do we observe at the internal node X ? Is this a potential problem?
- What is the worst-case charge sharing voltage that we observe at node X ?



Solution:

- (a) The function at the output is $F = AB + C$ (a noninverting function).
 (b) Clock feedthrough is due to the clock going from low to high. The PMOS device is in the linear region after precharge. The feedthrough capacitance is

$$C_f = C_{GD} = \frac{C_g W_p}{2} = \frac{(2 \text{ fF}/\mu\text{m})(16)(0.05 \mu\text{m})}{2} = 0.8 \text{ fF}$$

The grounded capacitance at node X , assuming no source/drain sharing, is

$$\begin{aligned} C_{\text{gnd}} &= C_{\text{eff}}(16\lambda + 16\lambda + 8\lambda) + C_g(8\lambda + 4\lambda) \\ &= (1 \text{ fF}/\mu\text{m})(40)(0.05 \mu\text{m}) + (2 \text{ fF}/\mu\text{m})(12)(0.05 \mu\text{m}) \\ &= 5 \text{ fF} \end{aligned}$$

The feedthrough voltage is

$$\Delta V_x = \frac{C_f \Delta V_{\text{clk}}}{C_f + C_{\text{gnd}}} = \frac{0.8 \text{ fF}(1.2 \text{ V})}{0.8 \text{ fF} + 5 \text{ fF}} = 0.17 \text{ V}$$

Therefore the output voltage due to feedthrough is $1.2 + 0.17 \text{ V} = 1.37 \text{ V}$. Since the value is above V_{DD} , it is not a problem.

- (c) The worst-case charge sharing assumes that the output is sitting at 1.2 V , $A = 1$, $B = C = 0$. We have already determined that $C_x = 5 \text{ fF}$. The capacitance at the intermediate node between transistors A and B is

$$C_y = (1 \text{ fF}/\mu\text{m})(16\lambda) = 16(0.05) = 0.8 \text{ fF}$$

Therefore, the worst-case charge sharing is given by

$$V^* = \frac{C_x(1.2 \text{ V})}{C_x + C_y} = \frac{5(1.2 \text{ V})}{0.8 + 5} = 1.0 \text{ V}$$

This level is not possible at node Y . Set $V_y = V_{DD} - V_{TN} = 0.73 \text{ V}$. The rest of the charge remains at node X . The remaining charge is $(5 \text{ fF})(1.2 \text{ V}) - 0.8 \text{ fF}(0.73 \text{ V}) = 5.42 \text{ fC}$. This translates to a voltage of 1.08 V at node X .

7.5.3 Dual-Rail (Differential) Domino Logic

A popular form of domino logic is called *dual-rail* or *differential* domino logic. It is also known as differential cascode voltage switch (DCVS) logic. Since standard domino can only implement noninverting logic functions, the dual-rail approach has been adopted by many microprocessor designers to overcome this limitation. The basic structure is shown in Figure 7.32. Rather than having only one output node, a complementary pair of logic outputs are generated. There are two precharge transistors that force the outputs low when CLK is low. When CLK goes high, either OUT or $\overline{\text{OUT}}$ will go high, depending on the logic blocks, $\overline{F}$ and F , respectively. Of course, both outputs cannot go high during the evaluate phase.

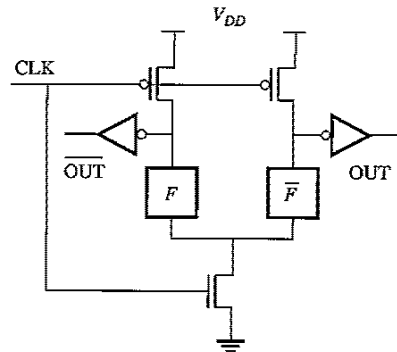


Figure 7.32
Structure of dual-rail domino logic.

An example of a dual-rail AND/NAND gate is shown in Figure 7.33. One side of the gate implements the AND function while the opposite side implements the NAND function.

This configuration has a number of problems similar to the standard domino gate. The internal nodes are susceptible to charge sharing. Therefore, a pair of *keeper* devices are needed to maintain high levels if the clock is high for a long period of time. The keeper devices are shown in Figure 7.34. It is clear that the number of devices needed to implement the gate has increased significantly from the modest levels that we expected when first considering dynamic logic. Furthermore, the power dissipation of these gates is much higher than the conventional CMOS gate or the standard domino gate since one side will always be precharged high and then

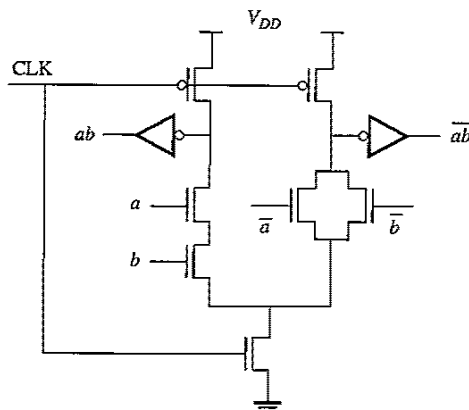


Figure 7.33
Dual-rail domino AND/NAND function.

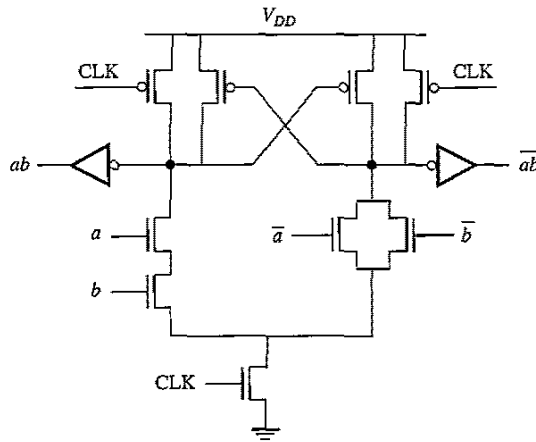


Figure 7.34
Dual-rail domino with keeper devices.

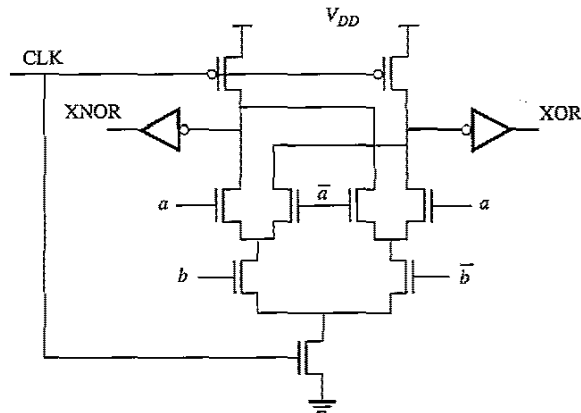
go low. This is a significant price to pay for the extra output. However, we should note that not every logic gate requires a complement function. Only those functions that explicitly require a complementary term would require the use of dual-rail gates. The rest can be implemented as standard domino gates.

Example 7.9 Dual-Rail XOR/XNOR Domino Gate

Problem:

Design a dual-rail XOR/XNOR domino gate and share as many transistors as possible between the true and complement logic blocks.

Solution:



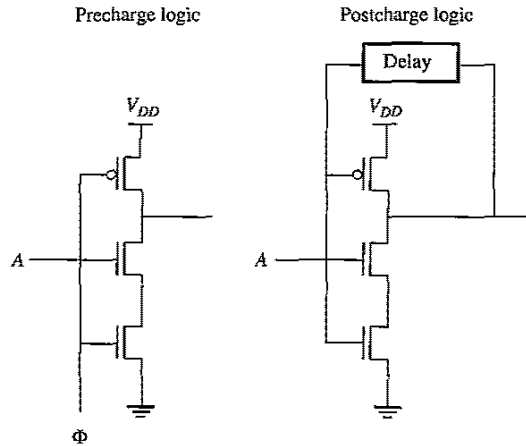


Figure 7.35
Precharge and postcharge circuits.

7.5.4 Self-Resetting Circuits

The routing of the clock to domino gates presents a problem to CAD tools and introduces issues of delay and skew into the circuit design process. There are situations that permit the use of circuits that can automatically precharge themselves (i.e., reset themselves) after a prescribed delay. These are called postcharge or self-resetting logic circuits. They find applications where a small percentage of gates switch in each cycle, such as memory decoders (described in Chapter 10). Here we will examine the basic operation of these circuits.

Figure 7.35 shows the standard precharge configuration and the postcharge configuration. In the domino case, the clock is used to operate the circuit. In the self-resetting case, the output is fed back to the precharge control input and, after a specified time delay, the pull-up is reactivated. The delay line is implemented as a series of inverters. The signals that propagate through these circuits are pulses. The width of the pulses must be controlled carefully or else there may be contention between NMOS and PMOS devices, or even worse, oscillations may occur. For example, if A is high for more than twice the delay around the loop, the circuit will oscillate. Therefore, special care must be taken to ensure correct timing.

7.6 Summary

Capacitive feedthrough and bootstrapping equation:

$$\Delta V_2 = \frac{C_f \Delta V_1}{C_f + C_{\text{gnd}}}$$

Charge-sharing equation:

$$V^* = \frac{(C_1 V_1 + C_2 V_2)}{C_1 + C_2}$$

Large-signal resistance for CMOS transmission gate:

$$R_{on} \approx \frac{V_{DS}}{I_{DS}}$$

$$R_{TG} = R_N // R_P \approx R_{eqn} \left(\frac{L}{W} \right)$$

REFERENCES

1. I. Sutherland, B. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufman Publishers, San Francisco, CA, 1999.
2. J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Designer Perspective*, Second Edition, Prentice-Hall, Upper Saddle River, NJ, 2003.
3. S. M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits, Analysis and Design*, Third Edition, McGraw-Hill, New York, NY, 2003.
4. H. Veendrick, *Deep-Submicron CMOS ICs*, Second Edition, Kluwer Academic Publishers, Boston, MA, 2000.
5. J. P. Uyemura, *CMOS Logic Circuit Design*, Kluwer Academic Publishers, Boston, MA, 1999.

PROBLEMS

- P7.1. Compute all the voltages at all the nodes for the pass transistor circuit of Figure P7.1 including the body effect. Use 0.13 μm technology parameters. Assume that all nodes are initially at 0 V.

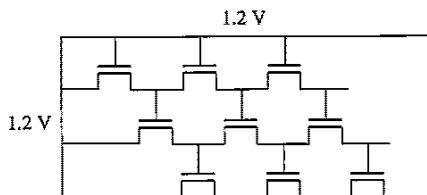


Figure P7.1

P7.2. Provide a truth table that describes the functionality of each circuit in Figure P7.2. (Note: In some cases, the output of the circuit may be high-Z.) The symbol A' refers to the complement of A .

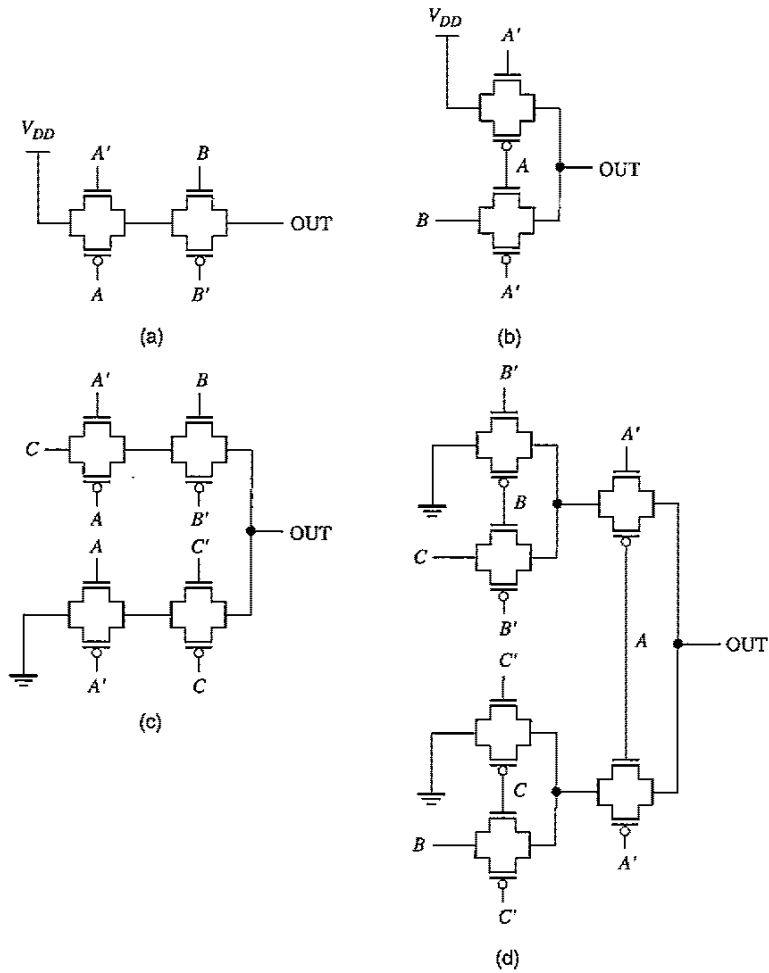


Figure P7.2

- P7.3. (a) For the dynamic D-latch shown in Figure P7.3a, estimate the output voltages for Q and $\bar{Q}$ given that the input is at V_{DD} when CLK goes high (V_{DD}). Assume that $W = 200$ nm and $V_{DD} = 1.8$ V. Then, recompute the values when the CLK goes low. Include the effects of clock feedthrough.

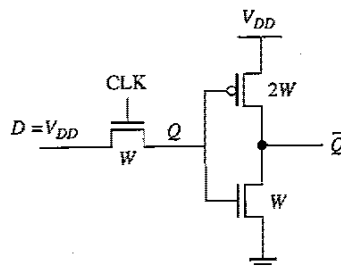


Figure P7.3a

- (b) Repeat part (a) for Figure P7.3b. Then, compute the worst-case delay through the D-latch when the clock input goes high. Include all capacitance. Assume $W = 200$ nm.

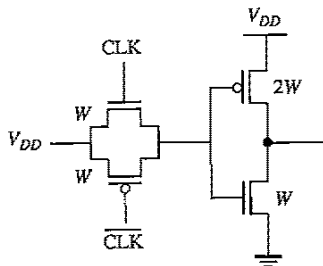


Figure P7.3b

- P7.4. Using transmission gates, design a circuit whose output is

- (a) $\text{Out} = A + BC$
 (b) $\text{Out} = AB + BC + \bar{C}$
 (c) $\text{Out} = \overline{(A + B + C)} + \bar{A}B$
 (d) $\text{Out} = \overline{\overline{(A + B + C)} + \bar{A}B}$

Then remove the transistors and switches that are not necessary.

P7.5. What is the output of each of the circuits in Figure P7.5?

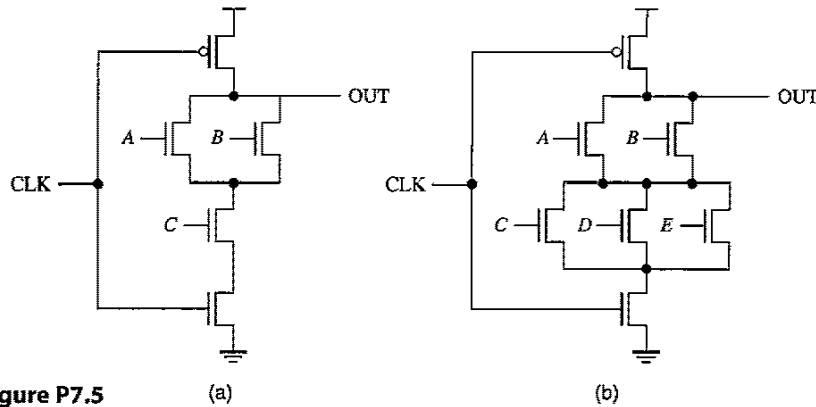


Figure P7.5

P7.6. Design a domino circuit whose output is

(a) $\text{Out} = \bar{A} + \bar{B}C$

(b) $\text{Out} = A\bar{B} + BC + \bar{C}$

(c) $\text{Out} = \overline{(\bar{A} + \bar{B} + C)} + A\bar{B}$

(d) $\text{Out} = \overline{((A + \bar{B} + C) + \bar{A}B)}$

P7.7. What is the LE (logic effort) of each input signal of the circuits of Figure P7.7? (All transistors are minimum size, $L = 2\lambda$ and $W = 2\lambda$, unless specified.)

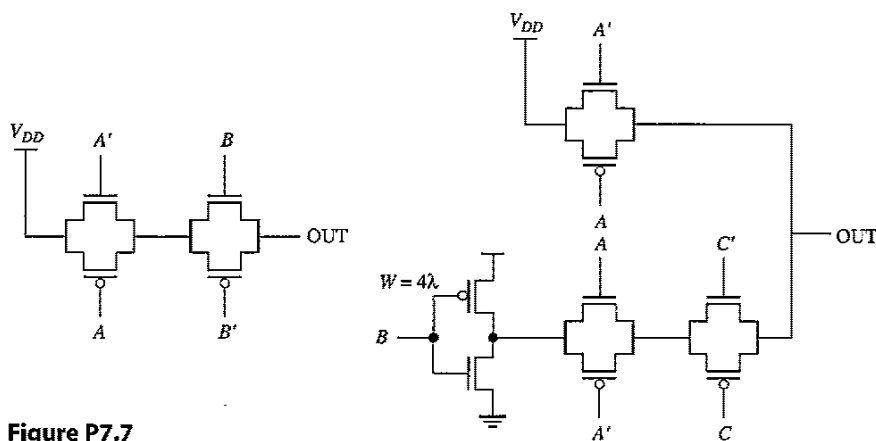


Figure P7.7

- P7.8.** For the circuit shown in Figure P7.8, assume that the transmission gates are all $4\lambda:2\lambda$ and that the inverters driving the transmission gates have PMOS transistors that are $8\lambda:2\lambda$, and NMOS transistors that are $4\lambda:2\lambda$, where $\lambda = 0.1 \mu\text{m}$. The output inverter is to drive a 50 fF load. The output inverter is f times larger than the input inverters.

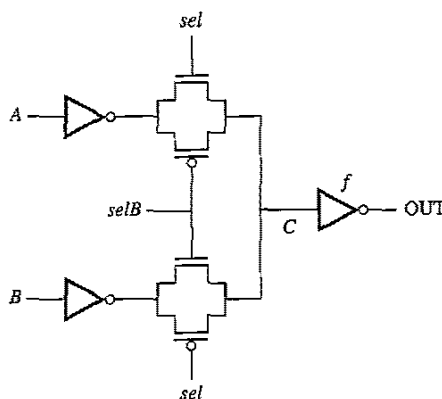


Figure P7.8

- Write the expression for the output function in terms of A , B , sel , and $selB$.
 - Draw an equivalent RC circuit model for the path from A to C assuming that the sel signal is high. Write down the individual contributions for each resistance and capacitance and place the total values at the appropriate nodes.
 - What is the expression for the delay from A to C ? Use Elmore delay here.
 - Write down the expression for the delay through the output inverter knowing that it is f times larger than the input inverters. Use an RC delay here.
 - Determine the optimal size of the output inverter to minimize the total delay from A to OUT . That is, write the complete expression for the delay and then derive the best value of f that would produce the smallest delay.
- P7.9.** What is the logical effort of the circuits of Figure P7.9 ($W_P = 8\lambda$, $W_N = 12\lambda$)? Consider the fall transition in each case.

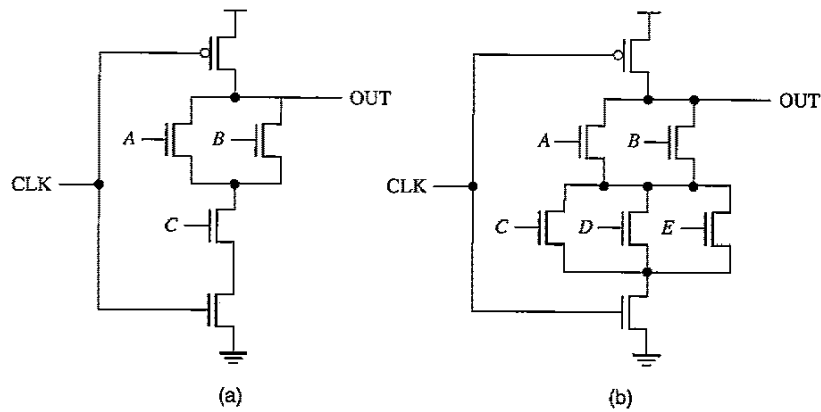


Figure P7.9

P7.10. In the circuit of Figure P7.10, determine the capacitance of each node and calculate the minimum and maximum delay. (All transistors are minimum size, $L = 2\lambda$ and $W = 2\lambda$, unless otherwise specified.)

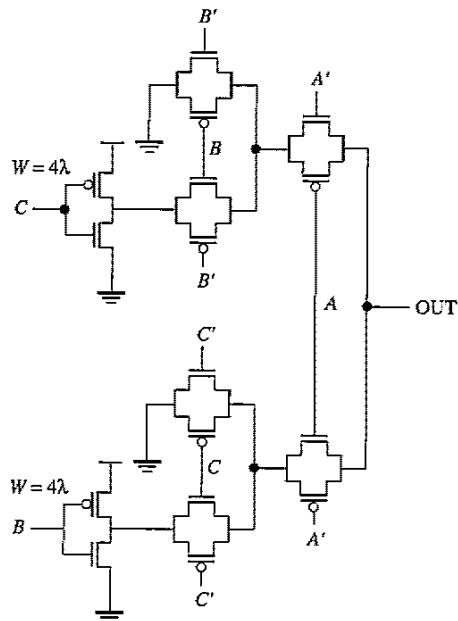


Figure P7.10

P7.11. The precharged circuit of Figure P7.11 is a 2-input AND gate. Draw the waveforms at nodes F , X , and Y as a function of time. You can assume short rise and fall times (i.e., you do not need to calculate them), but the final value in each phase must be correct. Only the capacitances shown here need to be considered. All devices are minimum size. Both F and Y start out unknown. Use $0.13\ \mu\text{m}$ parameters.

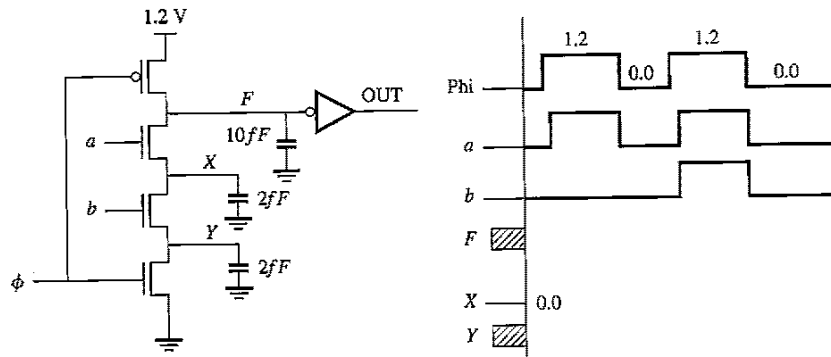


Figure P7.11

P7.12. What is the effect of temperature on static and dynamic logic circuits in terms of speed, power, and leakage current?

P7.13. Shown in Figure P7.13 is a dynamic logic gate in which the precharge device has a W/L of 5, the n -channel devices are all 3, and the inverter is a skewed inverter with a pull-up of 4 and a pull-down of 1 to reduce the delay.

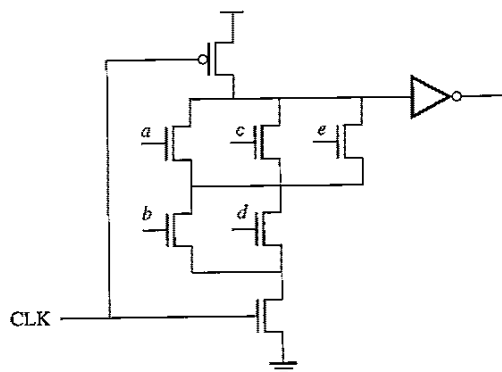


Figure P7.13

- (a) What input settings give you the worst-case charge sharing?
- (b) What is the amount of charge sharing in the worst-case?
- (c) Does this circuit fail in the worst-case charge sharing situation? Explain.

P7.14. The two CMOS circuits in Figure P7.14 are intended to be simple latches. Do these gates work as latches? Are there any problems with using them as latches? Is one better than the other? Explain.

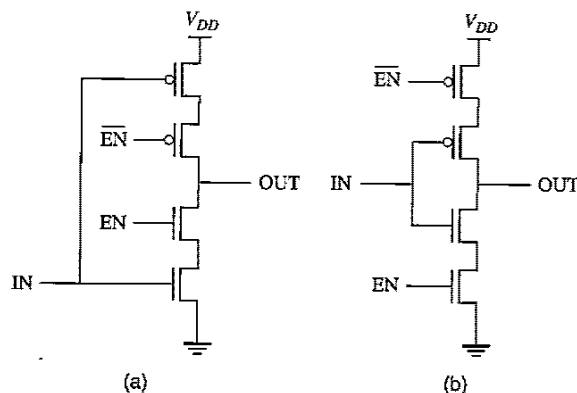


Figure P7.14

P7.15. The circuit shown in Figure P7.15 is a bootstrapped inverter. It uses a linear enhancement load, M_2 , with an addition transistor, M_3 , to provide gate drive to M_2 . A bootstrapping capacitor, C_b , is used to pull node X up when the output is being pulled up. Answer the following questions in terms of the quantities V_{DD} , V_T , V_{OL} , C_x , C_{out} , and C_b .

- (a) When the input is V_{DD} , what is the output voltage and the voltage at node X ?
- (b) When the input switches to a low voltage, what is the required value of C_b that allows the output to reach V_{DD} ?

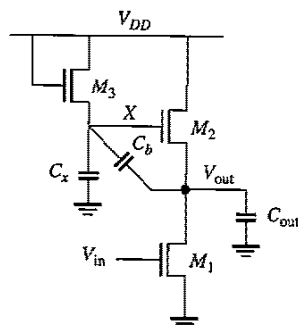


Figure P7.15

5

Semiconductor Memory Design

CHAPTER OUTLINE

- 8.1 Introduction
- 8.2 MOS Decoders
- 8.3 Static RAM Cell Design
- 8.4 SRAM Column I/O Circuitry
- 8.5 Memory Architecture
- 8.6 Summary

References

Problems

8.1 Introduction

Modern digital systems require the capability of storing and retrieving large amounts of information at high speeds. *Memories* are circuits or systems that store digital information in large quantity. This chapter addresses the analysis and design of VLSI memories, commonly known as *semiconductor memories*. Today, memory circuits come in different forms including SRAM, DRAM, ROM, EPROM, E²PROM, Flash, and FRAM. While each form has a different cell design, the basic structure, organization, and access mechanisms are largely the same. In this chapter, we classify the different types of memory, examine the major subsystems, and focus on the static RAM design issues. This topic is particularly suitable for our study of CMOS digital design as it allows us to apply many of the concepts presented in earlier chapters.

Recent surveys indicate that roughly 30% of the worldwide semiconductor business is due to memory chips. Over the years, technology advances have been driven by memory designs of higher and higher density. Electronic memory capacity in digital systems ranges from fewer than 100 bits for a simple function to standalone chips containing 256 Mb (1 Mb = 2¹⁰ bits) or more.¹ Circuit designers usually

¹ Recently, a memory chip with 1 Gbit of data storage capacity has been announced.

speak of memory capacities in terms of bits, since a separate flip-flop or other similar circuit is used to store each bit. On the other hand, system designers usually state memory capacities in terms of *bytes* (8 bits); each byte represents a single alphanumeric character. Very large scientific computing systems often have memory capacity stated in terms of *words* (32 to 128 bits). Each byte or word is stored in a particular location that is identified by a unique numeric *address*. Memory storage capacity is usually stated in units of kilobytes (K bytes) or megabytes (M bytes). Because memory addressing is based on binary codes, capacities that are integral powers of 2 are most common. Thus the convention is that, for example, 1K byte = 1,024 bytes and 64K bytes = 65,536 bytes. In most memory systems, only a single byte or word at a single address is stored or retrieved during each cycle of memory operation. *Dual-port* memories are also available that have the ability to read/write two words in one cycle.

8.1.1 Memory Organization

The preferred organization for most large memories is shown in Figure 8.1. This organization is a *random-access* architecture. The name is derived from the fact that memory locations (*addresses*) can be accessed in random order at a fixed rate, independent of physical location, for reading or writing. The storage array, or *core*, is made up of simple cell circuits arranged to share connections in horizontal rows and vertical columns. The horizontal lines, which are driven only from outside the storage array, are called *wordlines*, while the vertical lines, along which data flow into and out of cells, are called *bitlines*.

A cell is accessed for reading or writing by selecting its row and column. Each cell can store 0 or 1. Memories may simultaneously select 4, 8, 16, 32, or 64 columns

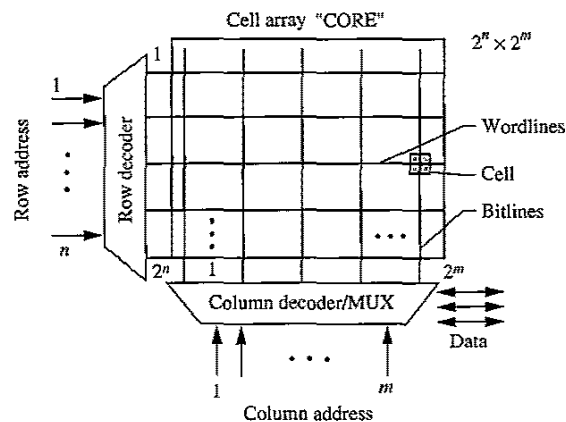


Figure 8.1
Organization of memory systems.

in one row depending on the application. The row and column (or group of columns) to be selected are determined by decoding binary address information. For example, an n -bit *decoder* for row selection, as shown in Figure 8.1, has 2^n output lines, a different one of which is enabled for each different n -bit input code. The column decoder takes m inputs and produces 2^m bitline access signals, of which 1, 4, 8, 16, 32, or 64 may be enabled at one time. The bit selection is done using a multiplexer circuit to direct the corresponding cell outputs to data registers. In total, $2^n \times 2^m$ cells are stored in the core array.

An overall architecture of a 64 Kb random-access memory is shown in Figure 8.2. For this example, $n = m = 8$. Therefore, the core array has a total of 65,536 cells. The memory uses a 16-bit address to produce a single bit output.

Memory cell circuits can be implemented in a wide variety of ways. In principle, the cells can be based on the flip-flop designs listed in Chapter 5 since their intended function is to store bits of data. However, these flip-flops require a substantial amount of area and are not appropriate when millions of cells are needed.

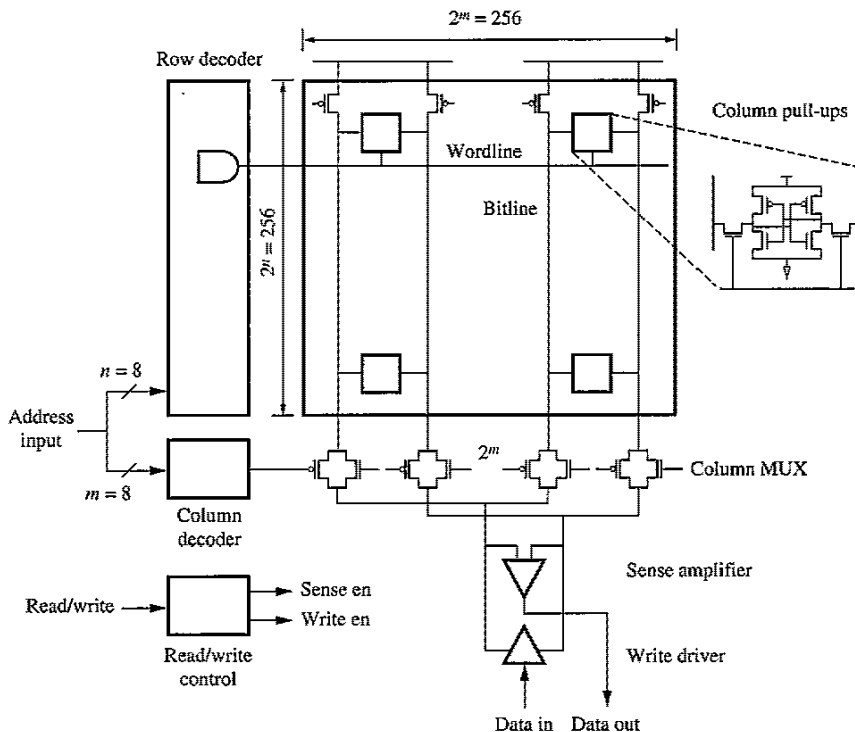


Figure 8.2
Overall architecture of memory design.

In fact, most memory cell circuits are greatly simplified compared to register and flip-flop circuits. While the data storage function is preserved, other properties including quantization of amplitudes, regeneration of logic levels, input-output isolation, and fanout drive capability may be sacrificed for cell simplicity. In this way, the number of devices in a single cell can be reduced to one to six transistors. Figure 8.2 illustrates a six-transistor memory cell.

At the level of a memory chip shown in Figure 8.2, the desired logic properties are recovered through use of properly designed *peripheral circuits*. Circuits in this category are the decoders, sense amplifiers, column precharge, data buffers, etc. These circuits are designed so that they may be shared among many memory cells. Read-write (R/W) circuits determine whether data are being retrieved or stored, and they perform any necessary amplification, buffering, and translation of voltage levels. Specific examples are presented in the following sections.

8.1.2 Types of Memory

Read-write random-access memories (RAM) may store information in flip-flop style circuits, or simply as charge on capacitors. Approximately equal delays are encountered in reading or writing data. Because read-write memories store data in active circuits, they are *volatile*; that is, stored information is lost if the power supply is interrupted. The natural abbreviation for read-write memory would be RWM. However, pronunciation of this acronym is difficult. Instead, the term RAM is commonly used to refer to read-write random-access memories. If the terms were consistent, both read-only (see below) and read-write memories would be called RAMs.

The two most common types of RAMs are the static RAM (SRAM) and the dynamic RAM (DRAM). The static and dynamic definitions are based on the same concepts as those introduced in earlier chapters. Static RAMs hold the stored value in flip-flop circuits as long as the power is on. SRAMs tend to be high-speed memories with clock cycles in the range of 5 to 50 ns. Dynamic RAMs store values on capacitors. They are prone to noise and leakage problems, and are slower than SRAMs, clocking at 50 ns to 200 ns. However, DRAMs are much more dense than SRAMs—up to four times more dense in a given generation of technology.

Read-only memories (ROMs) store information according to the presence or absence of transistors joining rows to columns. ROMs also employ the organization shown in Figure 8.1 and have read speeds comparable to those for read-write memories. All ROMs are *nonvolatile*, but they vary in the method used to enter (write) stored data. The simplest form of ROM is programmed when it is manufactured by formation of physical patterns on the chip; subsequent changes of stored data are impossible. These are termed *mask-programmed* ROMs. In contrast, *programmable* read-only memories (PROMs) have a data path present between every row and column when manufactured, corresponding to a stored 1 in every data position. Storage cells are selectively switched to the 0 state once after manufacture by applying appropriate electrical pulses to selectively open (blow out) row-column data paths. Once programmed, or *blown*, a 0 cannot be changed back to a 1.

Erasable programmable read-only memories (EPROMs) also have all bits initially in one binary state. They are programmed electrically (similar to the PROM), but all bits may be erased (returned to the initial state) by exposure to ultraviolet (UV) light. The packages for these components have transparent windows over the chip to permit the UV irradiation. *Electrically erasable* programmable read-only memories (EEPROMs, E²PROM, or E-squared PROMs) may be written and erased by electrical means. These are the most advanced and most expensive form of PROM. Unlike EPROMs, which must be totally erased and rewritten to change even a single bit, E²PROMs may be selectively erased. Writing and erasing operations for all PROMs require times ranging from microseconds to milliseconds. However, all PROMs retain stored data when power is turned off; thus they are termed non-volatile.

A recent form of EPROM and E²PROM is termed Flash memory, a name derived from the fact that blocks of memory may be erased simultaneously. Flash memory of the EPROM form is written using the hot-electron effect² whereas E²PROM Flash is written using Fowler-Nordheim (FN) tunneling.³ Both types are erased using FN tunneling. Their large storage capacity has made this an emerging mass storage medium. In addition, these types of memories are beginning to replace the role of ROMs on many chips, although additional processing is required to manufacture Flash memories in a standard CMOS technology.

Memories based on ferroelectric materials, so-called FRAMs or FeRAMs, can also be designed to retain stored information when power is off. The *Perovskite crystal* material used in the memory cells of this type of RAM can be polarized in one direction or the other to store the desired value. The polarization is retained even when the power supply is removed, thereby creating a nonvolatile memory. However, semiconductor memories are preferred over ferroelectric memories for most applications because of their advantages in cost, operating speed, and physical size. Recently, FRAMs have been shown to be useful nonvolatile memory in certain applications such as smart cards and may be more attractive in the future due to their extremely high storage density.

8.1.3 Memory Timing Parameters

The timing signals of random-access memories are illustrated in Figure 8.3. At a high level, the main inputs are the $(n + m)$ -bit address, the k -bit input data, and the *read/write* signal (write operations are active low in this example). The output is the k -bit value that was stored in the memory location associated with the $(n + m)$ -bit address. The *read access time*, t_{AC} , is the delay from presentation of an address until data stored at that address are available at the output. Maximum read access time is an important parameter, and it should not exceed the memory cycle time since

² Hot electrons are created by applying a high field in the channel region. These electrons enter the oxide and raise the threshold voltage of a device. Devices with this higher threshold voltage are viewed as a stored "1." Devices with the lower threshold voltage represent a stored "0."

³ Fowler-Nordheim tunneling occurs through thin insulating material such as thin-oxide associated with the gate. Current flows through the oxide by tunneling through the energy barrier.

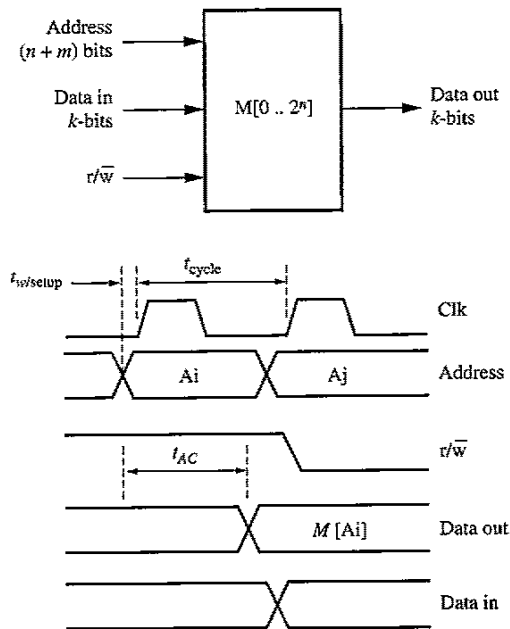


Figure 8.3
Random access memory timing parameters.

there are write setup operations needed before each memory operation, indicated by $t_{w/setup}$. The memory clock cycle time, t_{cycle} , is the minimum time needed to complete successive read or write operations.

The cycle time is essentially the reciprocal of the time rate at which address information is changed while reading or writing at random locations. Minimum access times for reading and writing are not necessarily the same, but for simplicity of design, most systems specify a single time for both reading and writing. For semiconductor read-write memories, the read access time is typically 50 to 80% of cycle time.

8.2 MOS Decoders

The row and column decoders identified in Figure 8.1 are essential elements in all random-access memories. Access time and power consumption of memories may be largely determined by decoder design. Similar designs are used in read-only and read-write applications. Row decoders take an n -bit address and produce 2^n outputs, one of which is activated. Obviously, the one that is activated depends directly on the address applied to the memory. Figure 8.4 shows two forms of decoders: AND and NOR decoders. The decoder can be implemented using AND gates or

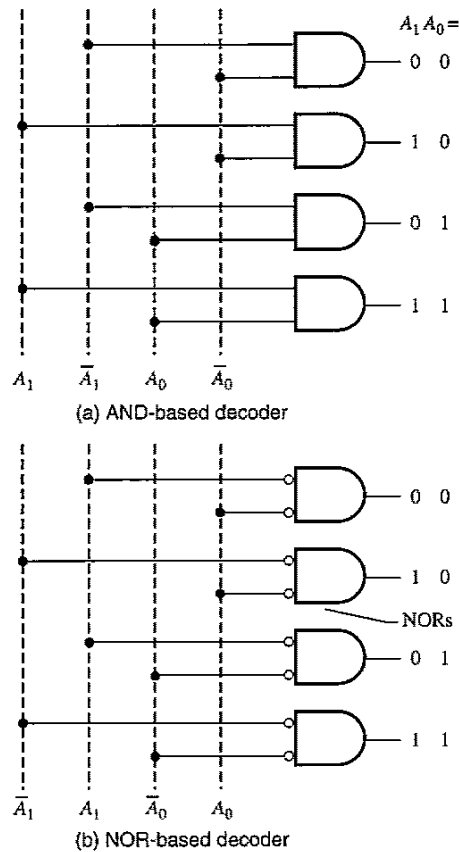


Figure 8.4
AND and NOR decoders.

NOR gates that take every possible combination of the inputs. There are two address bits in this example and we require both the true and complement of each address bit. The output line activated by each input combination is shown in the figure. Note that all outputs are normally low except one.

An n -bit decoder requires 2^n logic gates, each with n inputs. For example, with $n = 6$, we need 64 NAND6 gates driving 64 inverters to implement the decoder. From previous chapters, it is clear that gates with more than 3 or 4 inputs create large series resistances and long delays. Rather than using n -input gates, it is preferable to use a cascade of gates. Typically two stages are used: a predecode stage and a final decode stage. The predecode stage generates intermediate signals that are used by multiple gates in the final decode stage. In Figure 8.5, schematics are shown for two possible alternatives to implement a 6-input AND gate. We could choose three

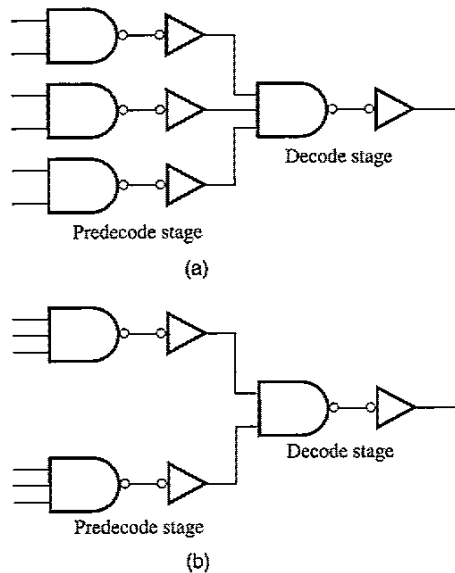


Figure 8.5
Predecoder configurations.

NAND2 gates and one NAND3 gate to implement each AND6 gate, as shown in Figure 8.5a. Alternatively, two NAND3 gates and one NAND2 gate of Figure 8.5b may be used. The better of the two can be determined using logical effort.

The main advantage of two-level decoding is that a large number of intermediate signals can be generated by the predecode stage and then reused by the final decoding stage. The result is a reduction in the number of inputs for each gate. Since this aspect is not clearly depicted in Figure 8.5, a more complete example for 6 address bits is shown in Figure 8.6. In the predecoder, a total of 12 intermediate signals are generated from the address bits and their complements. The signals that are generated in the predecoder are as follows: A_0A_1 , $A_0\bar{A}_1$, $\bar{A}_0A_1$, $\bar{A}_0\bar{A}_1$, A_2A_3 , $A_2\bar{A}_3$, etc. These signals may now be used by the final decoding stage to generate the 64 required outputs using NAND3/inverter combinations. This corresponds to the configuration shown in Figure 8.5a. Each predecoder output drives 16 NAND gates (i.e., 64 NAND3 gates $\times$ 3 inputs each / 12 intermediate outputs). Therefore, the branching effort $BE = 16$. The delay through the NAND2-inverter-NAND3-inverter stages can be minimized by sizing the gates using logical effort. A similar kind of two-level decoder can be constructed using the configuration of Figure 8.5b. Again, the better of the two approaches can be determined using logical effort. It is important to minimize the delay through the decoder as it may constitute up to 40% of the clock cycle.

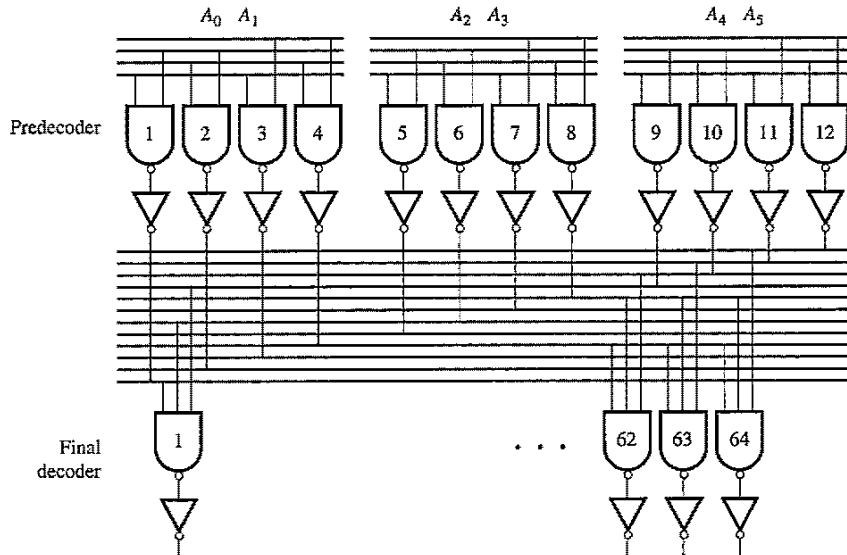


Figure 8.6
Structure of two-level decoder for 6-bit address.

Decoder Sizing Using Logical Effort Techniques

Example 8.1

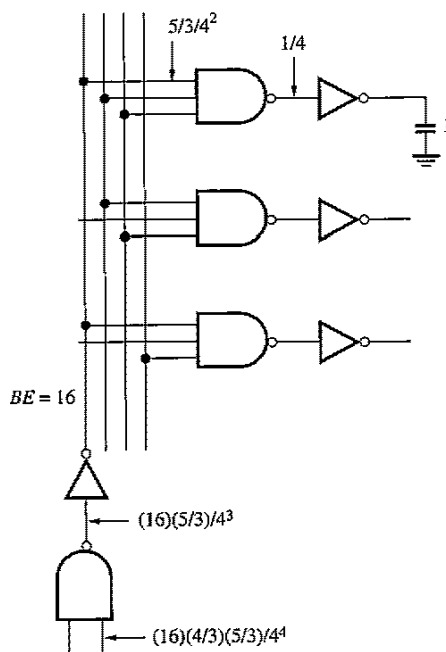
Problem:

Size the decoder of Figure 8.6 using FO4 rules assuming that the normalized output loading is 1. FO4 rules imply that the optimal stage effort is equal to 4.

$$\text{For each stage, } C_{in} = \frac{LE \times BE \times C_{out}}{4}$$

Solution:

Work backwards from the output to the input, taking into account logical effort and branching effort of each stage. First, the normalized output has a size of 1. Therefore, the inverter input capacitance is $1/4$ using FO4 sizing rules. The LE of the NAND3 gate is $5/3$. Therefore, its input capacitance is $(5/3)/4^2$. The branching effort at the output of the predecode stage is 16. Therefore, the input capacitance of the inverter is $(16)(5/3)/4^3$. Finally, the NAND2 gate has an input capacitance of $(16)(4/3)(5/3)/4^4$.



8.3 Static RAM Cell Design

Memories are said to be static if no periodic clock signals are required to retain stored data indefinitely. Memory cells in these circuits have a direct path to V_{DD} or Gnd or both. Read-write memory cell arrays based on flip-flop circuits are commonly referred to as static RAMs or SRAMs. The design issues for static memory cells are described in this section.

8.3.1 Static Memory Operation

The basic static RAM cell is shown in Figure 8.7a. It consists of two cross-coupled inverters and two access transistors. The access transistors are connected to the wordline at their respective gate terminals, and the bitlines at their source/drain terminals. The wordline is used to select the cell while the bitlines are used to perform read or write operations on the cell. Internally, the cell holds the stored value on one side and its complement on the other side. For reference purposes, assume that node q holds the stored value while node $\bar{q}$ holds its complement. The two complementary bitlines are used to improve speed and noise rejection properties, as described later in this chapter.

The VTC of cross-coupled inverters is shown in Figure 8.7b. This type of VTC was already described in Chapter 4 on single source noise margins, and in Chapter 5

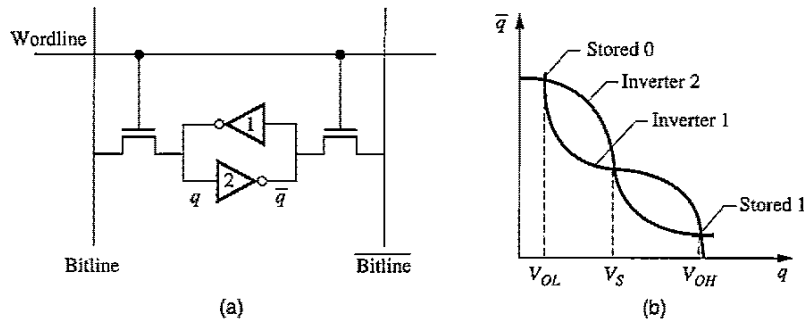


Figure 8.7
Basic SRAM cell and VTC.

on flip-flops. The VTC conveys the key cell design considerations for read and write operations. In the cross-coupled configuration, the stored values are represented by the two stable states in the VTC. The cell will retain its current state until one of the internal nodes crosses the switching threshold, V_S . When this occurs, the cell will flip its internal state. Therefore, during a read operation, we must not disturb its current state, while during the write operation we must force the internal voltage to swing past V_S to change the state.

The six transistor (6T) static memory cell in CMOS technology is illustrated schematically in Figure 8.8. The cross-coupled inverters, M_1 , M_5 and M_2 , M_6 , act as the storage element. Major design effort is directed at minimizing the cell area and power consumption so that millions of cells can be placed on a chip. The steady-state power consumption of the cell is controlled by subthreshold leakage currents, so a larger threshold voltage is often used in memory circuits. To reduce area, the cell layout is highly optimized to eliminate all wasted area. In fact, some designs replace the load devices, M_5 and M_6 , with resistors formed in undoped polysilicon.

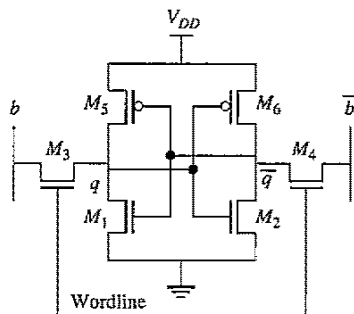


Figure 8.8
6T SRAM cell.

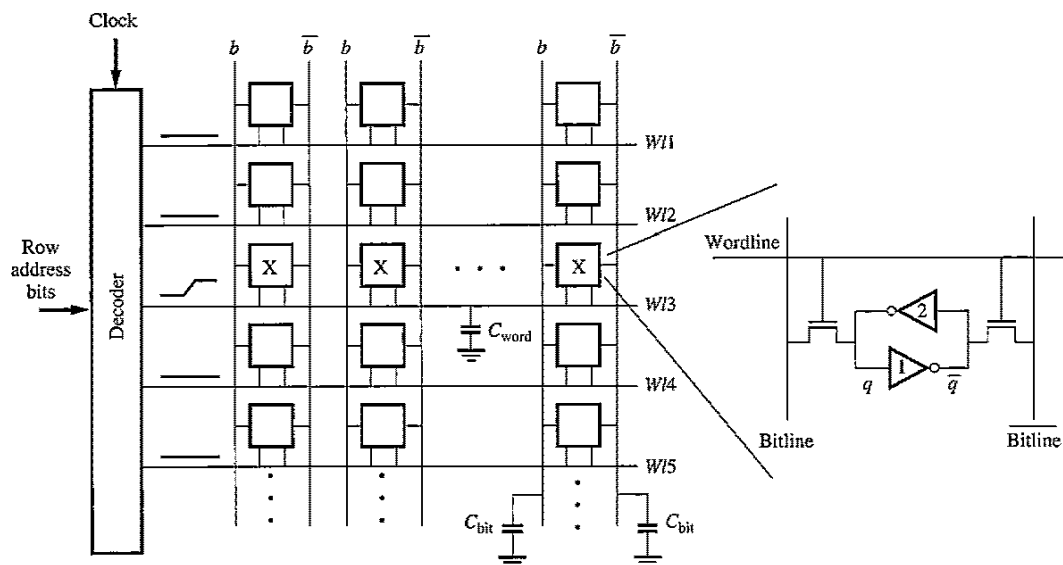


Figure 8.9
Wordline and double bitline configuration.

This is called a 4T cell since there are now only four transistors in the cell. To minimize power, the current through the resistors can be made extremely small by using very large pull-up resistances. Sheet resistance of these resistors is $10 \text{ M}\Omega$ per square or higher and the area is minimal. Standby currents are kept in the nanoampere range. Thus, power and area may be reduced at the expense of extra processing complexity to form the undoped polysilicon resistors. However, the majority of the designs today use the conventional 6T configuration of Figure 8.8.

The operation of an array of these cells is illustrated in Figure 8.9. The row select lines, or wordlines, run horizontally. All cells connected to a given wordline are accessed for reading or writing. The cells are connected vertically to the bitlines using the pair of access devices to provide a switchable path for data into and out of the cell. Two column lines, b and $\bar{b}$, provide a differential data path. In principle, it should be possible to achieve all memory functions using only one column line and one access device. Attempts have been made in this direction, but due to normal variations in device parameters and operating conditions, it is difficult to obtain reliable operation at full speed using a single access line. Therefore, the symmetrical data paths b and $\bar{b}$ as shown in Figure 8.9 are almost always used.

Row selection in CMOS memory is accomplished using the decoders described in the previous section. For synchronous memories, a clock signal is used in conjunction with the decoder to activate a row only when read-write operations are being performed. At other times, all wordlines are kept low. When one wordline goes high, say $w13$ in Figure 8.9, all the cells in that row are selected. The access transistors

are all turned on and a read or write operation is performed. Cells in other rows are effectively disconnected from their respective wordlines.

The wordline has a large capacitance, C_{word} , that must be driven by the decoder. It is comprised of two gate capacitances per cell and the wire capacitance per cell:

$$C_{\text{word}} = (2 \times \text{gate cap} + \text{wire cap}) \times \text{no. of cells in row} \quad (8.1)$$

Once the cells along the wordline are enabled, read or write operations are carried out. For a read operation, only one side of the cell draws current. As a result, a small differential voltage develops between b and $\bar{b}$ on all column lines. The column address decoder and multiplexer select the column lines to be accessed. The bitlines will experience a voltage difference as the selected cells discharge one of the two bitlines. This difference is amplified and sent to output buffers.

It should be noted that the bitlines also have a very large capacitance due to the large number of cells connected to them. This is primarily due to source/drain capacitance, but also has components due to wire capacitance and drain/source contacts. Typically, a contact is shared between two cells. The total bitline capacitance, C_{bit} , can be computed as follows:

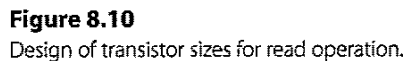
$$C_{\text{bit}} = (\text{source/drain cap} + \text{wire cap} + \text{contact cap}) \times \text{no. of cells in column} \quad (8.2)$$

During a write operation, one of the bitlines is pulled low if we want to store 0, while the other one is pulled low if we want to store 1. The requirement for a successful write operation is to swing the internal voltage of the cell past the switching threshold of the corresponding inverter. Once the cell has flipped to the other state, the wordline can be reset back to its low value.

The design of the cell involves the selection of transistor sizes for all six transistors of Figure 8.8 to guarantee proper read and write operations. Since the cell is symmetric, only three transistor sizes need to be specified, either M_1 , M_3 , and M_5 or M_2 , M_4 , and M_6 . The goal is to select the sizes that minimize the area, deliver the required performance, obtain good read and write stability, provide good cell read current, and have good soft error immunity (especially due to α -particles).

8.3.2 Read Operation

We now describe the design details of the 6T RAM cell for the read operation using Figure 8.10. Assume that a "0" is stored on the left side of the cell, and a "1" on the right side. Therefore, M_1 is on and M_2 is off. Initially, b and $\bar{b}$ are precharged to a high voltage around V_{DD} by a pair of column pull-up transistors (not shown). The row selection line, held low in the standby state, is raised to V_{DD} which turns on access transistors M_3 and M_4 . Current begins to flow through M_3 and M_1 to ground, as shown in Figure 8.10a. The resulting cell current slowly discharges the capacitance C_{bit} . Meanwhile, on the other side of the cell, the voltage on $\bar{b}$ remains high since there is no path to ground through M_2 . The difference between b and $\bar{b}$ is fed to a sense amplifier to generate a valid low output, which is then stored in a data buffer.



When designing the transistor sizes for read stability, we must ensure that the stored values are not disturbed during the read cycle. The problem is that, as current flows through M_3 and M_1 , it raises the output voltage at node q which could turn on M_2 and bring down the voltage at node $\bar{q}$, as shown in Figure 8.10b. The voltage at node $\bar{q}$ may drop a little but it should not fall below V_S . To avoid altering the state of the cell when reading, we must control the voltage at node q by sizing M_1 and M_3 appropriately. We can accomplish this by making the conductance of M_1 about 3 to 4 times that of M_3 so that the drain voltage of M_1 does not rise above V_{TN} . In theory, the voltage should not exceed V_S , but this design must be carried out with due consideration of process variations and noise. In effect, the read stability requirement establishes the ratio between the two devices.

The other consideration in the read cycle design is to provide enough cell current to discharge the bitline sufficiently within 20 to 30% of the cycle time. Since the cell current, I_{cell} , is very small and the bitline capacitance is large, the voltage will drop very slowly at b , as shown in Figure 8.10b. The rate of change of the bitline can be approximated as follows:

$$I_{\text{cell}} = C_{\text{bit}} \frac{dV}{dt}$$

$$\therefore \frac{dV}{dt} = \frac{I_{\text{cell}}}{C_{\text{bit}}}$$

Clearly, I_{cell} controls the rate at which the bitline discharges. If we want a rapid full-swing discharge, we can make I_{cell} large. However, the transistors M_1 and M_3 would have to be larger. Since we have millions of such cells, the area and power of the memory would be correspondingly larger. Instead, we choose a different approach. We attach a sense amplifier to the bitlines to detect the small difference, ΔV , between b and $\bar{b}$ and produce a full-swing logic high or low value at the output. The trigger point relative to the rising edge of the wordline, $\Delta\tau$, for the enabling of the sense amplifier can be chosen by the designer based on the response characteristics of the amplifier. If the voltage swing ΔV and a target delay $\Delta\tau$ are specified according to Figure 8.10b, then

$$I_{\text{cell}} = \frac{C_{\text{bit}} \Delta V}{\Delta\tau}$$

This leads to the cell current value which, in turn, determines the final transistor sizes for M_1 and M_3 . Alternatively, if the transistor sizes are determined to optimize the cell area, then the corresponding delay is computed as

$$\Delta\tau = \frac{C_{\text{bit}} \Delta V}{I_{\text{cell}}}$$

We now establish a rule of thumb for transistor sizes for the read cycle using an example.

Read Cycle Design Guidelines

Example 8.2

Problem:

Compute the widths of M_1 and M_3 in Figure 8.10 given that the circuit can only tolerate a rise in voltage of 0.1 V at node q during the read operation. Assume that $C_{\text{bit}} = 2$ pF and that the specification calls for a 200 mV transition of the bitline in 2 ns. Use 0.13 μm technology parameters.

Solution:

When the wordline, wl , goes high, M_3 is a saturated enhancement load for the M_1 driver. The driver transistor is expected to be in the linear region of operation. Therefore, we can write the following equation:

$$\frac{W_1}{L_1} \frac{\mu_n C_{\text{ox}}}{\left(1 + \frac{V_q}{E_{\text{CN}} L_1}\right)} \left[(V_{\text{DD}} - V_{\text{T1}}) V_q - \frac{V_q^2}{2} \right] = \frac{W_3 v_{\text{sat}} C_{\text{ox}} (V_{\text{DD}} - V_q - V_{\text{T3}})^2}{(V_{\text{DD}} - V_q - V_{\text{T3}}) + E_{\text{CN}} L_3}$$

We first eliminate C_{ox} from both sides of the equation. Now, setting $V_q = 0.1$ V and ignoring body effect, we obtain

$$\begin{aligned} \frac{W_1}{0.1 \mu\text{m}} \frac{\left(270 \frac{\text{cm}^2}{\text{V} \cdot \text{sec}}\right)}{\left(1 + \frac{0.1}{0.6}\right)} \left[(1.2 - 0.4)0.1 - \frac{0.1^2}{2} \right] \\ = \frac{W_3(8 \times 10^6 \text{ cm/s})(1.2 - 0.1 - 0.4)^2}{(1.2 - 0.1 - 0.4) + 0.6} \\ \therefore \frac{W_1}{W_3} \approx 1.7 \end{aligned}$$

This ratio would be smaller if body effect were taken into account. The actual values of the widths depend on the desired rate of change of the bitline voltage, the delay specification, and cell current. If we require a bitline transition of 200 mV in 2 ns, with a total bitline capacitance of 2 pF, then the cell current is

$$I_{\text{cell}} = C_{\text{bit}} \times \frac{\Delta V}{\Delta \tau} = 2 \text{ pF} \times \frac{200 \text{ mV}}{2 \text{ ns}} = 200 \mu\text{A}$$

This is the average cell current through M_1 and M_3 . As a rough estimate, we could simply use the current through the access transistor when it turns on:

$$\begin{aligned} I_{\text{cell}} \approx \frac{W_3(8 \times 10^6)(1.6 \mu\text{F}/\text{cm}^2)(1.2 - 0.1 - 0.4)^2}{(1.2 - 0.1 - 0.4) + 0.6} = 200 \mu\text{A} \\ \therefore W_3 = 0.4 \mu\text{m} \end{aligned}$$

This implies that $W_1 = 0.7 \mu\text{m}$. These two sizes are larger than we would desire if we were trying to create a 1 Mbit SRAM. However, this example is intended to show the steps in the design process.

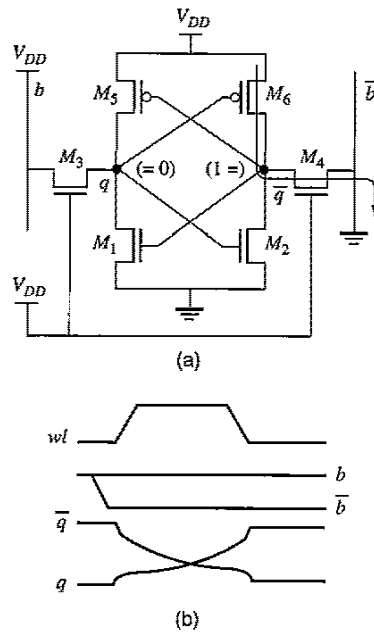
In practice, the device sizes are controlled by the RAM cell area constraints. As a rule of thumb, we typically use

$$\frac{W_1}{W_3} \approx 1.5 \quad (8.3)$$

and then optimize the sizes to provide the proper noise margin characteristics.

8.3.3 Write Operation

The operation of writing 0 or 1 is accomplished by forcing one bitline, either b or $\bar{b}$, low while the other bitline remains at about V_{DD} . In Figure 8.11, to write 1, $\bar{b}$ is forced low, and to write 0, b is forced low. The conditions when writing 1 are illustrated in Figure 8.11a. The cell must be designed such that the conductance of M_4 is several times larger than M_6 so that the drain of M_2 is pulled below V_S . This initiates a regenerative effect between the two inverters. Eventually, M_1 turns off and

**Figure 8.11**

Write operation and waveforms for 6T SRAM.

its drain voltage rises to V_{DD} due to the pull-up action of M_5 and M_3 . At the same time, M_2 turns on and assists M_4 in pulling output $\bar{q}$ to its intended low value. When the cell finally flips to the new state, the row line can be returned to its low standby level.

The design of the SRAM cell for a proper write operation involves the transistor pair M_6 - M_4 . As shown in Figure 8.11a, when the cell is first turned on for the write operation, they form a pseudo-NMOS inverter. Current flows through the two devices and lowers the voltage at node $\bar{q}$ from its starting value of V_{DD} . The design of device sizes is based on pulling node $\bar{q}$ below V_S to force the cell to switch via the regenerative action. This switching process is shown in Figure 8.11b. Note that the bitline $\bar{b}$ is pulled low *before* the wordline goes up. This is to reduce the overall delay since the bitline will take some time to discharge due to its high capacitance.

The pull-up to pull-down ratio for the pseudo-NMOS inverter can be determined by writing the current equation for the two devices and setting the output to V_S . To be conservative, a value much lower than V_S should be used to ensure proper operation in the presence of noise and process variations. Based on this analysis, a rule of thumb is established for M_6 - M_4 sizing:

$$\frac{W_4}{W_6} \approx 1.5 \quad (8.4)$$

The two ratios provided in Equations (8.3) and (8.4) are only estimates. One should remember that the actual values will depend on a number of factors such as area, speed, and power considerations. However, these two rules of thumb can be used to validate the solution, once obtained.

Exercise 8.1

Compute the ratio of M_6 to M_4 for the circuit in Figure 8.11 assuming that the node $\bar{q}$ is to be pulled down to V_{Th} which is well below the switching threshold.

8.3.4 SRAM Cell Layout

Once the cell transistors are designed, the next step is to construct a cell layout. A typical layout for a CMOS static cell is shown in Figure 8.12. Every effort is made to minimize the area of the memory cell. Extensive use is made of symmetry to allow the core array to be generated by simply "tiling" the cells together vertically and horizontally. Two levels of metal and one layer of poly are used to realize this memory cell. If we examine the layout carefully, we should be able to identify a total of six transistors.

Starting at the top, we have the two p -channel devices, M_5 and M_6 , laid out horizontally. In the middle of the cell, we have M_3 and M_4 , which are the two pull-down transistors laid out horizontally. Finally, near the bottom of the cell are the two access transistors, M_1 and M_2 , laid out vertically. The bitlines, b and $\bar{b}$, are routed in Metal2 vertically, while the wordline is horizontally routed in both poly and Metal1

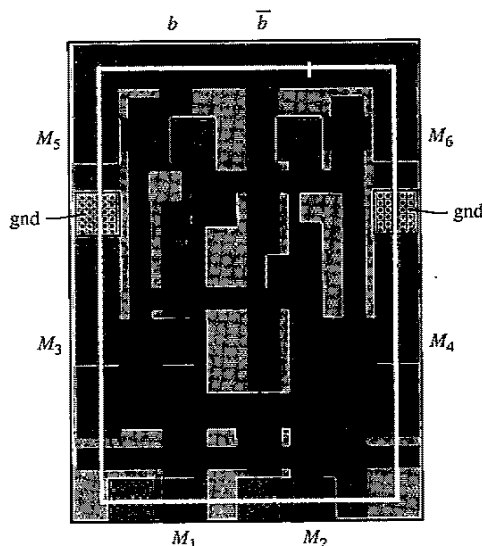


Figure 8.12
SRAM cell layout.

near the bottom of the cell. V_{DD} is routed in Metal1 at the top of the cell while Gnd is routed in Metal1 near the middle of the cell. The source/drain contacts are shared between pairs of neighboring cells by mirroring the cell vertically. The capacitance of the contacts per cell is therefore half the actual value due to sharing. The cell indicated by the center bounding box is replicated to create the core array. This cell is approximately 40λ by 30λ . Note that the substrate and well contacts are contained inside the cell. Removal of substrate and well plugs from the cell would result in a smaller cell.

The large number of devices connected to the wordline and bitlines gives rise to large capacitance (and resistance) values as described earlier. The row lines are routed in both Metal1 and poly to reduce resistance, while the bitlines are routed in Metal2. Calculations of total capacitance may be carried out using Equations (8.1) and (8.2).

Capacitance Calculations for the Wordline and Bitlines

Example 8.3

Problem:

What is the capacitance of the wordline and the bitlines for a 64K SRAM that uses the cell layout of Figure 8.12 with access transistors that are $0.5\ \mu\text{m}/0.1\ \mu\text{m}$ in size? The contacts on the bitlines are shared between pairs of cells and have a capacitance of $0.5\ \text{fF}$ each. Wire capacitance is $0.2\ \text{fF}/\mu\text{m}$. Assume $0.13\ \mu\text{m}$ technology parameters. The cell layout is 40λ by 30λ . Note that $1\ \mu\text{m} = 20\lambda$.

Solution:

If we were to design a 64K SRAM, it would contain a core area of 256×256 . Ignoring the resistance for the moment, the row capacitance would be due to the gate capacitance of 512 access transistors and the wire capacitance of 256 cells:

$$\begin{aligned} C_{\text{word}} &= 512 \times 2\ \text{fF}/\mu\text{m} \times 0.5\ \mu\text{m} + 256 \times 30\lambda \times 0.2\ \text{fF}/\mu\text{m} \times 1\ \mu\text{m}/20\lambda \\ &= 589\ \text{fF} \end{aligned}$$

The bitline capacitance per cell due to the source/drain capacitance of the access transistors is lower than usual since the voltage drop across the junction is close to V_{DD} . In addition, there is wire capacitance and a half contact capacitance per cell. The total is

$$\begin{aligned} C_{\text{bit}} &= 256 \times 0.5\ \text{fF}/\mu\text{m} \times 0.5\ \mu\text{m} + 256 \times 40\lambda \times 0.2\ \text{fF}/\mu\text{m} \times 0.1\ \mu\text{m}/2\lambda \\ &\quad + 128 \times 0.5\ \text{fF} = 230\ \text{fF} \end{aligned}$$

8.4 SRAM Column I/O Circuitry

In this section, we examine the column input/output (I/O) circuitry, which includes bitline precharge circuits, column multiplexers, write circuits, and read circuits. The column I/O must be designed with due consideration of the cell design and the timing specifications of the memory.

8.4.1 Column Pull-Ups

In both read and write operations, the bitlines are initially pulled up to a high voltage near V_{DD} . The circuits used to precharge the bitlines depend on the type of sensing that is used in the read operation. Figure 8.13 illustrates three possible precharge configurations. In Figure 8.13a, the precharge is similar to the dynamic logic precharge described earlier in Chapter 7. A precharge signal, PC, is applied to the two pull-ups and to a third transistor, called the *balance* transistor, connected between the two bitlines to equalize their voltage levels. When the wordline (*wl*) signal goes high, one bitline remains high and the other falls at a linear rate until *wl* goes low. The difference between the bitlines is fed into a voltage-sensing latch-based amplifier that is triggered when the differential voltage exceeds a certain threshold.

The precharge circuit of Figure 8.13b is reminiscent of the pseudo-NMOS circuits. Two static loads and a balance transistor form the precharge circuit. When PC is applied to the balance transistor, it simply equalizes the two voltage levels. Once the bitlines are precharged, the PC signal is turned off (raised to V_{DD}) and, at this point, the wordline can be activated. Of course, the pull-ups are still *on* so current will flow through one of them and into the cell side with the stored "0." Eventually, a steady-state output level will be reached by the bitline, as shown in the figure. This type of pull-up is suitable for current-sensing amplifiers since there is continuous

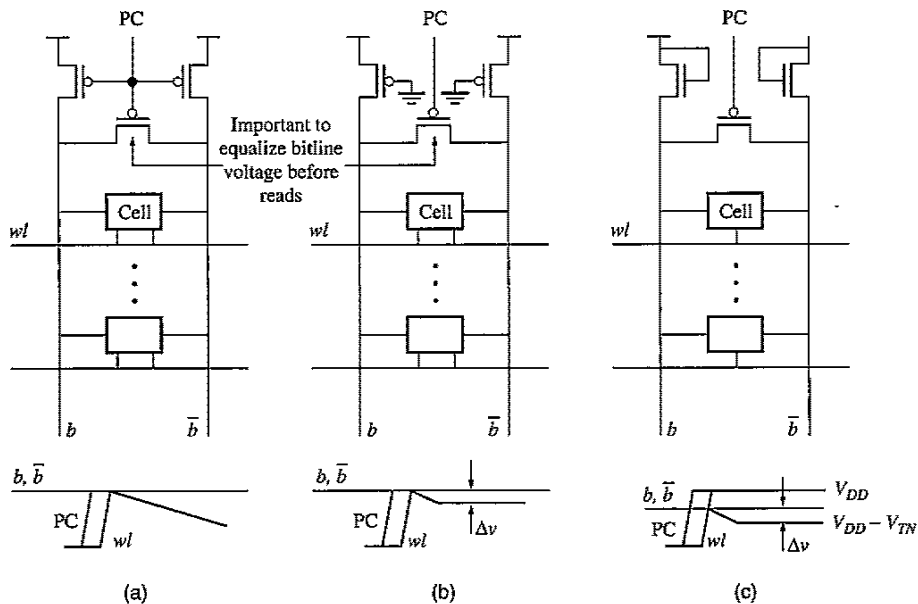


Figure 8.13
Column pull-up configurations.

current flow, or latch-based voltage-sensing amplifiers since the bitlines will establish a differential voltage, Δv .

Figure 8.13c is based on the NMOS saturated enhancement load. Therefore, the maximum possible voltage on the bitline is $V_{DD} - V_T$. When PC is applied to the balance transistor, it equalizes the two voltage levels. Once the lines are precharged high, the PC signal is turned off (raised to V_{DD}) and then wl goes high. At this point, the pull-ups are still active so current will flow through one of them into the cell side with the stored "0." Again, a steady-state output level will be reached by the corresponding bitline, as shown in the figure, although this value will be lower than the pseudo-NMOS case. This type of pull-up is suitable for differential voltage sensing amplifiers since the bitline voltages initially start at $V_{DD} - V_{TN}$. This lower voltage is needed for a proper biasing and output swing of the differential amplifier, as will be described later.

The PC signal may be generated in a variety of ways, but typically it is produced by an address transition detection (ATD) circuit. One form of this circuit is shown in Figure 8.14. The ATD signal is triggered by any transition on the address inputs. The basic circuit is comprised of a set of XOR gates, each with a delay element on one of the inputs. When an address line changes, it causes the XOR gate to generate a short pulse since the inputs differ in value for a short time. Circuits that generate a short pulse of this nature are called *one-shots*, which are part of the *monostable* family of circuits. The duration of the pulse, τ_D , is determined by the delay element. The delay line may be constructed from a simple inverter chain with an even number of inverters. In the figure, N is an even number and τ_{pinv} is the inverter propagation delay.

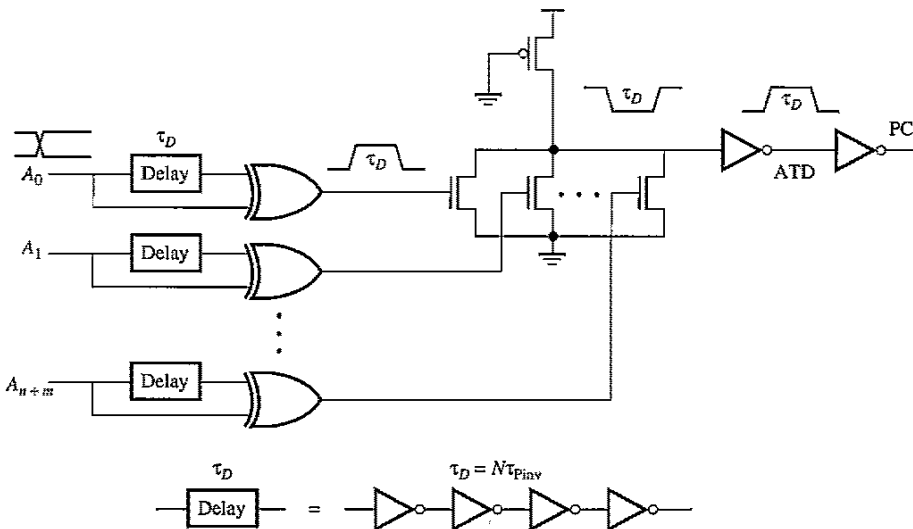


Figure 8.14
Address transition detection (ATD) circuit.

Once the pulse is generated, it turns on one of the pull-down transistors of the pseudo-NMOS NOR gate. A negative going pulse is generated at its output. This is passed through another inverter to generate the actual ATD signal. Many of the timing signals in SRAMs are derived from this basic circuit so it is required to drive a very high capacitance. Therefore, the signal should be properly buffered using logical effort or any other optimization method. Once generated, it can be inverted and applied to the bitline precharge elements as the PC signal. The address transitions usually take place before the beginning of a clock cycle and, as a result, the precharge operation typically occurs at the end of a previous memory cycle.

8.4.2 Column Selection

Once all the columns have been pulled up to a high voltage, the next step is to select the column(s) that will be involved in the read or write operation. This column selection is performed using a decoder/multiplexer combination. The m -bit column address is used to select one or more of the 2^m columns. This can be performed using a decoder, similar to the row decoder, driving a number of CMOS pass transistors as shown in Figure 8.15. The pass transistors require complementary signals. Of course, if an 8-bit output is desired, then the decoder outputs would each drive eight CMOS transmission gates, and fewer column address bits would be needed.

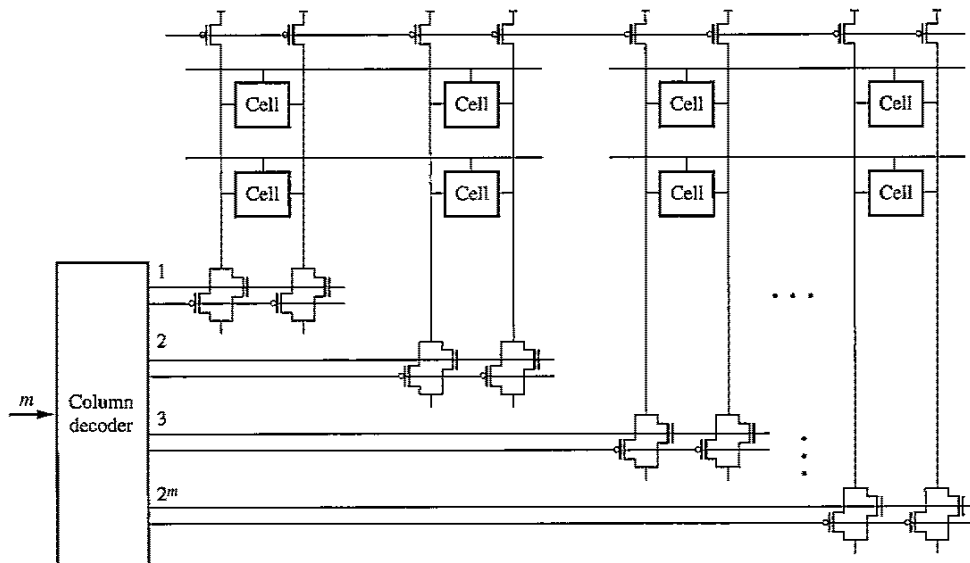


Figure 8.15
Column decoding and multiplexing.

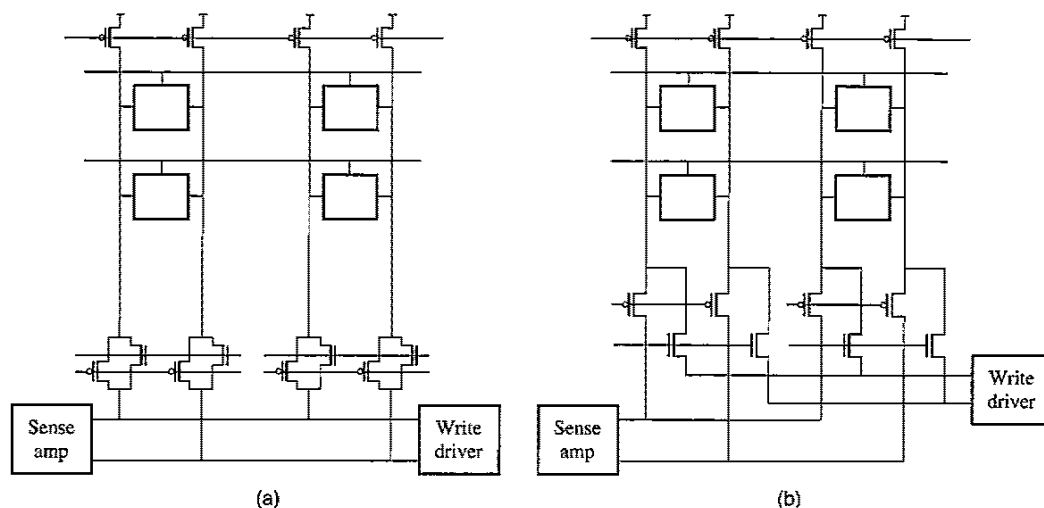


Figure 8.16
Column selection.

The optimal design of the column decoder proceeds in the same way as described earlier for the row decoder. The transmission gates driven by the decoder are also sized for optimal speed. They are connected to the sense amplifier for read operations and the write driver for write operations. This is shown in Figure 8.16a. Note that the use of the CMOS transmission gate presents a routing problem since each of the signals driving the pass transistors must be complementary (we are driving both PMOS and NMOS devices).

The routing can be simplified by realizing that the PMOS device is better at transmitting high signals while the NMOS device is better at transmitting low signals. Since the bitlines are near V_{DD} during a read, we should turn on the PMOS device during a read operation and leave the NMOS device off. During a write, one bitline is pulled to a low voltage. Therefore, we leave the PMOS device off and only turn on the NMOS device. It is possible to separate the NMOS and PMOS devices and only turn them on when needed. This is shown in Figure 8.16b. The NMOS devices are only connected to the write drivers while the PMOS devices are only connected to the sense amplifiers since they would be turned on during a read operation.

Now that the lines have been separated, there is one other improvement we can consider. Rather than a single level of multiplexing, it is possible to reduce the overall transistor count by using a tree decoding structure as shown in Figure 8.17. In this example, we have a 4-bit column address which would normally translate to 16 enable lines. Instead, we use two 2-to-4 decoders that select 1-out-of-4 pass transistors at each level. As we add more levels in the tree, the signal path is slower but the decoder size is reduced. For this example, we have shown two-level tree decoding

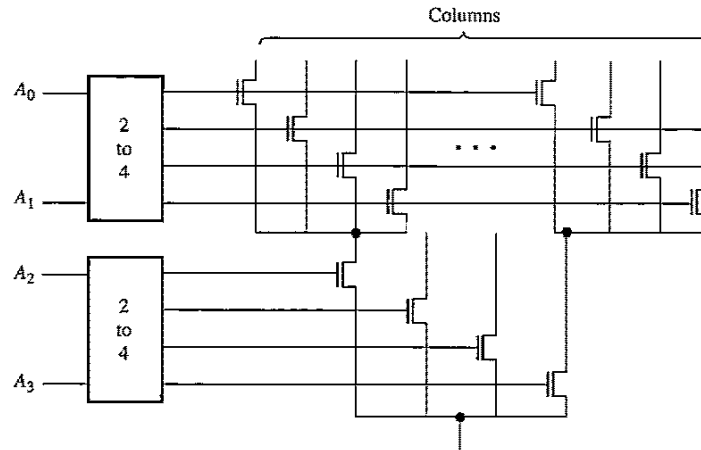


Figure 8.17
Two-level tree decoder for a 4-bit column address.

with exclusively NMOS pass transistors. We would need a corresponding set of PMOS transistors in a similar tree configuration. The tree decoding strategy requires less power but may be slower as more and more levels are added. These circuits should be designed based on the timing and power specifications for the memory.

8.4.3 Write Circuitry

Simplified write circuitry for the SRAM is shown in Figure 8.18. The operation is as follows. First, the columns are precharged to V_{DD} using M_7 and M_8 . Next, the address and data signals are set up and held stable for a required amount of time before the clock is applied. The address signals are converted into column select and wordline activation signals. Before the wordline is enabled, the data and write signals are applied to pull one column to ground while leaving the other side at V_{DD} . This is done by ANDing the input data with the write signal. The pull-down transistors, M_{13} , M_{14} , and M_{15} in this case, are sized to discharge the column line in a specified amount of time. When the wordline goes high, current flows out of the cell and flips the sense of the cell. As described earlier, the internal cell voltage must be pulled below the switching threshold to enable the cell to flip state. Once it has switched, the wordline and column select lines may be returned to their standby values.

8.4.4 Read Circuitry

The read circuitry is activated when the wordline goes high. The cell transistors draw current from the highly capacitive column during a read operation. Therefore,

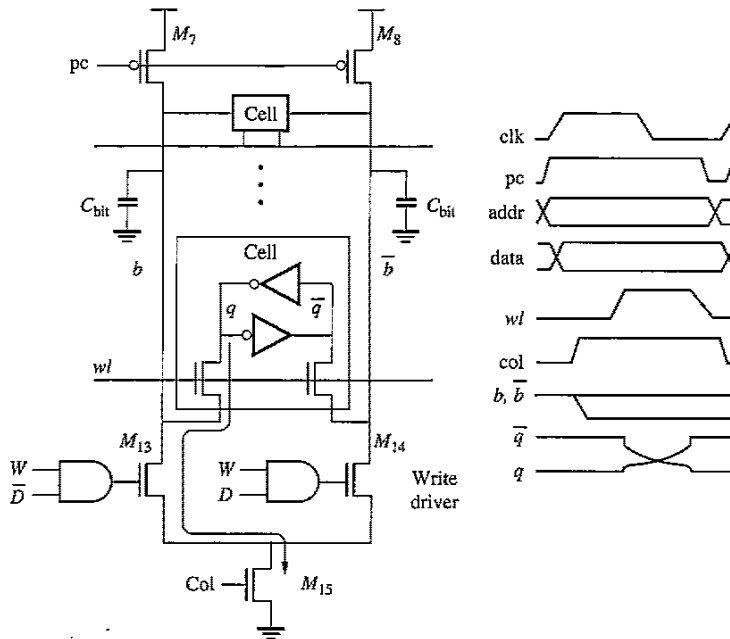


Figure 8.18
Write driver circuit.

the bitlines slowly drop in voltage and could potentially cause long access times. To reduce read access time, the memory is designed so that only a small voltage change on one column line or the other is needed to detect the stored value. Two or more amplifying stages are used to generate a valid logic output when the voltage difference between b and $\bar{b}$ is about 150–200 mV. Thus the column delay is only due to the time needed to achieve this small voltage change.

Figure 8.19 shows a simplified version of the read circuitry for a CMOS static memory. One of the precharge circuits of Figure 8.13 is used to pull the column lines high. In this case, the columns are biased at V_{DD} by transistors M_7 , M_8 . Then, the address, data (not used during read), and clock signals are applied. Again, the address signals translate into column enable and wordline activation signals. Usually the column selection and sense enable are activated at the same time. The sense amplifier depicted in Figure 8.19 is used to provide valid high and low outputs using the small voltage difference between inputs b and $\bar{b}$. The precharge circuit must be consistent with the sense amplifier circuit. Otherwise, the sense amplifier may not operate properly.

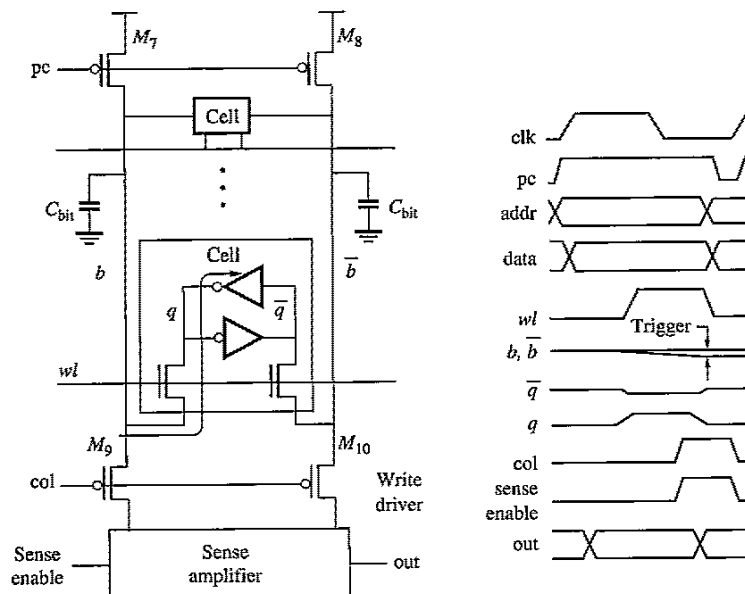
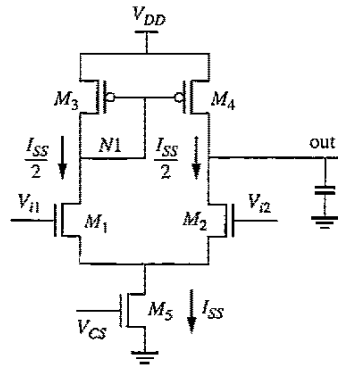


Figure 8.19
Basic read circuitry.

Figure 8.20 provides an example of a sense amplifier in the form of a conventional CMOS differential amplifier.⁴ Typically, SRAMs may use eight identical sense amplifiers to provide simultaneous output of eight data bits. When using these types of sense amplifiers, the column pull-up transistors must be saturated enhancement loads (shown at the top of Figure 8.13c). Otherwise, the inputs would start at V_{DD} which would make it difficult to bias M_4 and M_5 properly for the desired output swing at node *out*.

The main reason for using this type of sense amplifier is to improve the noise immunity and speed of the read circuit. Since the voltage swing on the bitlines is limited due to the large capacitances, any noise on these lines may cause an error in the reading process. In fact, any noise that is common to *b* and $\bar{b}$ should not be amplified. We are only interested in the differential signal changes between the two bitlines. The sense amplifier shown in Figure 8.20 attenuates *common-mode noise* and amplifies *differential-mode signals*. The detailed operation of this circuit involves the concepts of analog circuit design. However, since we are using it for

⁴ These are called *sense amplifiers* when used in memory applications; their role is to sense which bitline is dropping in voltage. Normally, such a circuit is used for small-signal voltage gain rather than large-signal sensing applications.

**Figure 8.20**

Differential voltage sense amplifier.

large-signal applications, we can study its properties from a more digital point of view.

The circuit can be divided into three components: the current mirror, common-source amplifier, and the biasing current source. All transistors are initially placed in the saturation region of operation so that the gain is large. They also use large values of channel length, L , to improve linearity.

The two transistors, M_3 and M_4 , act to provide the same current to the two branches of the circuit. That is, the current flowing through M_3 is mirrored in M_4 :

$$I_3 \approx I_4$$

Any difference in the two currents is due to differences in their V_{DS} values. The transistor M_5 sets the bias current, I_{SS} , which depends on the bias voltage V_{CS} . At steady-state, the current flowing through the two branches of the amplifier should be equal to $I_{SS}/2$.

The two input transistors, M_1 and M_2 , form a source-coupled differential pair. The two input voltages, V_{i1} and V_{i2} , are connected to the column lines. The biasing of the circuit must be done carefully to allow the output node to have a large enough swing. Specifically, the transistors must be *on* and in the saturation region of operation for high gain. In order to accomplish this, the inputs to M_1 and M_2 must be set to approximately $V_{DD} - V_{TN}$ rather than V_{DD} . To understand this, consider the case when the inputs are precharged to V_{DD} . To keep the input devices in saturation, their two drain nodes, $N1$ and *out*, would be biased at the saturation voltage:

$$V_{N1} = V_{out} = \frac{(V_{DD} - V_{TN})E_{CN}L_N}{(V_{DD} - V_{TN}) + E_{CN}L_N} \approx (V_{DD} - V_{TN})$$

The above simplification is possible since the channel lengths are large.

A problem arises if the two nodes, $N1$ and out , are biased at this value: both p -channel devices would be at the edge of cutoff. In practice, the PMOS threshold voltage is higher in magnitude than the NMOS threshold voltage. Therefore, both M_3 and M_4 would be completely off in the steady-state condition. Instead, if we biased the inputs of M_1 and M_2 at $V_{DD} - V_{TN}$, then

$$V_{N1} = V_{out} \approx (V_{DD} - V_{TN}) - V_{TN} \approx V_{DD} - 2V_{TN}$$

Now there is enough *headroom* for the two PMOS devices to be comfortably *on* and in saturation. This input bias condition requires the use of the column pull-up circuits of Figure 8.13c.

With the biasing established, the sense amplifier operates as follows. Initially, the bias currents in the two branches are equal and the two inputs are at $V_{DD} - V_{TN}$. When the voltage at one input decreases, it decreases the current in that branch. At the same time, the current in the other branch increases in value to maintain a total of I_{SS} through M_5 .

We examine the discharging and charging cases in Figure 8.21. Assume that M_1 is the input that has dropped below $V_{DD} - V_{TN}$ by the prescribed amount to turn it off, as in Figure 8.21a. This implies that the currents in M_3 and M_4 are both zero. However, since the current in M_5 is I_{SS} , it follows that this current must be discharging the output capacitance through M_2 . Therefore, the output voltage is quickly forced to ground. In the other scenario depicted in Figure 8.21b, if the input V_{i2} drops by the prescribed amount, M_2 is turned off. Then all the current flows through M_1 , M_3 , and M_5 . The current of M_3 is mirrored in M_4 . Since the current in M_2 is zero, this current must flow to the output to charge it to V_{DD} .

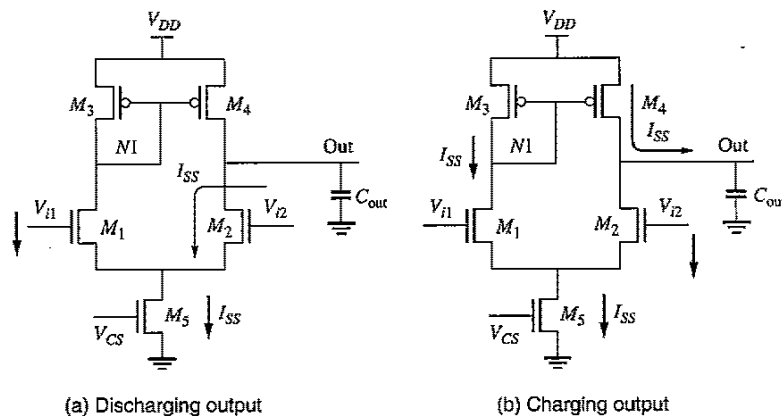


Figure 8.21

Detecting "0" and "1" using a differential sense amplifier.

This type of differential amplifier is used in high-speed applications. The required differential voltage for proper operation is of the order of 100 mV to 200 mV. Its speed can be adjusted depending on how much power dissipation can be tolerated. Consider modeling the output transition as a current source driving a capacitor. The rate of change of the output, whether switching high or low, is given by

$$\frac{dV}{dt} = \frac{I_{SS}}{C_{out}}$$

This is called the *slew rate* (i.e., dV/dt at the output). Rearranging, the delay through the sense amplifier is

$$\Delta\tau = \frac{C_{out} \Delta V_{out}}{I_{SS}}$$

To reduce the delay, a large I_{SS} can be selected. However, the power dissipation in steady-state is given by

$$P = I_{SS}V_{DD}$$

Therefore, a tradeoff exists between the speed and the power dissipation; both are controlled by the choice of I_{SS} . Once a suitable value of I_{SS} is selected, the W/L of the devices can be determined. For the input devices, the W/L determines the $V_{GS} - V_{TN}$ value. This value is the gate overdrive term that establishes the desired bitline swing value. As the input transistor sizes increase, the gate overdrive term decreases. Since we require a small gate overdrive, the input devices are required to be rather large. The sizes of the other transistors are based on the bias voltages needed at the internal nodes. The complete design of such amplifier circuits falls into the realm of analog circuit design. Further details can be obtained by consulting the references at the end of the chapter.

A second option for the sense amplifier is the latch-based circuit shown in Figure 8.22. The circuit is effectively a cross-coupled pair of inverters with an enabling transistor, M_1 . This circuit relies on the (slower) regenerative effect of inverters to generate a valid high or low voltage. It is a lower power option since the circuit is not activated until the required potential difference has developed across the bitlines. However, it is slower since it requires a large input voltage difference and is not as reliable in the presence of noise as the previous sense amplifier.

The initial sequence of operations is similar to the differential sense amplifier described above. The bitlines are precharged to V_{DD} with either Figure 8.13a or Figure 8.13b. Then the wordline is enabled and one of the bitlines drops in voltage. As the bitline differential voltage reaches the prescribed amount, the sense enable is activated. The timing of the sense enable is critical as described later. For now, assume that it arrives at the proper time. At this point, the bitline difference is fed into the cross-coupled inverters. One side drops in voltage faster than the other side, since one side will always have more gate overdrive than the other. When the voltage drops below V_{TN} on one side, it turns off the pull-down transistor of the opposite side, and

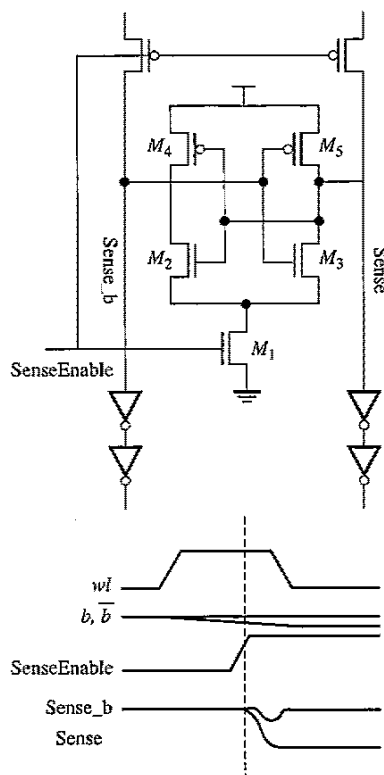
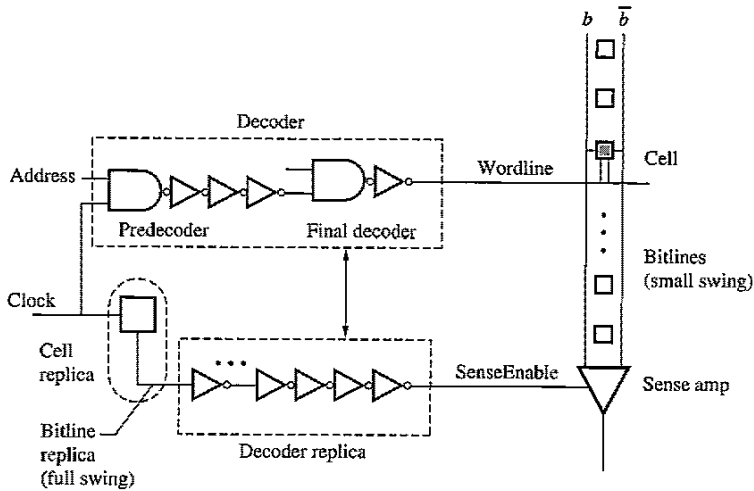


Figure 8.22
Latch-based sense amplifier.

the pull-up transistor acts to raise the voltage to V_{DD} . This regenerative process is shown in the timing diagram of Figure 8.22. The device sizing follows previously discussed methods for flip-flops.

The more important issue is the timing of the $SenseEnable$ signal. If the latch is enabled too early, the latch may not flip in the proper direction due to noise. If it is enabled too late, then it will add unnecessary delay to the access time. In addition, process variations will control the actual timing of the signal. In order to guarantee that the signal will arrive at the proper time in the presence of process variations, one needs to introduce a replica circuit that mimics the delay of the actual signal path, shown in Figure 8.23. Here, the upper path emanating from the clock is the actual signal path to the bitlines. The $SenseEnable$ should arrive as soon as the bit-line swing reaches the desired value. By creating a second path (the lower path) that exhibits the same delay characteristics, we can ensure that the $SenseEnable$ arrives at the correct time.

**Figure 8.23**

Replica circuit for sense amplifier clock enable.

The critical path for the read cycle starts from the clock and the address inputs and terminates at the sense amplifier inputs. The signal flows through the decoder and generates the wordline, which activates the memory cell that drives the bitlines. The swing on the bitlines is presented to the sense amplifier. This is the point at which we wish to enable the sense amplifier. The purpose of a replica circuit is to duplicate the delays along this path with circuits that correspond to each delay element. Essentially, we want to have a decoder replica that tracks the gate delays in the real decoder, and a cell replica that tracks the bitline discharge delay of the actual bit cell.

Note that we have placed the memory cell replica *before* the decoder in Figure 8.23. It is not appropriate to place the memory cell after the decoders in the replica path since it would have to drive all the sense amplifiers at the bottom of the memory. Since a small memory cell does not have the needed drive capability, we place the replica cell ahead of the decoder. We can keep the memory small and still deliver a full-swing signal as needed by the input to the decoder replica. The buffers of the decoder replica can be used to drive the sense amplifiers.

One issue for the replica memory cell in this configuration is that we require the full-swing output to have the same delay as the small swing on the actual bitlines. For example, if the actual cell requires 500 ps to transition by 180 mV, then the replica memory cell would require approximately 5 ns to transition by 1.8 V. This is not an acceptable delay in the replica path.

The replica cell should, in fact, be a replica column line with only enough cells to match the timing of the actual column. This is shown in Figure 8.24. For example,

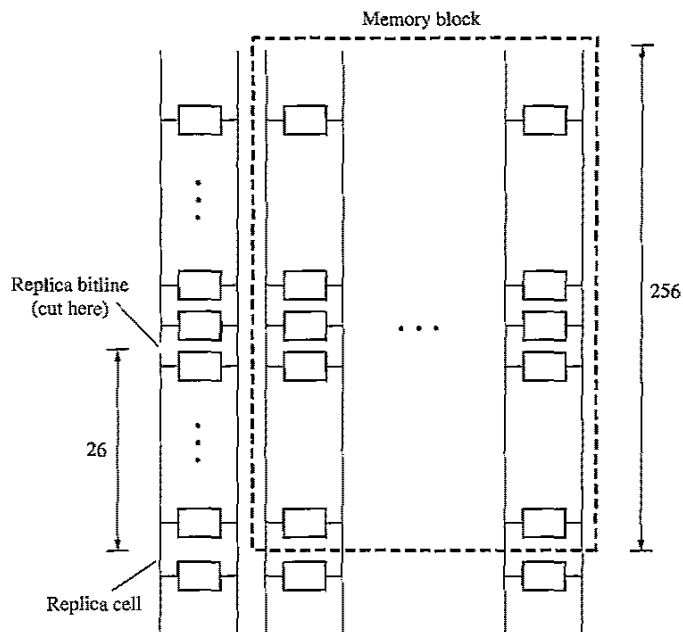


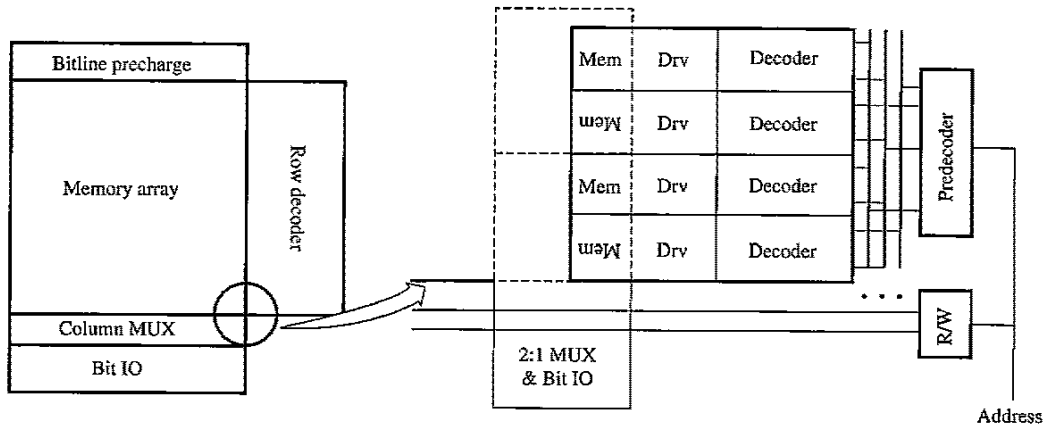
Figure 8.24
Replica cell design.

if we have 256 bits in the true bitline with a swing of 180 mV, then we only require roughly 26 cells in the replica circuit to produce a full swing of 1.8 V in the same time interval. The slight round off error in the number of cells used is not an issue since we will ensure that the replica path is slightly longer than the actual path delay. With the full swing from the replica path cells, we can drive the decoder replica gates. The needed 26 bits can be cut from a section of additional columns that are always fabricated alongside the main memory array to avoid “edge effects” on each end.

To ensure that the SenseEnable does not arrive too early for any reason, we should add an extra gate delay or two to the decoder replica. Designed in this manner, the SenseEnable will arrive at the proper time in the presence of process and environmental variations.

8.5 Memory Architecture

The overall memory architecture can now be described. In Figure 8.25, we illustrate a high-level layout of the memory array. The core array containing the cells is the largest block. The bitline precharge circuits are positioned above the core. The row

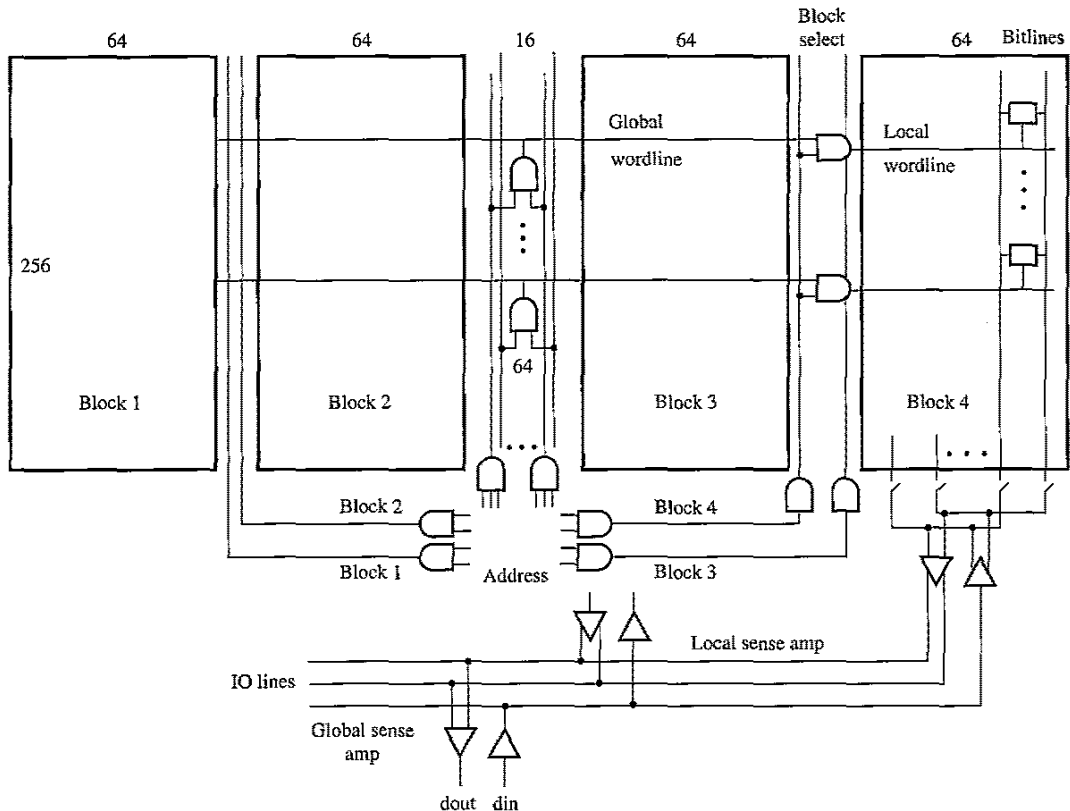
**Figure 8.25**

Basic memory architecture.

decoder is placed on the right side and the column multiplexer and bit I/O are located below the core array. If we zoom in on a corner region of the memory, as shown in the right-hand side of Figure 8.25, we see that the row decoder is comprised of a predecoder and a final decoder. The decoder drives the wordlines horizontally across the array. Each pair of bitlines feeds a 2:1 column decoder (in this case) which is connected to the bitline I/O circuits such as sense amplifier and write drivers. Each memory cell is mirrored vertically and horizontally to form the array, as indicated in the figure.

Several factors contribute to a limit on the maximum speed of operation. Delays in address buffers and decoders naturally increase as the number of inputs and outputs increase. Row lines are typically formed in polysilicon and may have substantial delays due to distributed RC parameters. A metal line may be placed in parallel and contacted to the poly line to reduce the delay (cf., Figure 8.12). Column lines are usually formed in metal, so resistance is not as significant, but the combined capacitance of the line and many parallel access transistors connected to them results in a large equivalent lumped capacitance on each of these lines. The large capacitances on the wordline and bitlines also contribute to excess power dissipation.

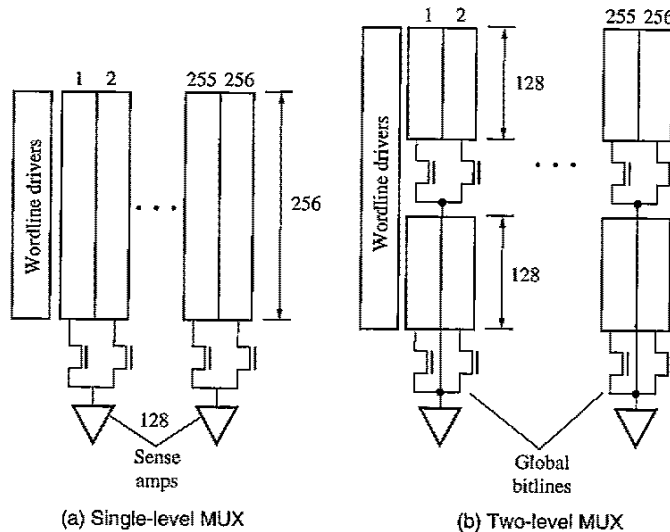
In order to reduce delay and power, a number of different partitioning approaches have been used. One technique is the divided wordline (DWL) strategy shown in Figure 8.26 for a 64K bit SRAM. Part of the 8 row address bits (6 in this case) are used to define global wordlines. A total of 64 global wordlines are created. These lines do not drive memory cells (i.e., the two access transistors within each cell) and therefore have far less capacitance than the regular wordlines. The remaining 2 bits of the address are used to generate local wordlines that actually drive the

**Figure 8.26**

Divided wordline strategy to reduce power and delay.

cell access transistors. In this example, four blocks are created and accessed using the local wordlines. The total cell capacitance is reduced by up to a factor of 4. Therefore, the power will be reduced greatly. In addition, the delay along the wordlines is also reduced.

A similar partitioning strategy can be applied to the bitlines, as shown in Figure 8.27. An architecture without partitioning is shown in Figure 8.27a. For this case, neighboring pairs of bitlines are multiplexed to produce 128 outputs (i.e., there are 128 sense amplifiers in this example). If the bitlines are partitioned into two sections, the bitline capacitance is reduced by a factor of 2. The proper cell must be selected using a two-level multiplexing scheme of Figure 8.27b. To achieve the same bitline swing as in Figure 8.27a would only require roughly half the time. Further partitioning can be carried out with a corresponding increase in the complexity of multiplexing.

**Figure 8.27**

Bitline partitioning to reduce delay.

8.6 Summary

This chapter has focused on the application of material in previous chapters to the design of an SRAM. Modern memories are, of course, much more complicated but most of them can be understood with the basic concepts that have been presented in this chapter. Since memory is very regular in nature, the design process has reached a point where it can be implemented in software. Today, there are CAD tools called *memory compilers* that can generate a memory design within minutes. While the sizes of the memories that can be generated are still limited to some degree, the automatic synthesis approach for memories will be used more frequently in the future. This concludes the discussion on SRAM circuits. Other types of memory circuits are described in Chapter 9.

REFERENCES

1. B. Prince, *Emerging Memories: Technologies and Trends*, Kluwer Academic Publishers, Boston, MA, 2002.
2. K. Itoh, *VLSI Memory Chip Design*, Springer-Verlag, Heidelberg, 2001.
3. J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Designer Perspective*, Second Edition, Prentice-Hall, Upper Saddle River, NJ, 2003.
4. S. M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits, Analysis and Design*, Third Edition, McGraw-Hill, New York, 2003.
5. H. Veendrick, *Deep-Submicron CMOS ICs*, Second Edition, Kluwer Academic Publishers, Boston, MA, 2000.

6. J. P. Uyemura, *CMOS Logic Circuit Design*, Kluwer Academic Publishers, Boston, MA, 1999.
7. Behrad Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw-Hill, New York, 2001.

PROBLEMS

- P8.1. What are the main differences between the ROM, SRAM, DRAM, EPROM, E²PROM, and Flash? Which is the most popular memory for embedded applications (i.e., on the same chip as the processor logic blocks)? Describe suitable applications for each one.
- P8.2. Draw the circuit equivalent for the 6T SRAM of Figure 8.12. Estimate the height and width of the cell. Assume that contacts are 4λ by 4λ .
- P8.3. Implement an 8-bit decoder using NAND2, NAND3, and NAND4 logic gates and inverters, following a two-level decoding scheme. You do not have to design the sizes of the transistors. Why is a two-level scheme preferred over a multilevel scheme? What is the *branching effort* of the decoder from input to output (see Chapter 6 for the definition of this term)?
- P8.4. Consider the SRAM cell of Figure 8.8 with a stored 0 on the left side and a stored 1 on the right side. Design the transistors of the SRAM such that node q does not exceed V_{TN} during a read operation and node $\bar{q}$ drops below V_S during a write operation. The desired cell current during a read operation is $300\ \mu\text{A}$. Use $0.18\ \mu\text{m}$ technology parameters.
- P8.5. Redesign the SRAM cell of the previous problem by assuming worst-case conditions for a read operation as follows: the threshold voltage of M_3 is reduced by 10%, the width M_3 is increased by 10%, the threshold voltage of M_2 is decreased by 10%, and the width of M_2 is increased by 10%. Explain why this is considered to be worst case for a read operation. Simulate the circuit in SPICE to demonstrate its operation under worst-case conditions.
- P8.6. Redesign the SRAM cell of the previous problem by assuming worst-case conditions for a write operation as follows: the threshold voltage of M_4 is increased by 10%, the width M_4 is decreased by 10%, the threshold voltage of M_1 is decreased by 10%, and the width of M_1 is increased by 10%. Explain why this is considered to be worst-case for the write operation. Simulate the circuit in SPICE to demonstrate its operation under worst-case conditions.
- P8.7. Consider the 6T SRAM cell of Figure 8.8. Replace M_3 and M_6 by poly resistors that are $100\ \text{M}\Omega$ in value. Explain how this new 4T cell works for read and write operations. How does the internal node get pulled to a high value? Is the new cell static or dynamic?

- P8.8.** For the sense amplifier shown in Figure 8.20, answer the following questions using 0.18 μm technology parameters:
- If the sense amplifier is driving a load of 50 fF in 500 ps, what is the required value of I_{SS} ?
 - In order to turn off the input transistors with a bitline swing of 100 mV, what values of W/L are needed?
 - Which of the three column pull-up configurations of Figure 8.13 would be used with this sense amplifier? What is the initial voltage at the inputs to the sense amp?
 - Given the size of the input transistors, what is the steady-state voltage at the gate node and the resulting size of M_3 ?
 - Choose the sizes of M_3 and M_4 to establish a suitable steady-state output voltage.

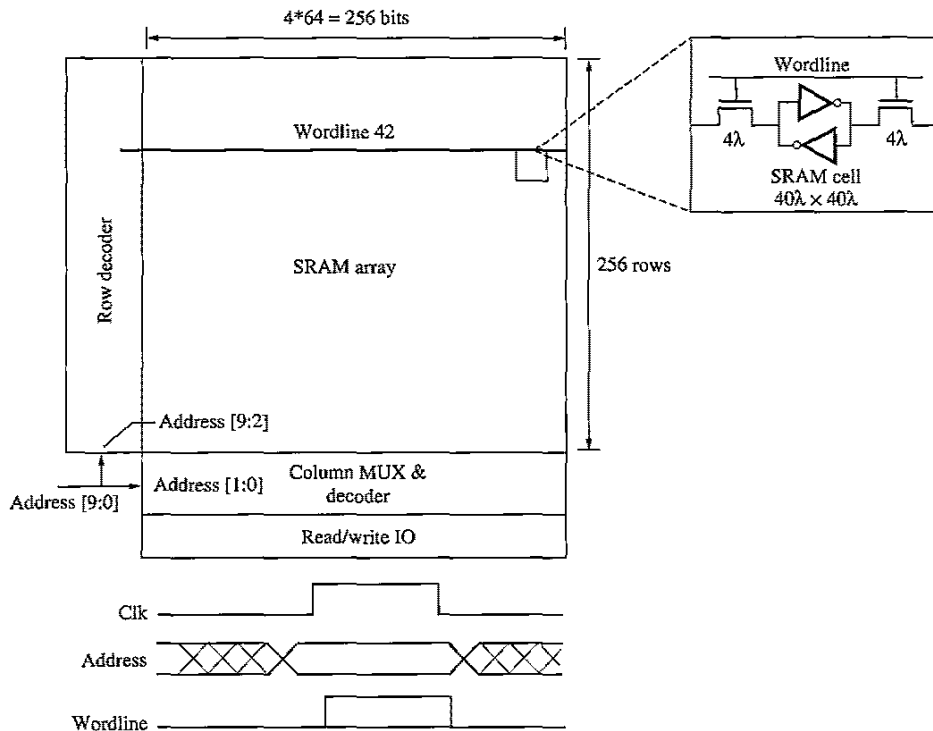
DESIGN PROBLEM 1: Decoder Design in 0.18 μm Technology

Your task is to create a decoder for a small memory shown in Figure D8.1. The memory consists of 1K words, each containing 64 bits, arranged as a 256×256 array of 6T SRAM cells. A word is selected for reading by the row and column decoders. The row decoder asserts one of the 256 wordlines. Each Metal2 wordline selects a row of four 64-bit words. Four selected words feed the input of the 4:1 MUX controlled by the outputs of the column decoder.

The SRAM receives a 10-bit address. The inputs to the row decoder are the upper 8 bits of the address, while the lowest 2 bits of the address feed the column decoder. Each access transistor in the memory cell is 4λ in width. Each SRAM cell is $40\lambda \times 40\lambda$. The diagram in Figure D8.1 illustrates the SRAM array, with a blow-up of one of the SRAM cells in the forty-second word.

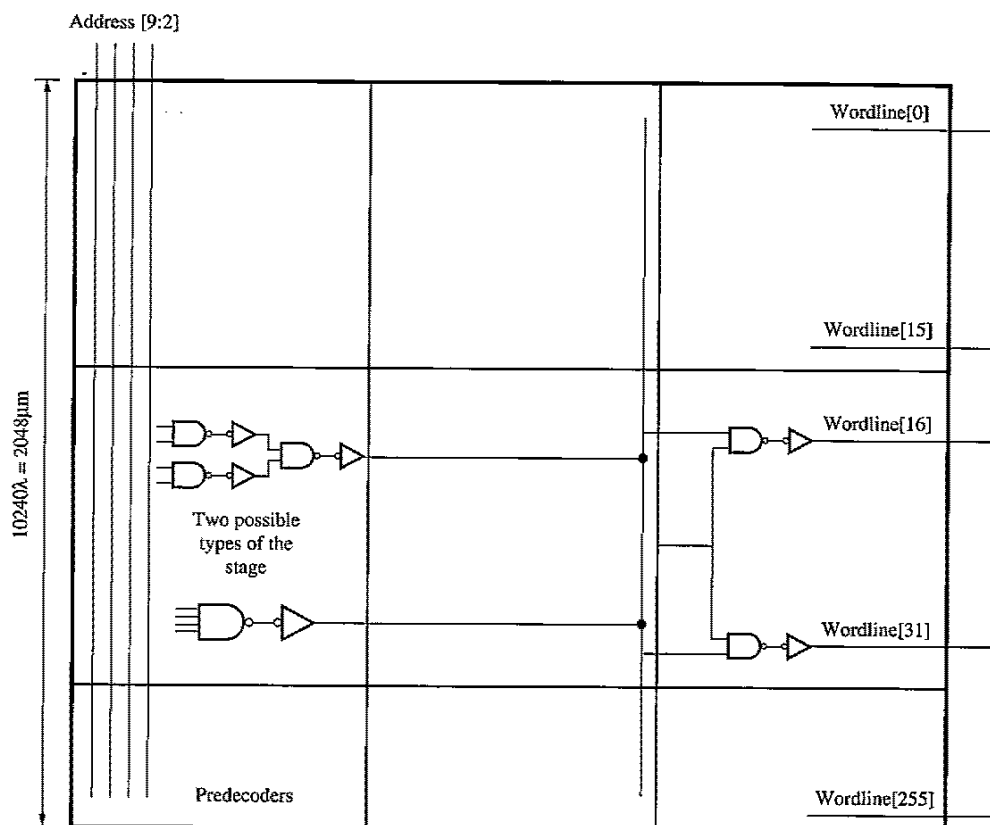
The spec allows a maximum of 5 fF of gate capacitance on any address input. You can assume that the clock period is 24FO4 delays. Your task is to minimize the delay from the address becoming stable to the wordline rising. The general topology for the decoder is provided in Figure D8.2. Your design goal is to implement an optimal row decoder that converts 8 address bits into 256 wordlines using logical effort.

- What is the total load on one wordline? Include both the loading of the SRAM cells and the capacitance of a Metal2 wordline. Assume a wire capacitance of 0.2 fF/ μm . You can assume that gate capacitance is 2 fF/ μm .
- Find the total path effort in the address decode path by estimating the fanout from input (address bit) to output (wordline), the branching factor, and the logical effort. Estimate the number of stages needed in the optimal design.

**Figure D8.1**

SRAM layout and timing information.

- (c) You can implement the predecoder in either 2 or 4 stages as shown in Figure D8.2. Which of the two approaches is better? Using logical effort, decide which architecture you are going to use. For both cases, the final decode stage is a NAND2-INV combination. Ignore any sload capacitances.
- (d) Next choose the correct design for the decoder, and size the gates to optimize performance. What is your estimate of the delay of the decoder in terms of FO4 delay (include the delay from the parasitics of the gates in your estimate)?
- (e) Now include the sload at the outputs of the predecoder stage. The sload is due to the wire running vertically in Figure D8.2. Compute the actual sizes when the sload is included.
- (f) Compare the hand-calculated delay against SPICE.

**Figure D8.2**

Decoder topology, branching, and wire loads.

DESIGN PROBLEM 2: Design of SRAM Cell and Read/Write Circuitry in 0.18 μm Technology

The problem explores cell and I/O design issues for a 64K SRAM based on the previous problem. Since there are 256 columns in the SRAM and 64 bits/word, it requires 64 sense amplifiers (1 per bit of output) and each one is driven by 1 of 4 possible columns (one column from each interleaved 64-bit word).

- (a) First consider the cell design. Recall that in Design Problem 1, the access transistor of the SRAM cell was given as 4λ . Size the rest of the transistors to satisfy the following requirements: (1) During a read, the internal voltage should not rise above V_{TN} , (2) During a write, the access transistor must be able to pull the internal node to at least 0.8 V (i.e., well below V_S

of the inverter), and (3) Make the transistors as small as possible. One assumption you could make is that the bitline is around 0 V during the write operation. For the read operation, assume that the bitline acts like a supply voltage at V_{DD} . Rules of thumb are available for the design but you may need to adjust these values to make the cell work properly. All transistors should be specified in integer units of λ . The minimum channel length/width is 2λ .

- (b) Estimate V_s of the inverters used in the cell. Why is this number important in the cell design? Use SPICE to verify the dc parameters you calculated. Provide both V_{OL} for the read operation and V_s as measured in HSPICE. Adjust any device sizes as necessary to achieve the design specifications.
- (c) Using hand calculations, estimate the current that the cell can draw from the bitline when the wordline goes high. Use an average between the initial value and the steady-state value to estimate this current.
- (d) Next, calculate the capacitance of the bitline. There are 256 rows of cells and, from Design Problem 1, each cell is $40\lambda \times 40\lambda$. The column pull-up transistors are each 50λ . Assume that each contact has a capacitance of 1 fF.
- (e) Next compute the time required to discharge one of the bitlines by 180 mV.
- (f) The last part of the read cycle involves the sense amplifier. The design of the sense amp has been provided in P8.8 using Figure 8.20. Compute the discharge time assuming that the bias current is $300 \mu\text{A}$ and the output capacitance is 50 fF. Assume a voltage swing at the output of the sense amp of 0.9 V.
- (g) The next step of the design is the write circuit of Figure 8.18. To write the cell, the bitline has to be driven low enough so that the cell switches. Ideally, the write time should be about the same speed as the read time. If we work to make it faster, we are wasting our effort because the read will be the limiting factor. If it is slower than the read, then the write is in the critical path and that is not desirable either. Using a simple RC model, determine the effective resistance needed to get the bitlines to swing down to $V_{DD}/5$ in the same amount of time it takes to do the read. Size M_{13} , M_{14} , and M_{15} so that they have this effective resistance.
- (h) As a last step, we need to set the values of the column pull-up transistors of Figure 8.18. They need to pull the bitlines up after a write but before the end of the precharge phase. How big do they need to be? The final difference of the bitlines should be less than 10% of the desired read swings to avoid confusing the sense amp. Typically, we make them twice the size of the pull-down devices for a fast precharge. Size these transistors accordingly.
- (i) Check your results using SPICE.

Interconnect Design

CHAPTER OUTLINE

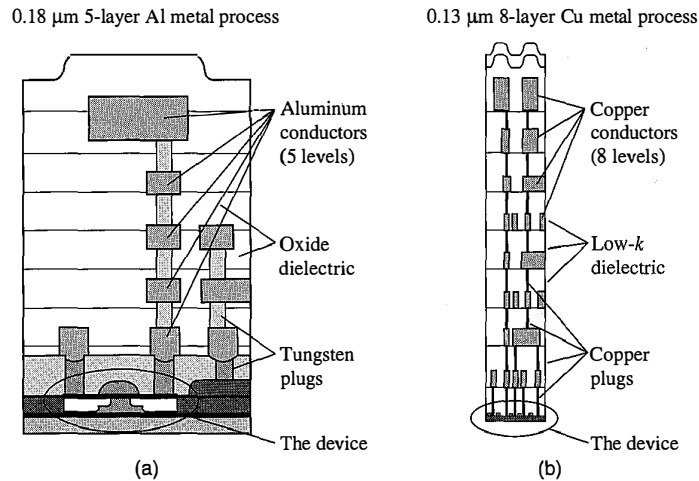
- 10.1 Introduction
- 10.2 Interconnect *RC* Delays
- 10.3 Buffer Insertion for Very Long Wires
- 10.4 Interconnect Coupling Capacitance
- *10.5 Interconnect Inductance
- *10.6 Antenna Effects
- 10.7 Summary

- References
- Problems

10.1 Introduction

This chapter is devoted to the study of interconnect issues that the IC designer faces when designing in deep submicron technologies. The relative importance of interconnect in deep submicron cannot be overstated. In fact, the topic of interconnect should actually appear much earlier in the text to stress its importance, but it is more appropriate at this stage since the discussion of transistor level circuit design has been completed. Initial coverage of interconnect topics was provided in Chapters 1 and 3. This chapter addresses the issues associated with *RLC* aspects of wires in detail. We begin by reexamining the *RC* delay calculation using the Elmore delay, and address the issue of buffer insertion in long wires. Then, we examine the capacitances associated with 3D interconnect. This topic is rather complex but we provide a simplified treatment to convey the most important concepts. This leads to a discussion of the effect of coupling capacitance on delay and noise injection in logic circuits. The chapter continues with a discussion of inductance effects and concludes with a brief look at antenna effects.

“Interconnect is everything” was an often-used expression starting in the mid-1990s to characterize the importance of interconnect in deep submicron technologies.

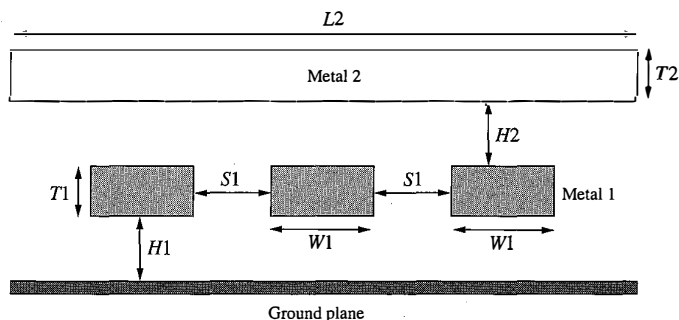
**Figure 10.1**

Cross sections of aluminum and copper processes.

While device sizes were shrinking with each technology generation, multilevel metal structures rose higher and higher above the surface of the silicon and soon began to dominate the landscape of the integrated circuit. This is illustrated in the cross-sectional views of 0.18 μm aluminum and 0.13 μm copper technologies in Figure 10.1. The number of layers of metal has increased from a few layers in the 1980s, to four or five layers in the 1990s, to eight or more layers today. The fabrication of these superstructures required significant advances in integrated circuit processing. However, the impact on design was even more significant. Suddenly, interconnect controlled timing, power, noise, design functionality, and reliability. It controlled all of these important electrical characteristics on the chip, even though the tiny devices (buried under a multilevel structure of wires and vias) defined the actual logic functions. Today interconnect still controls the design process in advanced technologies and will continue to do so for the foreseeable future.

The dimensions associated with a cross section of interconnect are shown in Figure 10.2. The rectangular wires are W wide and T thick, as shown. The separation between layers is a distance H (insulator thickness), and from other conductors is a distance S . The length of the wire, L , is shown only for Metal 2. The Metal 1 wires run into the page and so their lengths are not shown. It is important to realize that the vertical dimensions $H1$, $T1$, $H2$, and $T2$ are determined during the fabrication process. The designer has no control over these quantities. On the other hand, the horizontal dimensions $W1$, $S1$, $L1$, $L2$, etc. are all under the control of the designer.

There are a number of reasons to emphasize the importance of interconnect. First, as the width W of wires is decreased, the resistance increases. This increase in wire resistance causes an RC delay phenomenon that had not been experienced before

**Figure 10.2**

Wire dimensions for resistance and capacitance calculations.

1990. By the mid-1990s, a noticeable increase in wire delay led to widespread concern that it would soon be greater than gate delay. Second, the spacing S between wires has been decreasing to the point where the coupling between wires is significant. The resulting capacitive coupling introduces additional delay and noise effects that can cause failures in the design, requiring respins of silicon¹ in order to fix the problem. The overall term for all these problems is *signal integrity*. Recent issues of inductance in wires have been included in the growing list of signal integrity problems.

Other problems have surfaced due to the scaling of interconnect. The increase in resistance in the power supply lines due to scaling causes voltage drop along interconnect, referred to as *IR drop*. This affects the timing and functionality of gates connected to the power lines if the drop becomes too significant. A similar issue concerns the Ldi/dt drop in the power grid due to inductance. Still other effects such as charge storage on metal lines during manufacture, called *antenna effects*, lead to gate oxide breakdown or a threshold voltage shift. Further reliability problems exist due to the migration of the aluminum or copper material in the wires as high levels of current flow through them. Eventually, cracks and other imperfections in the metal cause material erosion until opens or shorts occur. This phenomenon is called *electromigration*, a long-term reliability issue.

Indeed, interconnect is more important than devices in many respects. This chapter and the next should be studied carefully as they contain valuable information pertaining to interconnect design issues. The material on interconnect in the open literature is rather extensive. Our purpose here is to convey the major issues and provide useful insight into the design methods to circumvent these issues. This chapter considers resistance, coupling capacitance, and inductance effects on delay and noise. Antenna effects are also described. The next chapter considers power grid

¹ A spin of silicon is the process of designing a chip and then having it fabricated. When errors are found in the design after fabrication, the whole process must be repeated. This is a very expensive and time-consuming exercise. Often several "spins of silicon" are needed to produce a working design for large complex chips.

and clock tree design in the context of interconnect issues. For further information, the reader is encouraged to consult the references at the end of the chapter.

10.2 Interconnect RC Delays

10.2.1 Wire Resistance

The delay due to interconnect has changed considerably due to deep submicron effects. It is our objective to develop models that can be used for accurate delay calculation when interconnect effects must be taken into account. So far, we have used a simple lumped capacitance to model the wire as shown in Figure 10.3a. We now consider the effect of wire resistance on delay. As the width dimension of the wires is reduced, the increased resistance has a pronounced effect on the delay. This was first observed at the 0.8 μm technology node at a number of semiconductor companies, but became a widespread issue at the 0.35 μm technology node. The interconnect model shown in Figure 10.3b is required when interconnect resistance is important.

The resistance of a material is given by

$$R = \frac{\rho L}{A} = \frac{\rho L}{TW} \quad (10.1)$$

where ρ is the resistivity of the material in $\Omega\text{-cm}$, L is the length of the wire, T is the thickness of the wire, and W is the wire width. The leading term in Equation (10.1) is an important one:

$$R_{sq} = \frac{\rho}{T} \quad (10.2)$$

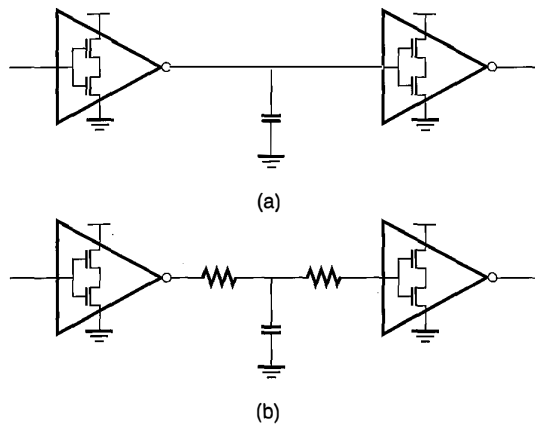


Figure 10.3

(a) Lumped capacitance model. (b) RC interconnect model.

where R_{sq} is the *sheet resistance* of the material and has the units of *ohms per square* ($\Omega/\square$). The sheet resistance of any metal layer can readily be computed from the resistivity and the thickness, T . To keep resistance relatively low in each technology generation, the thickness has not been scaled at the same rate as the minimum line width. However, the thickness of the metal lines will vary, depending on the metal layer. To simplify our description, we assume that two different thicknesses are used, one for the lower layers and one for the upper layers. Because the upper layers carry much larger currents over longer distances, they require a lower resistance, and therefore a higher thickness.

The two most common materials for interconnect are aluminum, with $\rho = 2.7 \mu\Omega\text{-cm}$, and copper, with $\rho = 1.7 \mu\Omega\text{-cm}$.² The metal hierarchies for a $0.18 \mu\text{m}$ aluminum process and a $0.13 \mu\text{m}$ copper process were shown in Figure 10.1. The approximate sheet resistances for a $0.18 \mu\text{m}$, 5-layer aluminum process assuming that metals 1 to 4 have a thickness of $0.5 \mu\text{m}$ and metal 5 has a thickness of $1.0 \mu\text{m}$ are as follows:

$$R_{sq} = \frac{\rho}{T} = \frac{2.7 \mu\Omega\text{-cm}}{0.5 \mu\text{m}} = 54 \text{ m}\Omega/\square \quad \text{Al metals 1-4}$$

$$R_{sq} = \frac{\rho}{T} = \frac{2.7 \mu\Omega\text{-cm}}{1.0 \mu\text{m}} = 27 \text{ m}\Omega/\square \quad \text{Al metal 5} \quad (10.3)$$

The approximate sheet resistances for a $0.13 \mu\text{m}$, 8-layer copper process assuming that metals 1–6 have a thickness of $0.4 \mu\text{m}$ and metals 7–8 have a thickness of $0.8 \mu\text{m}$ are as follows:

$$R_{sq} = \frac{\rho}{T} = \frac{1.7 \mu\Omega\text{-cm}}{0.4 \mu\text{m}} = 42 \text{ m}\Omega/\square \quad \text{Cu metals 1-6}$$

$$R_{sq} = \frac{\rho}{T} = \frac{1.7 \mu\Omega\text{-cm}}{0.8 \mu\text{m}} = 21 \text{ m}\Omega/\square \quad \text{Cu metals 7-8} \quad (10.4)$$

Given these parameters, the actual resistance of a segment of wire is:

$$R = R_{sq} \left(\frac{L}{W} \right) \quad (10.5)$$

The ratio of L/W is referred to as the *number of squares* of wire. It is the aspect ratio of the wire in terms of length and width. Designers count squares of interconnect as a way of obtaining a relative resistance value for a wire since R_{sq} is fixed for a given technology. As we know by now, the number of squares of resistance has continued to increase over the years since L is increasing and W is decreasing for the longest wires on the chip. Therefore, these are the wires that cause significant RC delay problems.

² As a point of comparison, tungsten plugs have a much higher resistivity of $5.5 \mu\Omega\text{-cm}$.

10.2.2 Elmore Delay Calculation

In long wires, the increased resistance presents a distributed RC load to a driving gate, as shown in Figure 10.4. For simplicity, the driver is represented as an ideal voltage source. The distributed RC line has been converted to a lumped RC ladder structure with n segments. The total resistance for the line is the sum of all the individual resistances, $R_{\text{wire}} = nR_w$. The same holds for the total capacitance, $C_{\text{wire}} = nC_w$.

The wire capacitance in the delay calculation is computed using the capacitance per unit length of interconnect, C_{int} . In modern technologies, there are multiple levels of metal and each one has a different C_{int} . In fact, as we shall see later, interconnect capacitance is distributed in nature and highly dependent on the layout. Today, there is more coupling capacitance between lines than there is capacitance to ground. For the moment, we use an average capacitance value and assume that all layers have the same C_{int} for hand calculations. The effective wire capacitance for metal interconnect is $C_{\text{int}} = 0.2 \text{ fF}/\mu\text{m}$ for densely packed lines ($3\text{--}4\lambda$ apart) and of the order of $C_{\text{int}} = 0.1 \text{ fF}/\mu\text{m}$ for loosely packed lines ($30\text{--}40\lambda$ apart). We will often use $C_{\text{int}} = 0.2 \text{ fF}/\mu\text{m}$ for the capacitance for an interconnect line to be somewhat pessimistic.

To determine the delay of an RC ladder, consider an aluminum wire in $0.18 \mu\text{m}$ technology. Assume a wire width of $W = 0.4 \mu\text{m}$, and $C_{\text{int}} = 0.2 \text{ fF}/\mu\text{m}$ and $R_{\text{sq}} = 0.054 \Omega/\square$. Then for a length of wire 1 mm , the total resistance is $R_{\text{wire}} = 0.054 \Omega/\square \times 1000/0.4 = 125 \Omega$ and the total capacitance is $C_{\text{wire}} = 0.2 \text{ fF}/\mu\text{m} \times 1000 \mu\text{m} = 200 \text{ fF}$. It is tempting to compute the total RC delay as $0.7 R_{\text{wire}} C_{\text{wire}} = (0.7) 125 \Omega \times 200 \text{ fF} = 17.5 \text{ ps}$. However, the actual delay is closer to 10 ps . Clearly, we need an accurate way to compute the delay of such RC ladders.

By multiplying R_{wire} and C_{wire} together, we assumed that the response was exponential in nature, as shown in Figure 10.5a. In actual fact, the response of a distributed RC line is given in the graph of Figure 10.5b. The key to developing a convenient way of computing the delay for a ladder network is to examine the area under the two graphs and establish a relationship between the two responses.

The area underneath the curve in Figure 10.5a is called the *first moment of the impulse response* and is given by

$$\text{Area} = \int_0^{\infty} e^{-t/RC} dt = R_{\text{wire}} C_{\text{wire}}$$

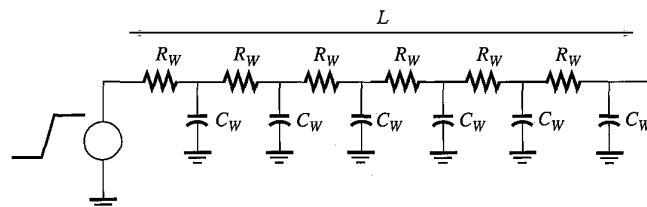
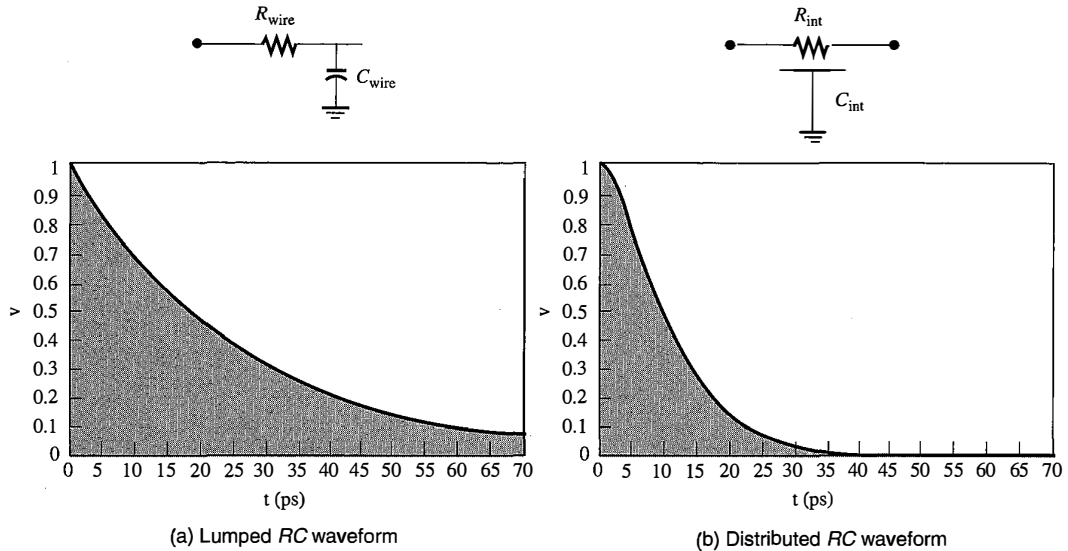


Figure 10.4
Distributed RC line as a lumped RC ladder.

**Figure 10.5**

Lumped RC versus distributed RC output waveforms.

This gives us exactly the time constant for a lumped RC circuit. Apparently the area under the curve provides first-order delay information. If we apply this observation to the waveform for the distributed RC given by Figure 10.5b, we find that the area under this curve is some value which we call τ given by

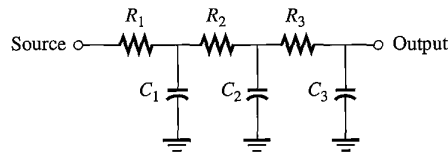
$$\text{Area} = \int_0^{\infty} v(t) dt = \tau$$

This quantity is also the first moment of the impulse response but is more commonly referred to as the *Elmore delay* by circuit designers. It can be shown that this is the dominant time constant of the network and is a good estimate of the delay. While this value is difficult to compute in general, it is fairly straightforward for RC networks. The Elmore delay for an RC ladder involves visiting each node in the circuit and multiplying the capacitance at that node by the sum of all the resistances from the source node to the node being visited.

To further illustrate this procedure, consider the circuit in Figure 10.6. It has an Elmore delay of

$$\tau = R_1 C_1 + (R_1 + R_2) C_2 + (R_1 + R_2 + R_3) C_3$$

We can examine two extreme cases of this delay value to see if the Elmore delay is consistent. Specifically, if $C_1 = C_2 = 0$, then the Elmore delay and the RC time constant are equal to $(R_1 + R_2 + R_3) C_3$. Similarly if $R_2 = R_3 = 0$, then the Elmore delay and the RC time constant are both $R_1(C_1 + C_2 + C_3)$. Since both cases are

**Figure 10.6**

RC ladder for Elmore delay calculation.

correct, we have some confidence the Elmore delay is a reasonable delay estimator. In fact, it is the most popular form of delay calculation for RC ladder and tree circuits.

For a general network, we can compute the Elmore delay as

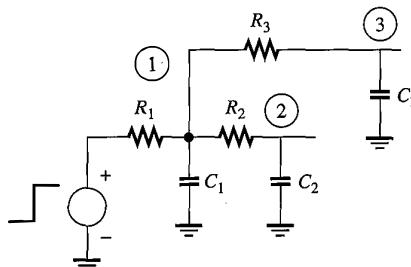
$$\tau_i = \sum_k (C_k \times R_{ik}) \quad (10.6)$$

where the node of interest is node i , C_k is the capacitance at node k , and R_{ik} is the sum of all the *resistances in common* from the source to node i and the source to node k . In other words, we visit each node and multiply the capacitance at that node by a resistance value that is the sum of the resistances that are common when we draw two paths: one from the source to node i and another from the source to node k . The process is best illustrated with an example.

Example 10.1 Elmore Delay of a Simple RC Tree

Problem:

Compute the Elmore delay from the input to nodes 1, 2, and 3 in the following RC tree. The Elmore delay is not accurate for internal nodes such as node 1. However, we are not usually interested in the delay from the input to an internal node in RC trees. In any case, write the expression for the time constant due to Elmore for each node.



Solution:

$$\tau_1 = R_1 C_1 + R_1 C_2 + R_1 C_3$$

$$\tau_2 = C_1 R_1 + C_3 R_1 + C_2 (R_1 + R_2)$$

$$\tau_3 = C_1 R_1 + C_2 R_1 + C_3 (R_1 + R_3)$$

10.2.3 RC Delay in Long Wires

For a long wire, we can compute the RC delay in terms of the total resistance and total capacitance. Consider Figure 10.4 once again. If the total wire length is L , and each segment is ΔL in length, then $L = n\Delta L$. For convenience, set the wire resistance per unit length to $R_{\text{int}} = r$ and the wire capacitance unit length to $C_{\text{int}} = c$. The total resistance is $R_{\text{wire}} = rL$ and the total capacitance is $C_{\text{wire}} = cL$. Using the Elmore delay calculation assuming n segments, we obtain

$$\begin{aligned} \tau_{\text{Elmore}} &= (r\Delta L)(c\Delta L) + 2(r\Delta L)(c\Delta L) + \cdots + n(r\Delta L)(c\Delta L) \\ &= (\Delta L)^2 rc(1 + 2 + \cdots + n) \\ &= (\Delta L)^2 rc(n)(n+1)/2 \\ &\approx (\Delta L)^2 rcn^2/2 = L^2 rc/2 = R_{\text{wire}} C_{\text{wire}}/2 \end{aligned} \quad (10.7)$$

The actual propagation delay value is closer to $0.38 R_{\text{wire}} C_{\text{wire}}$ which is slightly less than what Elmore predicts, so Elmore can be viewed as being somewhat pessimistic. Note that, according to Equation (10.7), the *delay increases as the square of the length*. This alerts us to the fact that long wires will create significant delays.

In practice, there must be a buffer or inverter driving the interconnect as shown in Figure 10.7. The driver can be represented as a source voltage and a source resistance. The source voltage is either Gnd or V_{DD} depending on the direction of the switching event. The source resistance is the on-resistance of the buffer.

We should always compute the total delay with the buffer attached to the interconnect. In addition, we must include all fanout loading capacitances on the interconnect for accurate delay estimation. To simplify the calculation, we replace the distributed RC line with a lumped RC model after computing R_{wire} and C_{wire} of the line. There are three forms of RC models as shown in Figure 10.8. While they all

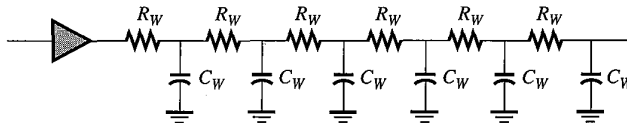
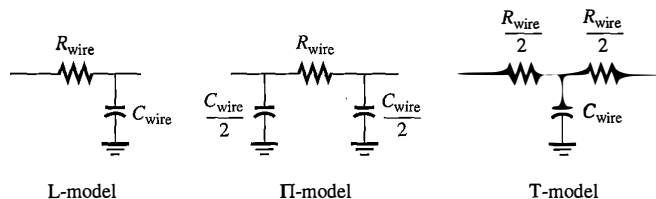


Figure 10.7

Driver and interconnect.

**Figure 10.8**

Three possible lumped RC models.

appear to be similar in form, they are not all suitable for the purposes of delay calculation. We can evaluate each one by computing its corresponding time constant.

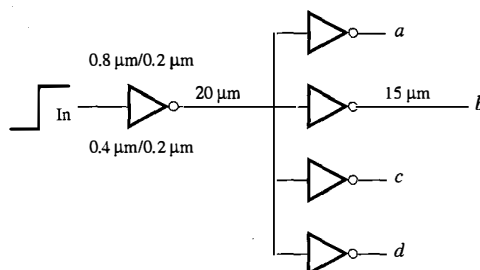
First, the L-model produces a time constant of $R_{\text{wire}} C_{\text{wire}}$. We already know from Equation (10.7) that the delay should be $R_{\text{wire}} C_{\text{wire}}/2$ so this model should not be used. The second model is the π -model where the capacitance has been divided in half and assigned to the input and output. The time constant is $R_{\text{wire}} C_{\text{wire}}/2$. The third model is a T-model where the resistance has been divided in half and assigned to the input and output. Here, the time constant is also $R_{\text{wire}} C_{\text{wire}}/2$. Therefore, the L-model is not an accurate model for long distributed RC lines, whereas the π -model and the T-model produce the correct answer, so they are both suitable for the delay calculation. The T-model has an extra node that may increase the number of calculations. As a result, the π -model is the most popular lumped model for a distributed RC line.

Example 10.2 Delay with and without Resistance Effects for Short Wires

Problem:

What is the propagation delay through the first inverter shown below, ignoring the wire resistance? Assume all inverters are equal in size. Include all capacitances and use $0.18\text{ }\mu\text{m}$ technology parameters. Recompute the delay including interconnect resistance. Assume that the metal is Al and that the width of the wire is $0.2\text{ }\mu\text{m}$. Is wire resistance significant?

Parameters: $C_{\text{eff}} = 1\text{ fF}/\mu\text{m}$, $C_{\text{int}} = 0.2\text{ fF}/\mu\text{m}$, $C_g = 2\text{ fF}/\mu\text{m}$.



Solution:

Fanout capacitance: inverter is driving four identical inverters.

$$C_{\text{fanout}} = 4 \times C_g(W_N + W_P) = 4(2 \text{ fF}/\mu\text{m})(0.8 \mu\text{m} + 0.4 \mu\text{m}) = 9.6 \text{ fF}$$

Self-capacitance:

$$C_{\text{self}} = C_{\text{eff}}(W_N + W_P) = (1 \text{ fF}/\mu\text{m})(0.8 \mu\text{m} + 0.4 \mu\text{m}) = 1.2 \text{ fF}$$

Wire capacitance:

$$C_{\text{wire}} = C_{\text{int}}L_{\text{wire}} = (0.2 \text{ fF}/\mu\text{m})(20 \mu\text{m}) = 4 \text{ fF}$$

Total capacitance:

$$C_{\text{load}} = C_{\text{fanout}} + C_{\text{self}} + C_{\text{wire}} = 9.6 + 1.2 + 4 = 14.8 \text{ fF}$$

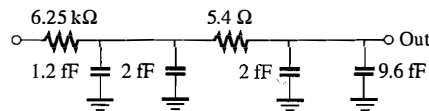
Based on this total capacitance, the delay assuming a ramp input is given by

$$\begin{aligned}\tau_{\text{driver}} &= R_{\text{eff}}C_{\text{load}} = (12.5k)\left(\frac{L}{W}\right)(14.8 \text{ fF}) \\ &= (12.5k)\left(\frac{1}{2}\right)(14.8 \text{ fF}) = 6.25k(14.8 \text{ fF}) = 92.5 \text{ ps}\end{aligned}$$

To include wire resistance, we first compute the resistance for the $20 \mu\text{m}$ wire:

$$R_{\text{wire}} = R_{\text{sq}}\left(\frac{L}{W}\right) = 54 \text{ m}\Omega/ \times \left(\frac{20 \mu\text{m}}{0.2 \mu\text{m}}\right) \approx 5.4 \Omega$$

Next, use π -model for the wire:



Use Elmore delay calculation:

$$\begin{aligned}\tau_{\text{driver}} &= (6.25 \text{ k}\Omega)(3.2 \text{ fF}) + (6.25 \text{ k}\Omega + 5.4 \Omega)(11.6 \text{ fF}) \\ &= 92.6 \text{ ps}\end{aligned}$$

Conclusion: the inclusion of wire resistance made no appreciable difference to the solution.

The above example illustrates that short wires can (and should) be handled as a lumped capacitance with little or no error in the calculations. The definition of a short wire depends on the technology but typically wires that are a few microns long or as high as $20 \mu\text{m}$ should be handled as a lumped capacitance. Over 90% of the wires in a chip are short wires and, therefore, can be computed in this way.

While the analysis above is appropriate when the length of interconnect is relatively short, it is not appropriate when the interconnect line is long. In this case,

the wire should be treated as a distributed RC line rather than a simple lumped capacitance. Consider the example below that illustrates the impact of a long wire.

Example 10.3 Delay for Inverter Driving a Long Wire

Problem:

Consider an isolated metal 5 Al wire that is 20 mm long and driven by a 100X inverter (i.e., 100 times larger than the minimum size inverter). Since the wire is isolated, its capacitance is $0.1 \text{ fF}/\mu\text{m}$. Estimate the total line resistance and the total line capacitance assuming that the width of the wire is $0.5 \mu\text{m}$. What is the propagation delay as calculated by Elmore for the π -model from the input of the inverter to the end of the wire?

Solution:

The wire resistance is

$$R_{\text{wire}} = R_{\text{int}} L = \left(\frac{0.027 \Omega/\square}{0.5 \mu\text{m}} \right) (20,000 \mu\text{m}) = 1080 \Omega$$

The wire capacitance is

$$C_{\text{wire}} = C_{\text{int}} L = (0.1 \text{ fF}/\mu\text{m}) (20,000 \mu\text{m}) = 2 \text{ pF}$$

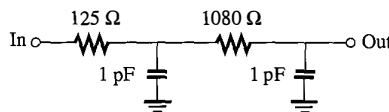
The inverter on-resistance is

$$R_{\text{eff}} = R_{\text{eqn}}/100 = 12.5 \text{ k}\Omega/100 = 125 \Omega$$

The inverter output capacitance is

$$C_{\text{self}} = C_{\text{eff}}(2W + W)100 = (1 \text{ fF}/\mu\text{m})(0.6 \mu\text{m})100 = 60 \text{ fF}$$

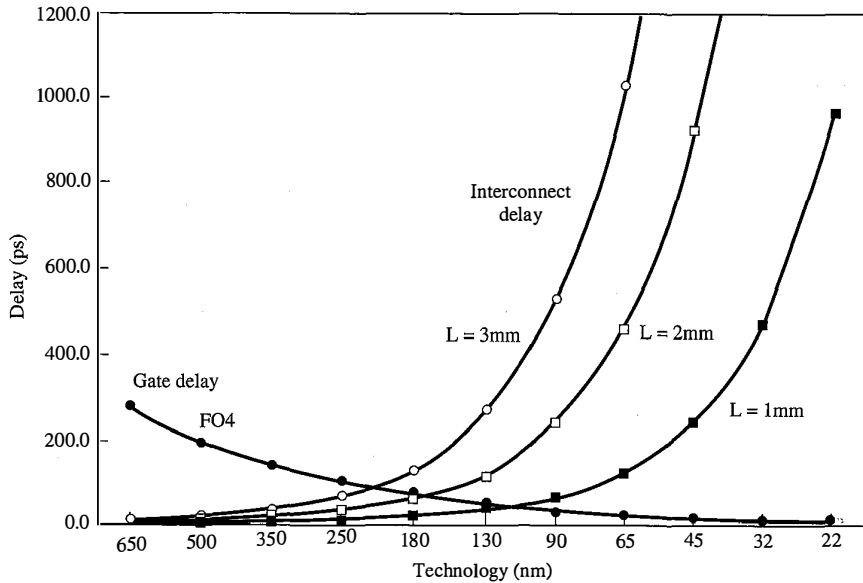
We can ignore the junction capacitance since it is rather small. Using the π -model, the wire capacitance is split in half while the full wire resistance is used in the model.



The resulting Elmore delay is

$$\tau_{\text{Elmore}} = (125 \Omega)(1 \text{ pF}) + (125 \Omega + 1080 \Omega)(1 \text{ pF}) \approx 1.3 \text{ ns}$$

In order to appreciate the large delays associated with long wires, we plot the FO4 gate delay and the wire delay for a number of different wire lengths as technology scales in Figure 10.9. As reference points, the FO4 delay for 350 nm is approximately 150 ps; for 130 nm, the FO4 delay is reduced to approximately 40 ps; for 65 nm, the value reduces to about 25 ps. On the other hand, for $L = 1 \text{ mm}$ the interconnect delay increases from 3 ps, to 27 ps, to 116 ps, respectively. For $L = 2 \text{ mm}$ the delay

**Figure 10.9**

Gate delay versus interconnect delay.

values are 12 ps, 110 ps, and 460 ps, respectively. For $L = 3$ mm the delay values are 30 ps, 240 ps, and 1035 ps, respectively. The wire delay doubles with each technology node and increases quadratically as a function of wire length. Clearly, when a single buffer drives a long wire, the interconnect delay will eventually dominate the FO4 delay as technology scales or as the length increases.

We can define a rule of thumb for long wires by setting the FO4 delay equal to the interconnect delay in each technology. A critical wire length can be derived as follows:

$$\text{FO4} = 0.38R_{\text{int}}C_{\text{int}}L_{\text{crit}}^2$$

$$\therefore L_{\text{crit}} = \sqrt{\frac{\text{FO4}}{0.38R_{\text{int}}C_{\text{int}}}}$$

For example, in a $0.18 \mu\text{m}$ technology, the value of L_{crit} can be computed using this equation or estimated from the graph in Figure 10.9 to be approximately 2.5 mm. Of course, this value will vary depending on the values of R_{int} and C_{int} for a given wire.

10.3 Buffer Insertion for Very Long Wires

For very long wires, the quadratic delay characteristics described above cannot be tolerated in any design. The interconnect delay would be so long that it would dominate the gate delay. In ICs, global wires tend to be very long since they must be distributed

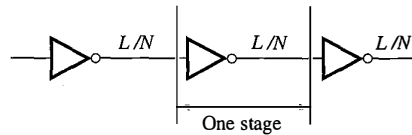
all over the chip. These wires are associated with clocks, busses, and other major signals in the design. Since the delay for a long wire is quadratic with respect to L , the standard solution is to insert *repeaters* or *buffers* periodically along the wire. This effectively shortens the wire into smaller segments. The buffers act to convert the quadratic delay to a more linear delay. Of course, if we insert too many buffers, the delay increases due to the buffers themselves. But with too few buffers, we are prone to the quadratic effect. Finding the optimum number of equal-sized buffers needed to minimize the overall delay is referred to as the *buffer insertion problem*.

Consider a wire of length L that has an unacceptably long delay without any buffers driving it. If the per unit length resistance and capacitance are given by R_{int} and C_{int} , respectively, the total resistance and capacitance of the wire is given by

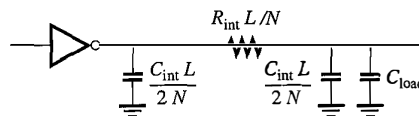
$$R_{\text{wire}} = \frac{\rho}{T} \frac{L}{W} = R_{\text{sq}} \frac{L}{W} = R_{\text{int}} L$$

$$C_{\text{wire}} = C_{\text{int}} L$$

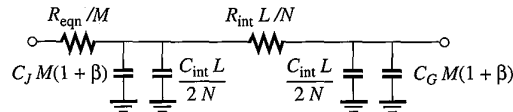
To reduce the delay we seek to insert a number of buffers that produce the minimum delay. Let N be the number of such buffers (including the first buffer) and L/N be the corresponding wire lengths between buffers as shown in Figure 10.10a. Each segment consists of a buffer driving a wire load and a downstream gate load. A π -model may be used for the interconnect in one segment as shown in Figure 10.10b.



(a) Wire of length L with N buffers inserted



(b) One segment of buffer and interconnected



(c) RC model for one segment

Figure 10.10

Buffer insertion for very long wires.

Assuming that a minimum size buffer is 1X, the corresponding input and output capacitances are as follows:

$$C_{out} = C_{eff}W(1 + \beta) = C_J(1 + \beta) \quad (\text{in fF})$$

$$C_{in} = C_gW(1 + \beta) = C_G(1 + \beta) \quad (\text{in fF})$$

Here, β is the ratio of the PMOS to NMOS device size. The value of C_J is $C_{eff} \times W$, where W is the size of the NMOS device for a 1X inverter. Similarly, C_G is $C_g \times W$.

Let the multiplier M represent the optimal buffer size for the buffer insertion problem. Then, for a buffer which is M times larger than a 1X buffer:

$$R_{eff} = R_{eqn}/M$$

$$C_{out} = C_{eff}W(1 + \beta)M = C_J(1 + \beta)M$$

$$C_{in} = C_gW(1 + \beta)M = C_G(1 + \beta)M$$

In Figure 10.10c, we illustrate the RC model used for each stage during delay calculation. Using the Elmore delay, the delay for each segment is given by

$$t_{segment} = \frac{R_{eqn}}{M} \left(C_J M(1 + \beta) + \frac{C_{int}L}{2N} \right) + \left(\frac{R_{eqn}}{M} + \frac{R_{int}L}{N} \right) \left(\frac{C_{int}L}{2N} + C_G M(1 + \beta) \right)$$

The total delay is N times the segment delay. After some rearrangement, we find that it produces a quadratic equation with respect to L .

$$t_{total} = N \times t_{segment} = N(C_G + C_J)R_{eqn}(1 + \beta) + \left(C_G(1 + \beta)R_{int}M + \frac{C_{int}R_{eqn}}{M} \right)L + \left(\frac{C_{int}R_{int}}{2N} \right)L^2 \quad (10.8)$$

When we examine each delay term, we find that the first term is due solely to the buffer while the last term is due solely to the interconnect. The middle term has two components: one is due to the wire resistance driving the input capacitance of the next buffer, and a second RC term is due to the buffer driving the wire capacitance.

To find the optimal values for N and M , we apply the derivative with respect to each of the variables, as follows. The number of wire segments is

$$\frac{\partial t_{total}}{\partial N} = 0 = R_{eqn}(C_J + C_G)(1 + \beta) - \frac{R_{int}C_{int}L^2}{2N^2}$$

$$\therefore N = \sqrt{\frac{R_{int}C_{int}L^2/2}{R_{eqn}(C_J + C_G)(1 + \beta)}} = \sqrt{\frac{t_{wire}}{FOI}} \quad (10.9)$$

The buffer size, assuming all buffers are equal, is

$$\frac{\partial t_{\text{total}}}{\partial M} = 0 = R_{\text{int}} L C_G (1 + \beta) - \frac{R_{\text{eqn}} C_{\text{int}} L}{M^2}$$

$$\therefore M = \sqrt{\frac{R_{\text{eqn}}}{C_G (1 + \beta)} \frac{C_{\text{int}}}{R_{\text{int}}}} \quad (10.10)$$

Using these values of N and M , we can obtain the optimal delay through the interconnect. We can further understand these values by examining the total delay equation provided in Equation (10.8). The variable N exists in the first and last term of t_{total} ; the terms are due to the buffer and interconnect, respectively, each one acting as if the other did not exist. If we set these two terms to be equal to one another, we arrive at the equation for the optimal N . The expression under the square-root sign in Equation (10.9) can be viewed as the interconnect delay (t_{wire} according to Elmore) divided by the inverter delay when driving an identical inverter (FO1 delay).

The middle term of the total delay expression in Equation (10.8) is dependent on M . If we set the two expressions in the middle term to be equal to one another, we arrive at the optimal M . This value does not depend on the length of the wire but rather the physical characteristics of the transistors and wires. In fact, this quantity depends only on technology parameters of the wire and the minimum-size inverter. Therefore, we can compute this value once the process parameters have been determined. Note that precise values of the terms in the expressions are needed to produce accurate results.

It is useful to compare and contrast the buffer insertion problem to the logical (LE) effort problem described in Chapter 6. For buffer insertion, we are concerned with driving a long wire so we break up the wire into segments and insert very large equal-sized buffers between the wire segments. For logical effort, the objective is to drive a very large output capacitance starting from a small input capacitance. The solution is obtained by determining the number of buffers and then sizing up the buffers by a given factor, typically by 4X. In LE, the delay through each buffer stage is the same at the optimal solution. In the case of buffer insertion, the delay due to the buffer and the wire are the same at the optimal point.

Example 10.4 Buffer Insertion Example

Problem:

From Example 10.3, consider the 20 mm Al metal 5 wire again, but this time reduce the delay using optimal buffer insertion. What is the new delay through the wire after buffer insertion?

Solution:

Using the following values:

$$R_{\text{int}} = 54 \text{ m}\Omega/\mu\text{m} \quad C_{\text{int}} = 0.1 \text{ fF}/\mu\text{m} \quad L = 20,000 \mu\text{m}$$

$$R_{\text{eqn}} = 12.5 \text{ k}\Omega/\square \quad C_g = 2 \text{ fF}/\mu\text{m} \quad C_{\text{eff}} = 1 \text{ fF}/\mu\text{m} \quad W = 0.2 \mu\text{m} \quad \beta = 2$$

The number of segments in the wire is

$$N = \sqrt{\frac{R_{\text{int}} C_{\text{int}} L^2 / 2}{R_{\text{eqn}} (C_G + C_I) (1 + \beta)}} = 6.9 \approx 7$$

The buffer size, assuming all equal buffers, is

$$M = \sqrt{\frac{R_{\text{eqn}} C_{\text{int}}}{C_G (1 + \beta) R_{\text{int}}}} \approx 140$$

Then, the buffer resistance is

$$R_{\text{eff}} = \frac{12.5k\Omega}{140} \approx 90\Omega$$

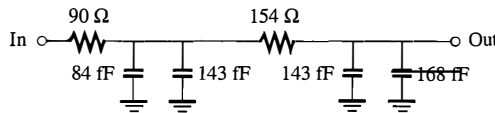
The buffer output capacitance is:

$$C_{\text{self}} = C_{\text{eff}} (2W + W) 140 = (1 \text{ fF}/\mu\text{m}) (0.6 \mu\text{m}) 140 = 84 \text{ fF}$$

The buffer input capacitance is

$$C_{\text{fanout}} = C_g (2W + W) 140 = (2 \text{ fF}/\mu\text{m}) (0.6 \mu\text{m}) 140 = 168 \text{ fF}$$

Each segment produces this RC delay model.



The total Elmore delay is

$$\begin{aligned} \tau_{\text{Elmore}} &= 7 \times [(90\Omega)(84 \text{ fF} + 143 \text{ fF}) + (90\Omega + 154\Omega)(143 \text{ fF} + 168 \text{ fF})] \\ &= 0.67 \text{ ns} \end{aligned}$$

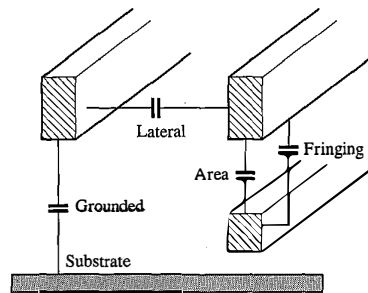
The total delay has been cut in half using buffer insertion.

10.4 Interconnect Coupling Capacitance

In this section, the issues associated with coupling capacitance are described. The goal is to understand the various components of wire capacitance and to develop formulas for hand calculation. In reality, IC interconnect structure is three-dimensional in nature as seen in Figures 1.1 and 1.2 of Chapter 1. However, for hand calculations, we can develop simple formulae that work well in practice. Thus far, we have been using $0.2 \text{ fF}/\mu\text{m}$ for wire capacitance without really knowing its origin or range of validity. This section attempts to clarify its origin and justify the use of this value from a modeling perspective.

10.4.1 Components of Coupling Capacitance

In modern integrated circuits, interconnect capacitance modeling is a complex task. The capacitance of such wires is dependent on their topology, on the distances from other wires above or below, and on the separation between wires. To acquire a basic

**Figure 10.11**

Area, lateral, and fringing components.

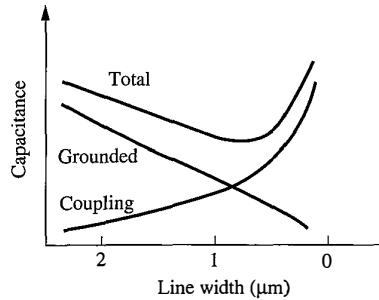
understanding of the problem, we can use simple first-order models to describe interconnect capacitances and derive useful constants for design purposes.

Before beginning the analysis, we describe the different components of capacitance. These components are all shown in Figure 10.11. The first capacitance worth noting is between a line and the substrate, called the *grounded* capacitance. This was the dominant capacitance in the past, but its value has been diminishing due to scaling effects. The next capacitance to note is between metal lines on the same layer. This is called the *lateral* capacitance. This is the main coupling capacitance that is the source of delay and noise issues. This capacitance has been increasing over the years. There are two other capacitances that may give rise to coupling between lines. One is referred to as the *fringing* capacitance which occurs between various edges and surfaces between two crossing lines. One of many such fringing capacitances is shown in the figure.³ The other type of coupling capacitance occurs when two wires on different metal layers overlap. It is called the *area* capacitance.

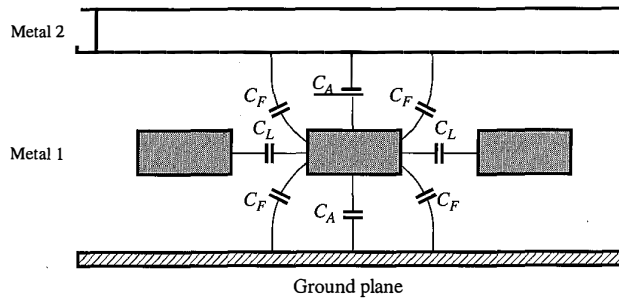
Over the years, coupling capacitance has been increasing while the grounded capacitance has been decreasing. The general trend is depicted in Figure 10.12, where the coupling capacitance due to area, lateral, and fringing components is overtaking the grounded component as the line widths shrink. In a typical $0.13\text{ }\mu\text{m}$ design, over 70% of the wire capacitance at a node may be due to coupling between wires. This is usually the case for long nets, although short nets may also have significant coupling to their neighbors. Most nets have significant junction and gate capacitances due to the logic gates that increase the overall grounded capacitance. However, the coupling to other nodes is still significant.

To obtain a quantitative view of coupling capacitance, consider the interconnect structure shown in Figure 10.13. We would like to determine the capacitance of the center wire on a component by component basis so that it can be used for delay and noise calculations. The capacitances associated with the middle conductor are given in Figure 10.13. The dimensions of the wires were shown earlier in Figure 10.2. The total values of area, lateral, and fringing capacitances are all indicated in the figure. The grounded capacitance is treated as an area capacitance since the

³ Actually, there are many other fringe capacitances that are not shown in the figure to keep it simple.

**Figure 10.12**

Coupling capacitance, grounded capacitance, and total capacitance.

**Figure 10.13**

Capacitances associated with middle conductor.

calculation will be the same for both. While the capacitance cannot be exactly partitioned as shown, we model it this way for convenience. Once computed, we sum them together to compute the total wire capacitance. We now compute each one in units of capacitance per unit length.

The easiest component to compute is the area capacitance per unit length. There are two such capacitances, one from the conductor to ground and the other from the conductor to Metal 2 above it. This is illustrated separately in Figure 10.14. Assuming silicon dioxide with a dielectric constant of 4 is used as the insulating material, each of these capacitances is computed as

$$C_{\text{area}} = \epsilon_{\text{ox}} \frac{W}{H} \quad (10.11)$$

The numerical value of this capacitance is

$$C_{\text{area}} = \epsilon_{\text{ox}} \frac{W}{H} = 4\epsilon_0 \frac{W}{H} = 4(88.5 \times 10^{-4} \text{ fF}/\mu\text{m}) \frac{W}{H} = 0.035 \frac{W}{H} \text{ fF}/\mu\text{m}$$

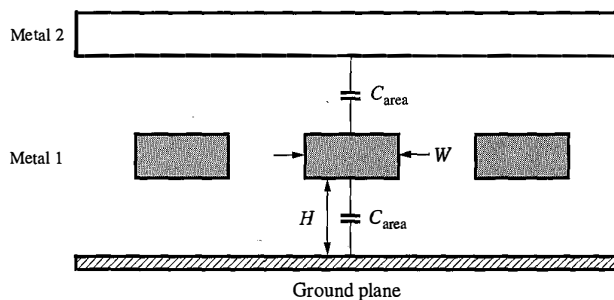


Figure 10.14
Modeling area capacitance.

As W shrinks in value with technology scaling, the area capacitance reduces linearly. In $0.18\text{ }\mu\text{m}$ technology, the upper layer of metal would have $W = 0.4\text{ }\mu\text{m}$ and $H = 0.5\text{ }\mu\text{m}$. Therefore, C_{area} is approximately $0.03\text{ fF}/\mu\text{m}$.

Next, we examine the lateral capacitance. This value can be complicated to compute but there are two limiting situations that lend themselves to a straightforward formula. These two situations are shown in Figure 10.15. When the conductors are

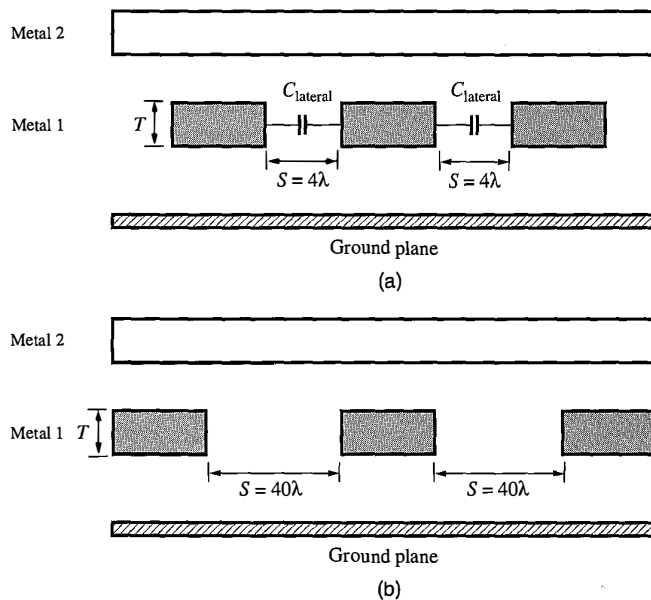


Figure 10.15
Modeling lateral capacitance.

very closely spaced by 4λ , as shown in Figure 10.15a, the capacitance can be computed using the parallel-plate formula:

$$C_{\text{lateral}} = \epsilon_{\text{ox}} \frac{T}{S} \quad (10.12)$$

The capacitance is inversely proportional to the spacing, S . Applying the dielectric constant value to the equation:

$$C_{\text{lateral}} = \epsilon_{\text{ox}} \frac{T}{S} = 4\epsilon_0 \frac{T}{S} = 4(88.5 \times 10^{-4} \text{ fF}/\mu\text{m}) \frac{T}{S} = 0.035 \frac{T}{S} \text{ fF}/\mu\text{m}$$

In our generic $0.18 \mu\text{m}$ technology, the upper layer metal has $T = 0.8 \mu\text{m}$ and $S = 0.4 \mu\text{m}$. Therefore, C_{lateral} is $0.070 \text{ fF}/\mu\text{m}$. When the conductors are spaced out by 40λ as in Figure 10.15b, then C_{lateral} is $0.007 \text{ fF}/\mu\text{m}$ and can effectively be ignored.

The last capacitance to consider is the fringing capacitance. Again, its complexity prohibits any sort of simplified analysis. However, for hand calculations, it is sufficient to consider two cases as shown in Figure 10.16. The first case in Figure 10.16a consists of densely packed wires where the lateral capacitance dominates. For this case, C_{fringe} is approximately zero. However, if the conductors are widely spaced, as in Figure 10.16b, there is a fringe capacitance on each edge of approximately

$$C_{\text{fringe}} = \epsilon_{\text{ox}} \ln\left(1 + \frac{T}{H}\right) \quad (10.13)$$

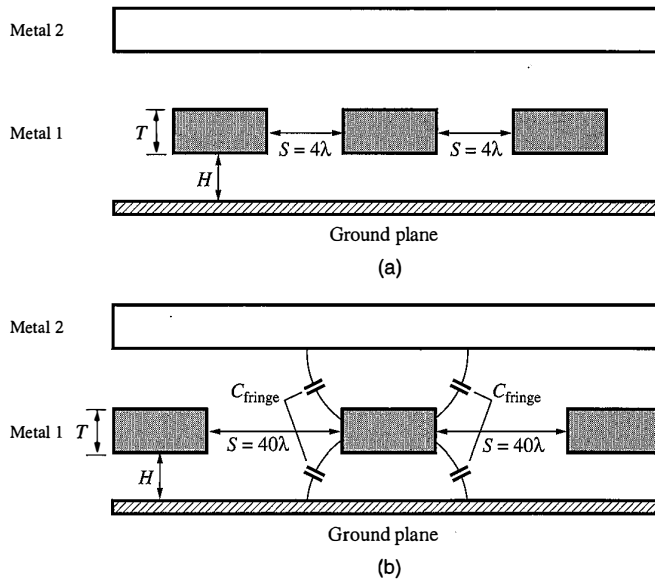


Figure 10.16
Modeling fringing capacitance.

Since T and H are fixed by the technology, this capacitance is almost independent of the topology. While this is not strictly true, it is sufficient for our modeling purposes here. We can compute the value of the fringing component as follows:

$$\begin{aligned} C_{\text{fringe}} &= \epsilon_{\text{ox}} \ln \left(1 + \frac{T}{H} \right) = 4(88.5 \times 10^{-4} \text{ fF}/\mu\text{m}) \ln \left(1 + \frac{0.5}{0.5} \right) \\ &= 0.025 \text{ fF}/\mu\text{m} \end{aligned}$$

Now that each of the components has been derived, we are ready to compute the quantity C_{int} .

The total capacitance for the middle wire depends on whether the wires are closely spaced or widely spaced. The case for the closely spaced wires on the upper layers is given by

$$\begin{aligned} C_{\text{int}} &= 2C_{\text{area}} + 2C_{\text{lateral}} + 2C_{\text{fringe}} \\ &\cong 2(0.03) + 2(0.070) + 2(0.0) \text{ fF}/\mu\text{m} \\ &= 0.2 \text{ fF}/\mu\text{m} \end{aligned}$$

If the wires are widely spaced, we can use the following formulation:

$$\begin{aligned} C_{\text{int}} &= 2C_{\text{area}} + 2C_{\text{fringe}} + 2C_{\text{lateral}} \\ &\cong 2(0.03) + 2(0.025) + 2(0.0) \text{ fF}/\mu\text{m} \\ &\approx 0.1 \text{ fF}/\mu\text{m} \end{aligned}$$

Now consider the closely spaced and widely spaced cases for the lower level metal lines. The case for the closely spaced wires is given by

$$\begin{aligned} C_{\text{int}} &= 2C_{\text{area}} + 2C_{\text{lateral}} + 2C_{\text{fringe}} \\ &= 2(0.015) + 2(0.035) + 2(0.0) \text{ fF}/\mu\text{m} \\ &= 0.1 \text{ fF}/\mu\text{m} \end{aligned}$$

If the wires are widely spaced, we can use the following formulation:

$$\begin{aligned} C_{\text{int}} &= 2C_{\text{area}} + 2C_{\text{fringe}} + 2C_{\text{lateral}} \\ &\cong 2(0.015) + 2(0.025) + 2(0.0) \text{ fF}/\mu\text{m} \\ &= 0.08 \text{ fF}/\mu\text{m} \approx 0.1 \text{ fF}/\mu\text{m} \end{aligned}$$

To summarize, we have considered the 3D capacitances of deep submicron interconnect and found that upper layers of metal typically exhibit total capacitance of about $0.2 \text{ fF}/\mu\text{m}$ for closely spaced lines and $0.1 \text{ fF}/\mu\text{m}$ for widely spaced lines. The

capacitance of wires that are spaced between the two extremes would lie between $0.1 \text{ fF}/\mu\text{m}$ and $0.2 \text{ fF}/\mu\text{m}$ but the calculations are much more complex. However, capacitance extraction tools are available to obtain these components accurately. The lower levels of metal exhibit wire capacitances of about $0.1 \text{ fF}/\mu\text{m}$ in both situations. Individual values of the coupling capacitances may be estimated using Equations (10.11), (10.12), and (10.13). These first-order equations provide valuable insight into the origin and nature of the 3D effects. Worst-case capacitances lie in the range of $0.2 - 0.3 \text{ fF}/\mu\text{m}$. We now consider the effect of coupling capacitance on delay and noise.

10.4.2 Coupling Effects on Delay

The RC equations given in Section 10.3 using the Elmore delay did not properly take the coupling effects into account. A more realistic situation for delay calculation is shown in Figure 10.17 where neighboring lines are coupled together. The grounded capacitances shown include self-capacitance, fanout capacitance, and wire capacitance (to substrate). The coupling capacitances due to area, lateral, and fringing effects are actually distributed along the nets. The fact that the coupling capacitors exist between all neighboring lines implies that delay is a function of not only what is happening on the line of interest, but also what is happening on the neighboring lines.

To understand this effect, consider the case of two adjacent buffers driving their respective lines with a single coupling capacitance, as shown in Figure 10.18. To simplify the problem, we ignore the wire resistance and concentrate on the effects of coupling capacitance on delay. The line of interest is called the *victim* net while the neighboring line is called the *aggressor* for reasons that will become clear in the descriptions that follow. Consider the delay associated with the switching of net 2. It is coupled to net 1 with a large capacitor C_c . How should the coupling capacitance be handled? One is tempted to connect the other side of the capacitance to ground while computing the delay for net 2. However, this may not be correct depending on what is happening at net 1. The actual delay depends on whether net

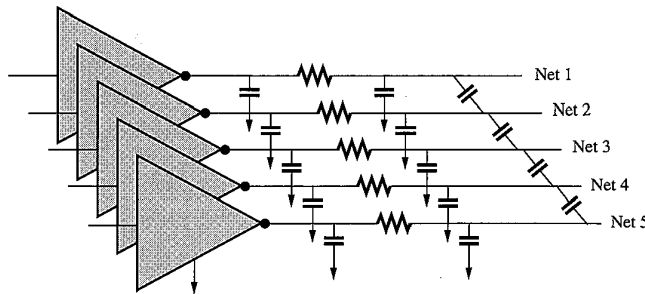
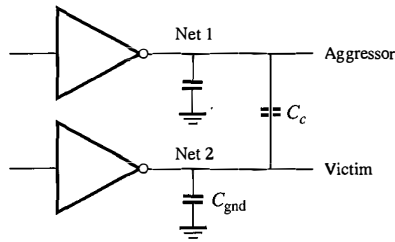


Figure 10.17
Coupling capacitance between lines.

**Figure 10.18**

Circuit for analysis of coupling delay and noise effects.

1 is switching from high to low or low to high, or perhaps not switching at all. If it is not switching at all, then it is appropriate to ground the other side of the capacitance. This would make the total capacitance at net 2 equal to $C_{\text{gnd}} + C_c$. That covers only one of the three possible cases that may occur.

To complete the story, we examine the total loading capacitance, C_L , of net 2 for a number of different cases. Again, if the neighboring net is stationary, then

$$C_L = C_{\text{gnd}} + C_c \quad \text{aggressor not switching} \quad (10.14)$$

since the adjacent net is not switching and the coupling capacitor behaves like a grounded capacitor. If the two nets are switching in the same direction, then

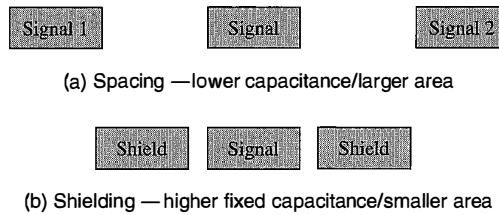
$$C_L = C_{\text{gnd}} \quad \text{both switching together} \quad (10.15)$$

since net 2 does not have to charge the coupling capacitance. However, if net 1 is switching in the opposite direction to net 2, then we should include the so-called Miller effect. Since net 2 must supply twice the charge on the coupling capacitor to compensate for the switching process:

$$C_L = C_{\text{gnd}} + 2C_c \quad \text{both switching opposite} \quad (10.16)$$

Even more combinations are possible if there are more neighboring nets. For example, in Figure 10.17, net 3 is coupled to net 2 and net 4. Obviously, there are many combinations of switching that result in different values of its C_L . In this example, the proper value of C_L must be determined after considering 3^2 possible cases. For n coupled nets, there are 3^n possible combinations. The result is that the delay for a net is unpredictable as it may vary from one switching event to another. This is one of the reasons why timing is so difficult to characterize accurately in deep submicron technologies.

A number of layout-based approaches have been developed to cope with this unpredictable delay, as shown in Figure 10.19. One approach is to *space out* the wires so that the lateral capacitance term decreases. Equation (10.12) tells us that

**Figure 10.19**

Spacing versus shielding to reduce coupling effects.

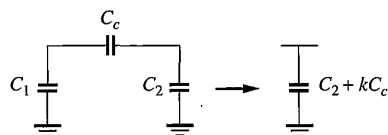
there is a $1/S$ relation between lateral capacitance and spacing, so we can reduce the amount of coupling by *spacing* signals apart from one another as in Figure 10.19a. Of course this will cost area, but it is effective when applied to a subset of the signals. Another approach is to interleave switching signals with nonswitching signals (such as power and ground) wherever possible. This is referred to as *shielding* as in Figure 10.19b. We can place well-behaved signals on either side of a signal so that its delay may be more predictable. However, in this case the total capacitance increases since there will always be a fixed capacitance to ground due to the shields.

Process engineers are also working to reduce the effects of coupling. Typically SiO_2 is used as the insulator in most CMOS processes. It has a relative dielectric constant of approximately 4. If the relative dielectric constant of the insulating material could be reduced from 4 to 2, the coupling capacitance would be cut in half. The pursuit of these so-called *low-k* dielectrics has been ongoing for years. Currently, the relative constant is roughly 3 but there are many issues to resolve to reach the target value of 2.

Timing verification tools can be used to obtain more accurate delay information since they have some notion of when signals switch. These tools require input from capacitance extraction tools that provide detailed coupling information. One way to characterize the effective load capacitance for these tools is to create a pre-multiplier for the coupling capacitance called a *k-factor*.⁴ Each coupling capacitor can be multiplied by this factor to adjust its value according to switching activity. This *k-factor* is nominally in the range 0 to 2.

To handle coupling effects, the timing verifier would ground all coupling capacitances after taking the *k-factors* into account. While this is a simplistic solution, it is one of the more practical approaches to coupling analysis. Figure 10.20 shows this process for a pair of adjacent nets. The load capacitance for net 2 is adjusted by the *k-factor* depending on the switching of net 1. One complication is that the *k-factors* may vary over time due to the switching conditions associated with adjacent nets. Fortunately the designer is primarily interested in the worst-case values of *k* since it

⁴ This is not the same as the *k* used in low-*k* dielectrics. This one is used to adjust capacitance values taking into account the switching behavior. To distinguish them, we will use the terms low-*k* and *k-factor* when referencing them.



$k = 0$ switching in same direction
 $k = 1$ aggressor not switching
 $k = 2$ switching in opposite directions

Figure 10.20

Handling coupling using k -factors.

will produce the worst-case delay values. This also simplifies the analysis significantly because it would be impractical to analyze the circuit exhaustively with case-dependent k -factors. For example, setup and hold time checks are important for the timing verification of flip-flops. When checking setup time, the designer uses the maximum k -factor on signal nets and the minimum on the clock net. The opposite settings are applied for hold time violations.

If this were not complicated enough, it turns out that the exact timing of the aggressor and victim signals also plays a role in the observed delay. Figure 10.21 illustrates the effect of signal timing in the presence of wire coupling in a SPICE transient analysis. The uncoupled waveforms of the victim (net 2) and aggressor nets (net 1) are shown along with a second plot of the victim waveform when many aggressors switch against it. Because of the early timing of the aggressors, the coupling actually causes capacitive feedthrough in the wrong direction before the proper transition begins. The resulting propagation delay in this example doubles

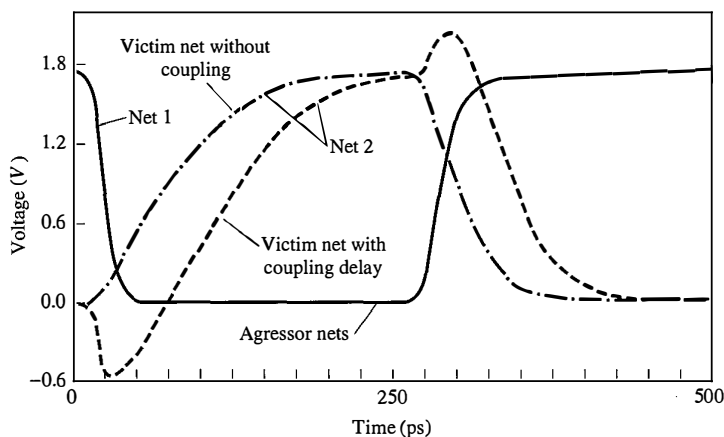


Figure 10.21

Transient analysis of coupled nets.

when compared to the case without any signal coupling. It is clear that the k -factor can lie outside of the range of 0 to 2 due to feedthrough.⁵ For the designer, the use of structured layout to avoid coupling wherever possible, and careful timing verification using k -factors are the two main approaches to cope with this signal integrity problem.

10.4.3 Capacitive Noise or Crosstalk

The second type of problem is noise injection due to capacitive coupling. This problem arises when there is a nonswitching net, again the victim net, surrounded by aggressor nets that are switching. When the aggressor nets switch, noise is injected into the victim line due to capacitive feedthrough effects. We have already observed this effect during a transition in Figure 10.21. Here, we consider the case where the victim line is quiet while aggressor lines are switching and injecting unwanted noise. Depending on the size and duration of the noise, it may act to upset the functionality of a design—that is, a “soft” error may occur. Coupling noise at latch or flip-flop inputs and in dynamic logic circuits is particularly troublesome. Specifically, incorrect values can be captured in flip-flops, or voltage levels at dynamic nodes may be upset due to noise injection. Noise may also propagate through logic gates, assuming that the noise level is high enough. This can cause problems for downstream logic circuits, particularly for dynamic logic. Tracking down this type of problem in a multimillion transistor chip has consumed many engineers for months, often without success. It can also increase the power consumption unnecessarily.

To understand coupling noise, consider Figure 10.22 which depicts our aggressor and victim nets connected only by a coupling capacitance, C_c . The feedthrough equation for the noise injected from net 1 into net 2 is

$$\Delta V_2 = \frac{C_c \Delta V_1}{(C_{gnd} + C_c)} \quad (10.17)$$

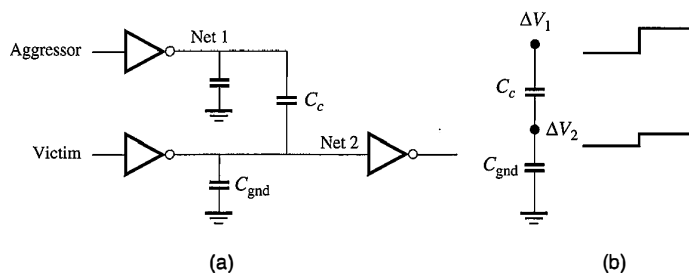


Figure 10.22
Coupling noise analysis.

⁵ The k -factor actually lies in the range of -1 to 3 when feedthrough is taken into account.

where ΔV_1 is the voltage transition at net 1 during a switching event. This equation was described in Chapter 7 on dynamic logic. If the change on the aggressor net is V_{DD} , then the injected noise is

$$\Delta V_2 = \frac{C_c V_{DD}}{(C_{\text{gnd}} + C_c)}$$

This first-order analysis is useful in obtaining an idea of the relative magnitude of noise due to coupling capacitance. To compute the noise level more accurately, a number of other factors must be taken into account such as the driver resistances, wire resistance, switching times of the aggressor signals, the switching direction of the signals, the slew rates of the signals, resistive shielding of driver, etc. Of course, once the noise is injected, the victim acts to restore the original level as quickly as possible. Complex situations that include these factors require detailed transient simulation in SPICE to obtain accurate results. Linear RC simulators are also useful for this purpose.

In practice, designers try to minimize coupling to neighboring signals to reduce noise. This can be accomplished by spacing the signals apart to reduce the C_c term, or by shielding signals with V_{DD} or Gnd, thereby eliminating the C_c term. These two approaches were illustrated in Figure 10.19.

To summarize, the delay and noise effects are difficult to analyze by hand, except for the smallest circuits. Designers must use structured design methodologies to avoid coupling effects as much as possible, especially when critical nets are involved. CAD tools must be used extensively to identify problem areas and apply the spacing and shielding approaches to resolve the problems. Process engineers must continue to develop *low-k* dielectric material to further reduce the effect of coupling capacitance. Ultimately, signal integrity-aware CAD tools, good design practices, and improved dielectrics are all needed to cope with these problems.

*10.5 Interconnect Inductance

A relatively new and even more complex issue of inductance is lurking around the corner as we scale technology and increase the clock frequency. Unfortunately, it may become a problem soon if not addressed properly from a design perspective. It is interesting to note the order in which we encountered DSM interconnect issues: first we experienced the increase in resistance; then, coupling capacitance; and now inductance. This order of appearance was quite fortunate since each one is progressively more difficult to understand, extract, or analyze than the previous one. Interconnect resistance is easy to understand and is confined to the metal conductor experiencing the current flow. Coupling capacitance is an electrostatic phenomenon between wires in a localized region around the wire of interest. It is clearly more complex to analyze than resistance, but still a tractable problem. Inductance, on the other hand, is an electromagnetic phenomenon and its effect is not necessarily confined to a localized region. It is more complicated to understand and almost impossible to extract and analyze, except for small sections. Nevertheless, we must understand its effects and explore options to avoid its impact.

The first step is to review inductance from an electromagnetic point of view. When current flows through a conductor, it sets up an electromagnetic field around it with a flux that stores energy. This flux is linearly related to the current, as follows:

$$\phi = Li \quad (10.18)$$

The proportionality constant is obviously the inductance, L , which is measured in *Henrys* (H). As the current changes, the magnetic flux acts on the wire itself and induces a voltage drop. For a linear inductor, we can take the time derivative of Equation (10.18) to obtain

$$\Delta V = L \frac{di}{dt} \quad (10.19)$$

This equation implies that the voltage drop is zero for a dc current. That is, the inductor can be treated as a short circuit for low-frequency operation. As the current demand of a chip increases, the voltage drop associated with the inductor increases. In high-speed designs, the rate of change of current with respect to time is increasing rapidly. Therefore, inductance effects are especially important at the package pins and in the power distribution system. We will examine this issue in more detail in Chapter 11.

Signal lines are also beginning to exhibit increased effects of inductance. It is important to realize that interconnect is really a distributed RLC line, not simply a distributed RC line. As such, it behaves like a transmission line under certain circumstances. We now elaborate these circumstances and describe how the inductance effects modify the delay equation. We will use R_{int} , C_{int} , and L_{int} to represent the resistance, capacitance, and inductance per unit length, respectively. An estimate of the inductance per unit length can be obtained from a well-known relationship in electromagnetic theory

$$C_{int}L_{int} = \epsilon_{ox}\mu_{ox} \quad (10.20)$$

where the insulating material is assumed to be oxide with permittivity, ϵ_{ox} , and permeability, μ_{ox} . This equation only applies to a conductor that is completely surrounded by a uniform dielectric, but it is useful in obtaining first-order inductance values. Assuming that $C_{int} = 0.1 \text{ fF}/\mu\text{m}$, $\epsilon_{ox} = (4)(8.85 \times 10^{-14} \text{ F/cm})$ and $\mu_{ox} = 12.6 \times 10^{-7} \text{ Wb/amp}$, then $L_{int} = 0.45 \text{ pH}/\mu\text{m}$. This is a very crude estimate but does provide the correct order of magnitude for inductance.

In Figure 10.23, a lumped model of the RLC line is shown. We can analyze this simple circuit to develop some intuition before proceeding on to the distributed line. First, consider the complex impedance associated with the resistance and inductance of the signal line

$$Z_{RL} = R + j\omega L \quad (10.21)$$

where $\omega = 2\pi f$ is the signal frequency in radians/second. If $R \gg j\omega L$, then Equation (10.21) tells us that inductance is not important. If $R < j\omega L$, then inductance

**Figure 10.23**Lumped model of RLC interconnect.

effects will be observed in the signal line. In the past, the resistance was larger than the reactive component so that RC effects were dominant. In recent years ω has been increasing significantly in very high-speed designs to the point where the reactance term is important. At the same time, process engineers and designers have been working to reduce wire resistance. First, the resistance of the metal lines has been reduced with the introduction of copper as a replacement for aluminum. Second, certain signals use very wide wires to lower the resistance significantly. This reduction in resistance combined with an increase in the operating frequency has made the reactive term non-negligible in certain signals. Interestingly, while inductance values have not increased over the years, the increase in frequency and decrease in resistance have combined to raise concerns about inductance.

The above analysis is somewhat localized to the interconnect itself. When considering the significance of inductance, it is important to consider both wire resistance and the on-resistance of the buffer driving the wire. In most cases, the buffer resistance is very large (on the order of kilo-ohms) and will dominate. That is,

$$R_{\text{buffer}} + R_{\text{wire}} \gg j\omega L$$

Therefore, we should be able to ignore self-inductance for most signals on the chip. Of course, if the buffer is large and the wire is wide, we would have

$$R_{\text{buffer}} + R_{\text{wire}} < j\omega L \quad (10.22)$$

and inductance would be noticeable. This is most prominent in clock lines. Therefore, the delay should be computed using RLC delay equations.

Next, we include capacitance in the analysis. The lossless transmission line of Figure 10.24 has inductance and capacitance but no resistance. It is driven by a voltage source V_s with a source resistance of R_s . Since the interconnect has no resistance, it can be represented by a characteristic impedance of Z_0 given by

$$Z_0 = \sqrt{\frac{L_{\text{int}}}{C_{\text{int}}}}$$

When we apply a step input to the interconnect, the response depends on the relative values of R_s and Z_0 . When the resistance is much greater than the characteristic impedance, i.e., $R_s \gg Z_0$, the RC effect will dominate the transient response. We expect an exponential response given by

$$V_{\text{out}}(t) = 1 - e^{-t/RC}$$

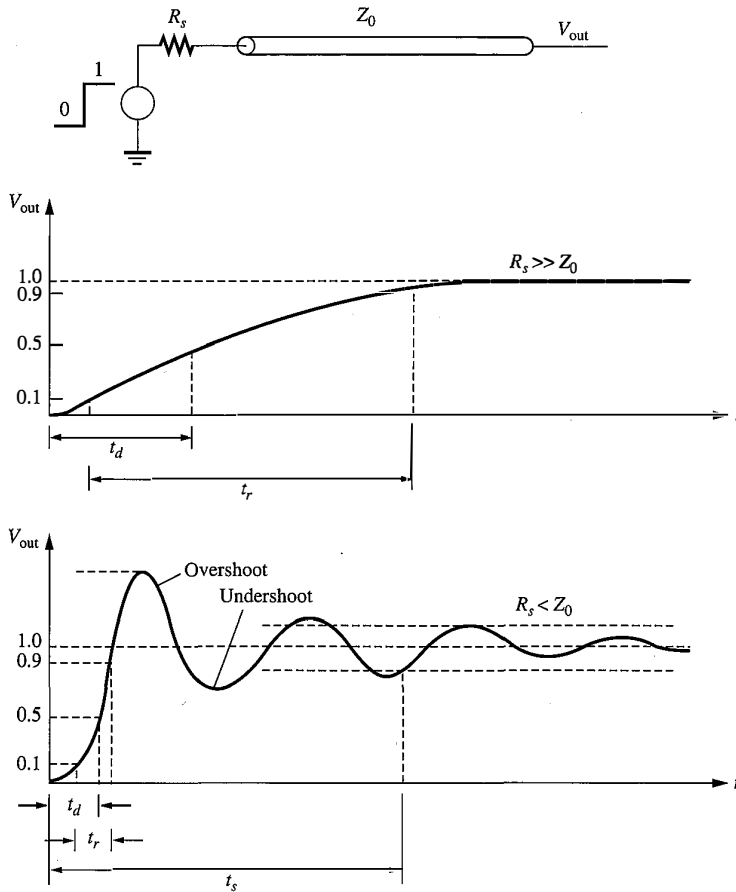


Figure 10.24
Responses of RLC interconnect.

As shown in the figure, the propagation delay, t_p , and rise time, t_r , are rather large.

However, if $R_s < Z_0$, inductance effects will be important and we obtain an oscillatory response within an exponential envelope. The response has a general form given by

$$V_{\text{out}}(t) = 1 - re^{-t/a} \cos(bt + c)$$

where r , a , b , and c depend on the RLC values. The figure illustrates the nature of the response. Note that t_d and t_r are much shorter in this case, but the tradeoff is a relatively large overshoot and undershoot of the waveform, and a longer settling time, t_s . These delay values can be computed from the response equation above.

Finally, we consider a lossy transmission line where distributed effects are taken into account. For this case, the length, l , of a distributed RLC line plays a role in whether it exhibits the effects of inductance. In particular, as a signal propagates down the wire, it experiences a certain degree of attenuation due to wire resistance. When it reaches the end of the wire, it reflects off the load impedance and propagates back toward the source. This reflected signal is also attenuated as it travels back down the wire. The interaction of the forward and backward propagating signals leads to the ringing effects observed in Figure 10.24. If the wire is very long, then inductance effects will not be observed due to signal attenuation. A standard derivation can be used to find the maximum length at which inductance effects occur

$$l_{\max} < \frac{2}{R_{\text{int}}} \sqrt{\frac{L_{\text{int}}}{C_{\text{int}}}} \quad (10.23)$$

Qualitatively, this tells us that as the resistance per unit length increases, the length over which inductance is important decreases.

The transition time of the signal at the output of the buffer also controls the transmission line effect. If the round-trip of the signal down the line and back to the buffer is short relative to the rise time, it does not have a significant transmission line effect. The round-trip time is given by

$$t_{\text{round-trip}} = 2T_0 = 2l\sqrt{L_{\text{int}}C_{\text{int}}} \quad (10.24)$$

The quantity T_0 is referred to as the time-of-flight. If the rise time (or fall time) is shorter than twice the time-of-flight, we will observe the transmission line effects. Therefore, we require that

$$t_r < 2l\sqrt{L_{\text{int}}C_{\text{int}}} \quad (10.25)$$

In terms of l , this translates to a minimum length of

$$l_{\min} > \frac{t_r}{2\sqrt{L_{\text{int}}C_{\text{int}}}} \quad (10.26)$$

Combining the results of Equations (10.23) and (10.26), we find that

$$\frac{t_r}{2\sqrt{L_{\text{int}}C_{\text{int}}}} < l < \frac{2}{R_{\text{int}}} \sqrt{\frac{L_{\text{int}}}{C_{\text{int}}}} \quad (10.27)$$

This result tells us that the wire must be larger than the length specified by $t_r/2\sqrt{L_{\text{int}}C_{\text{int}}}$ but smaller than the value specified by $(2/R_{\text{int}})\sqrt{L_{\text{int}}/C_{\text{int}}}$ for inductance to be important in delay calculation. We should view these results as being more qualitative than quantitative since the terms in the equations are difficult to compute with great precision. Furthermore, the effect of buffer resistance must also be considered to obtain accurate results.

What makes the inductance problem particularly difficult is that, fundamentally, inductance can only be defined for a closed current loop, not simply for a seg-

ment of wire. The inductance of the loop is proportional to the area of the loop. Knowing the forward path of the current is not sufficient; the *return* path must also be identified. This is not easy to determine for most signals. In the past, the return path was assumed to be in the substrate. For high-speed designs, the return path often occurs in the power and ground distribution systems, and even in other signal lines. Worse yet, there may be multiple return paths, some through power/ground rails, and others through signal lines that are not close to the signal of interest. The extraction of inductance values is difficult since the effect of inductance is not confined to a localized region around the conductor.

More problematic is *mutual inductance* since it causes noise and delay effects on unsuspecting neighboring lines in unexpected ways. This type of problem is a growing concern for most integrated circuit designers, as it is difficult to extract and analyze. Mutual inductance refers to time-varying current in one line giving rise to a voltage drop in a second line. Two *RLC* lines are coupled via mutual inductance, as follows

$$\begin{aligned}v_1 &= L_1 \frac{di_1}{dt} + M \frac{di_2}{dt} \\v_2 &= M \frac{di_1}{dt} + L_2 \frac{di_2}{dt}\end{aligned}\tag{10.28}$$

where L_1 and L_2 are the self-inductances of the two wires and M is the mutual inductance between the two wires. When a current flows in one line, it induces a return current in the other line due to its mutual inductance. The voltage drop on each line is the sum of the drop due to its self-inductance and its mutual inductance with the other line. This effect can occur over distances of 100 μm or more and create functional failures that are intermittent in nature. The details of mutual inductance effects, while important in future technologies, are beyond the scope of this book. The interested reader is encouraged to consult the references at the end of this chapter for further information.

To summarize, self and mutual inductance effects have been observed most frequently in clocks and busses. Clocks are prone to self-inductance effects whereas busses are more likely to experience mutual inductive effects. To limit the effects of self-inductance, wire lengths can be kept within the boundaries suggested by Equation (10.27). For mutual inductance, shielding techniques of the form shown in Figure 10.19 can be used to limit its effects.

*10.6 Antenna Effects

Another important issue associated with interconnect is the *antenna effect*. This is a reliability issue that may destroy the thin-oxide of transistors during fabrication or modify the threshold voltage of devices. During the fabrication process, charge builds up on each metal layer during plasma etching, sputtering, or chemical-mechanical polishing (CMP). This phenomenon is referred to as an antenna effect since charge is attracted to each metal layer like an antenna. Since $Q = CV$, the accumulated charge

creates a potential difference across the gate oxide of a MOS transistor. When too much charge accumulates on the gate, the voltage drop across the oxide damages the transistor by shorting the gate to the substrate. A less dramatic, but still undesirable, effect is a shift in the V_T due to injection of mobile carriers into the oxide through Fowler-Nordheim tunneling.

The antenna effect manifests itself as an interconnect design issue since the metal layer selection and routing determine the impact of this phenomenon. When Metal 1 is processed, its charge buildup is rather small since the lengths of the wires associated with Metal 1 tend to be short. As we add more and more layers, the area tends to increase since the wires are longer and thicker. Higher levels of metal have more surface area and therefore accumulate more charge, so they tend to be more troublesome with respect to the antenna effect. If a transistor survives the process on a given layer, that layer does not play as significant a role in the antenna effect during subsequent processing operations. Therefore, we usually assess antenna problems for each metal segment in each layer separately.

The *antenna rules* specify maximum allowable ratios of metal (or poly) area to gate area for each layer.

$$\text{Antenna Ratio} = \frac{\text{Area}_{\text{metal}}}{\text{Area}_{\text{gate}}} \quad (10.29)$$

For example, if the antenna ratio is set to 100, the area of a metal line would have to be 100 times larger than the gate area for an antenna rule violation to occur. There is a different ratio for each metal line. The design-rule checking (DRC) tools can check to see if each wire segment on each layer causes a violation, and flag those that are potential problems. For threshold voltage shifting, all metal layers connected to a gate need to be combined to obtain $\text{Area}_{\text{metal}}$ in Equation (10.29).

An example of the antenna effect is shown in Figure 10.25. Relative to the indicated device, poly, Metal 1, or Metal 2 may act to damage the gate region. For example, when Metal 1 is deposited and polished, charge develops on it and builds up a potential difference that could damage the oxide of the indicated transistor. If it tolerates the voltage level, then the corresponding Metal 1 segment is not considered to be a problem in the rest of the processing sequence. However, when the next layer of metal is applied, the corresponding charge on this second layer alone is checked against its antenna ratio. A more conservative approach recognizes that process induced damage is a cumulative effect. In that case, the total area of poly + Metal 1 + Metal 2 is divided by the gate area for the antenna check.

A number of different approaches have been used to avoid antenna effects. First, the insertion of diodes along the wire can be used to discharge the metal lines during the processing sequence. This approach is illustrated in Figure 10.26 where a *pn* junction has been connected to Metal 1. The job of this diode is to discharge each layer of metal as it is deposited so that the transistor does not see any significant voltage drop. It is reverse-biased and harmlessly removes charge as it accumulates.

The use of antenna diodes is very effective but presents some practical problems. Diodes cost area and their location must be decided based on the metal structure. Ideally, the diode should be near the transistor gate as shown in Figure 10.27. There is already a reverse-biased junction at the output of the inverter, but the

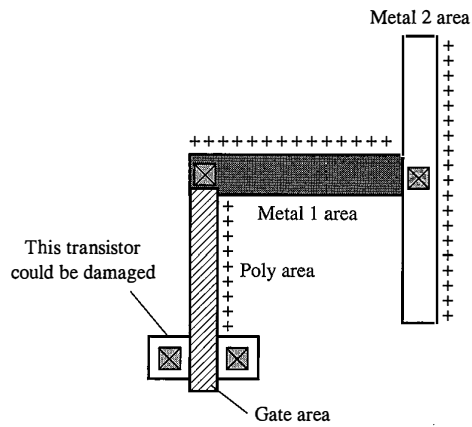


Figure 10.25
Antenna effects during IC processing.

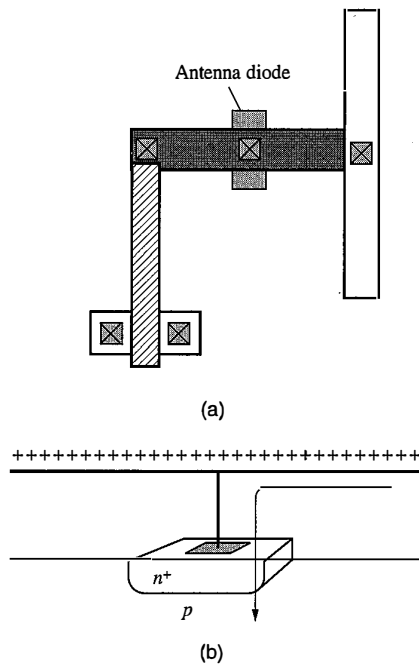
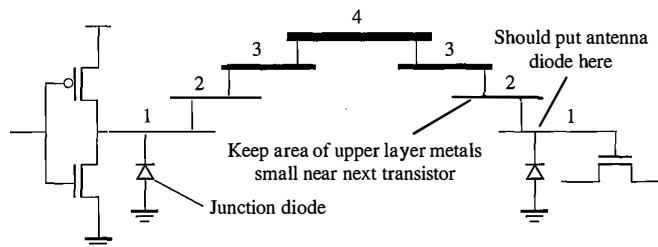


Figure 10.26
Antenna diodes to discharge metal lines.

**Figure 10.27**

Proper placement of antenna diodes.

antenna diode is needed near the transistor gate as shown. Note that as each layer in the example is deposited, the far-end transistor must be able to handle all the charge accumulation on each successive layer in this configuration. Therefore, the far-end metal lines must be kept short or we must place diodes near the transistor. However, it would be too expensive in terms of area to place diodes near all transistor gate inputs, or embed the antenna diodes inside cell libraries. It should only be used where absolutely necessary. This involves a complete DRC verification of the entire layout to identify the locations that are appropriate. The second issue is to define the area of the diode. This depends on the area of the metal segments that it supports. If large diodes are used unnecessarily, area may be compromised.

A second approach is to break up long wires with buffer insertion. This will reduce the total area available for charge accumulation and increase the transistor gate area, thereby reducing the total charge on each metal line. Of course, such a process is already carried out as part of reducing the RC delay, but we obtain a side benefit of a reduction of antenna effects. We must still verify that no segment creates a problem, but we will definitely reduce the number of violations after buffer insertion.

A third approach is to create metal jumpers from one level to the next to break up the lines near the far-end of a wire. This is illustrated in Figure 10.28. In this case, we show a Metal 1 wire that violates the antenna ratio. A seemingly unnecessary jog to Metal 2 is introduced near the gate of the transistor to avoid the total charge of Metal 1. Recall that, when the processing steps are applied to introduce the vias, the charge of Metal 1 is removed. Therefore, the transistor will only experience the small charge from the Metal 2 segment. The additional resistance and capacitance of the vias and Metal 2 segment are usually negligible.

In summary, the proper solution to the antenna problem depends on the physical design methodology, tools, and engineering tradeoffs associated with the design. All of the above solutions are viable if carried out with careful considerations of the tradeoffs. In advanced processes, diode and buffer insertion appear to be most helpful. Hopefully, tools of the future will automatically take care of this issue on behalf of the designer.

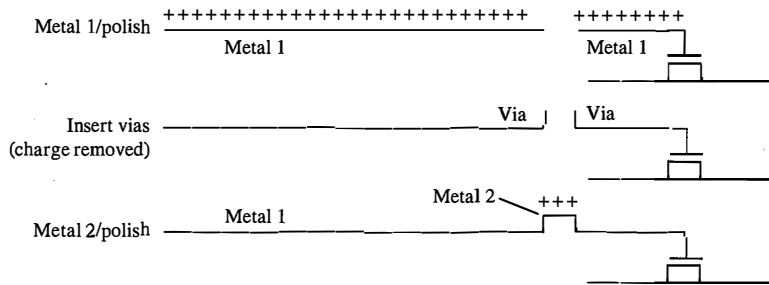


Figure 10.28
Metal line jogs to avoid antenna effects.

10.7 Summary

Interconnect data:

- (a) 0.18 μm 5-layer aluminum process: $\rho = 2.7 \mu\Omega\text{-cm}$; Metals 1–4, $T = 0.5 \mu\text{m}$, $H = 0.5 \mu\text{m}$, $W_{\min} = 0.3 \mu\text{m}$; and Metal 5, $T = 1 \mu\text{m}$, $H = 0.5 \mu\text{m}$, $W_{\min} = 0.4 \mu\text{m}$
- (b) 0.13 μm , 8-layer copper process: $\rho = 1.7 \mu\Omega\text{-cm}$; Metals 1–6, $T = 0.4 \mu\text{m}$, $H = 0.5 \mu\text{m}$, $W_{\min} = 0.2 \mu\text{m}$; and Metals 7–8, $T = 0.8 \mu\text{m}$, $H = 0.5 \mu\text{m}$, $W_{\min} = 0.4 \mu\text{m}$

Wire resistance calculations:

$$R = \frac{\rho L}{A} = \frac{\rho L}{TW} \quad R_{sq} = \frac{\rho}{T} \quad R = R_{sq} \left(\frac{L}{W} \right)$$

$$R_{sq} = \frac{\rho}{T} = \frac{2.7 \mu\Omega\text{-cm}}{0.5 \mu\text{m}} = 54 \text{ m}\Omega/\square \quad \text{Al Metals 1–4}$$

$$R_{sq} = \frac{\rho}{T} = \frac{2.7 \mu\Omega\text{-cm}}{1.0 \mu\text{m}} = 27 \text{ m}\Omega/\square \quad \text{Al Metal 5}$$

$$R_{sq} = \frac{\rho}{T} = \frac{1.7 \mu\Omega\text{-cm}}{0.4 \mu\text{m}} = 42 \text{ m}\Omega/\square \quad \text{Cu Metals 1–6}$$

$$R_{sq} = \frac{\rho}{T} = \frac{1.7 \mu\Omega\text{-cm}}{0.8 \mu\text{m}} = 21 \text{ m}\Omega/\square \quad \text{Cu Metals 7–8}$$

Elmore delay calculation:

$$\tau_i = \sum_k (C_k \times R_{ik})$$

Optimal buffer insertion:

$$\text{Number of stages } N = \sqrt{\frac{R_{\text{int}} C_{\text{int}} L^2 / 2}{R_{\text{eqn}} (C_J + C_G) (1 + \beta)}}$$

$$\text{Size of buffers } M = \sqrt{\frac{R_{\text{eqn}} C_{\text{int}}}{C_G (1 + \beta) R_{\text{int}}}}$$

Coupling capacitance:

$$C_{\text{area}} = \epsilon_{\text{ox}} \frac{W}{H}$$

$$C_{\text{lateral}} = \epsilon_{\text{ox}} \frac{T}{S}$$

$$C_{\text{fringe}} = \epsilon_{\text{ox}} \ln \left(1 + \frac{T}{H} \right)$$

Coupling capacitances in delay calculation:

$$C_L = C_{\text{gnd}} + C_c \quad \text{aggressor not switching}$$

$$C_L = C_{\text{gnd}} \quad \text{both switching together}$$

$$C_L = C_{\text{gnd}} + 2C_c \quad \text{both switching opposite}$$

Crosstalk noise:

$$\Delta V_2 = \frac{C_c \Delta V_1}{(C_{\text{gnd}} + C_c)}$$

Self-inductance:

$$\Delta V = L \frac{di}{dt}$$

Mutual inductance:

$$v_1 = L_1 \frac{di_1}{dt} + M \frac{di_2}{dt}$$

$$v_2 = M \frac{di_1}{dt} + L_2 \frac{di_2}{dt}$$

Conditions where inductance should be considered in delay calculation:

$$R_{\text{buffer}} + R_{\text{wire}} < j\omega L$$

$$\frac{t_r}{2\sqrt{LC}} < l < \frac{2}{R}\sqrt{\frac{L}{C}}$$

Antenna rule:

$$\text{Antenna Ratio} = \frac{\text{Area}_{\text{metal}}}{\text{Area}_{\text{gate}}}$$

REFERENCES

1. M. Celik, L. Pileggi, and A. Odabasioglu, *IC Interconnect Analysis*, Kluwer Academic Publishers, Boston, MA, 2002.
2. Y. Ismail and E. Friedman, *On-chip Inductance in High-Speed Integrated Circuits*, Kluwer Academic Publishers, Boston, MA, 2001.
3. H. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley Publishing, Reading, MA, 1990.
4. N. Arora, K. V. Raol, R. Schumann, and L. Richardson, "Modeling and Extraction of Interconnect Capacitances," *IEEE Transactions on Computer-Aided Design*, vol. 15, no. 1, pp. 58–67, Jan. 1996.

PROBLEMS

- P10.1.** Compare the resistance of a 20 mm Metal 5 wire in 0.18 μm technology with a 20 mm Metal 8 wire in 0.13 μm technology (a non-scaled wire). Use the information provided in the Summary section. Compute the delay for each wire assuming an ideal source and a wire capacitance of $C_{\text{int}} = 0.1 \text{ fF}/\mu\text{m}$.
- P10.2.** Compare the resistance of a 20 mm Metal 5 wire in 0.18 μm technology with a 14 mm Metal 8 wire in 0.13 μm technology (a scaled wire). Use the information provided in the Summary section. Compute the delay for each wire assuming that the wire capacitance is $C_{\text{int}} = 0.1 \text{ fF}/\mu\text{m}$.
- P10.3.** Two adjacent Metal 5 wires in 0.18 μm technology are spaced out from 4λ to 40λ in 4λ increments. Plot the capacitance per unit length of these wires as a function of spacing, assuming that they are shielded above and below. Use minimum width dimensions for the wire.
- P10.4.** Simulate a distributed RC line as an RC ladder circuit in SPICE with 1 stage, 3 stages (as shown in Figure P10.4), and 20 stages. You can choose your own sizes but ensure that the total resistance and total capacitance are the same in all three cases. Measure the propagation delay for the three cases when a step input is applied at the source. Derive a delay formula of the form:

$$\tau = \alpha R_{\text{wire}} C_{\text{wire}}$$

and compute α for the three cases. Comment on the results.

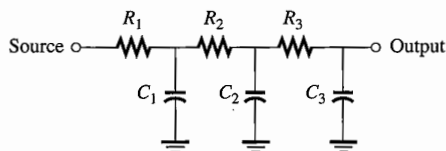


Figure P10.4

P10.5. Repeat the same experiment as described in Problem P10.4 but this time use each of the following three models shown in Figure P10.5 in place of the RC ladder. Which model produces accurate results?

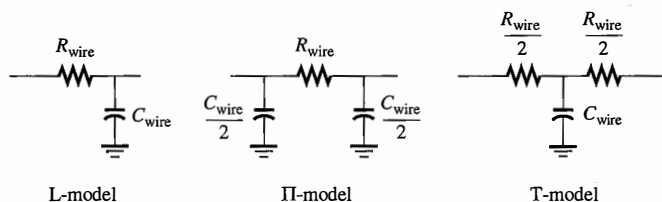


Figure P10.5

P10.6. Simulate the RC circuit shown in Figure P10.6 using SPICE. Set $R_1 = R_2 = R_3$ and $C_1 = C_2 = C_3$. Compare the Elmore delay with the delay derived from SPICE. How accurate are the results for each node using Elmore? Explain why any particular result is not accurate.

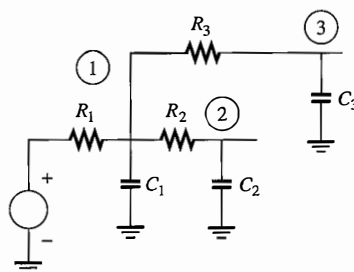


Figure P10.6

P10.7. Wires can be placed into short, medium, long, and very long categories. Establish rules of thumb for a Metal 5 wire in $0.18\ \mu\text{m}$ technology that quantify each of these categories based on the following definitions:

- (a) Short wires do not require any resistance or capacitance calculations if they contribute less than 5% to the delay. What is the maximum short wire length?
- (b) Medium wires must include capacitance in the delay calculation, but do not require any resistance information. What is the maximum medium wire length?
- (c) Long wires require both resistance and capacitance information but do not require buffer insertion. What length of wire can remain unbuffered?
- (d) Very long wires require buffer insertion.

P10.8. Figure P10.8 shows a circuit that has two interconnect lines with a large coupling capacitance between them.

- (a) Estimate the peak value of noise on the victim line when the aggressor line switches by 1.8V.
- (b) What is the effective capacitance loading on the victim for delay calculation if the aggressor switches in the opposite direction of the victim?

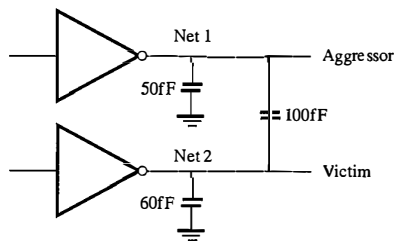


Figure P10.8

- P10.9.** For a $0.13\ \mu\text{m}$ technology, estimate the range over which inductance is important assuming a Metal 8 wire. You can assume typical numbers in the given technology. How does this range compare against the critical length in the same technology? (See Section 10.2.3.)
- P10.10.** The amount of noise injected on a wire depends on the strength of the driver, rise time of the aggressor, as well as the percent of the total capacitance that is due to coupling. Create a simple RC model based on Figure P10.8 that you can use to help you estimate the size of noise caused by the coupling.
- P10.11.** Compare the number of repeaters required in a $0.18\ \mu\text{m}$ technology versus a $0.13\ \mu\text{m}$ technology for a 10 mm wire. Use the top layer of metal and the standard technology parameters to compute the numbers.

P10.12. Consider an 18 mm Metal 7 wire in a $0.13\text{ }\mu\text{m}$ technology. Assume the wire is $0.4\text{ }\mu\text{m}$ wide and $0.8\text{ }\mu\text{m}$ thick with a spacing to adjacent wires of $2\text{ }\mu\text{m}$. The height above and below to Metal 8 and Metal 6 is $0.5\text{ }\mu\text{m}$. Assume Cu for interconnect ($\rho = 0.017\text{ }\Omega\text{-}\mu\text{m}$) and a low- k dielectric material ($\epsilon_r = 3.0$). Assume a worst-case coverage of Metal 6 below, Metal 8 above, and Metal 7 for adjacent lines that are shielding the wire of interest.

- (a) Compute the resistance per unit length and capacitance per unit length for this wire.
- (b) Estimate the delay of the distributed RC wire, assuming that the driver is a perfect voltage source and that the load capacitance is 50 fF .
- (c) Assume that the wire is being driven by a 25X inverter. What is the new delay?
- (d) Now, assume that the inductance is $0.5\text{ pH}/\mu\text{m}$. Will the inductance be a factor in the delay calculation for this buffer size? Compare hand analysis with results in SPICE.
- (e) Now remove the 25X buffer and perform buffer insertion on this wire using the standard formulas. Should inductance be a factor for the new wire lengths? Compare your hand analysis with results from SPICE.

Power Grid and Clock Design

CHAPTER OUTLINE

- 11.1 Introduction
- 11.2 Power Distribution Design
- 11.3 Clocking and Timing Issues
- *11.4 Phase-Locked Loops/Delay-Locked Loops
- References
- Problems

11.1 Introduction

In this chapter, we address two chip-level design challenges that are dominated by interconnect issues: power and clock distribution. Here, the word *power* is intended to mean the power and ground distribution systems, which include V_{DD} and Gnd (or V_{SS}). The power supply, up to this point in the book, has been assumed to be a constant value that is available everywhere on the chip. However, it must actually be routed from the power pads to the gates, memory circuits, and all other functions on the chip. Furthermore, its value fluctuates over time, depending on the switching activity on the chip. These variations in the voltage must be tolerated by the gates on the chip. The same applies to the clock signal. It has been assumed to be available at all required points on the chip with a well-defined clock period. However, it too must be routed from the clock source or root node to the appropriate destinations inside the chip. It also has some variability in its period and arrival times across the chip that must be tolerated by the logic circuits.

Power distribution and clock design are now complex tasks that must be done with great care in deep submicron technology. Gone are the days of simply laying out metal tracks for the power supply or running a clock signal to all the flip-flops in the design. Today, many issues come into play and affect the integrity of the power grid or the clock network. This chapter addresses many of these issues, although it is difficult to completely describe all of the issues and their interactions

in a single chapter. The chapter will highlight the interaction of the power system with the clock design. Many design groups work on power system and clock design together so that the complete problem is properly addressed. We will also address clock generation and phase-lock loop (PLL) circuits as part of this chapter.

11.2 Power Distribution Design

Power distribution, when considered in its entirety, actually involves on-chip and off-chip issues starting from off-chip dc-dc converters, printed circuit boards (PCBs), power planes, packages, sockets, power pins or solder bumps, and finally the connection to the gates. Proper power grid design requires interaction of system designers, thermal designers, system architects, board designers, and chip designers. The problem demands global optimization rather than localized chip-level optimization. This section will focus on the chip-level issues, but the reader should keep in mind that this is only one part of the complete design problem.

Much of the complexity of power distribution systems arises due to the large number of transistors on a chip, the *RLC* nature of interconnect, the frequency of operation, and the current demand of high-speed circuits today. The highest frequency of operation is well over 1 GHz, while power has exceeded 100 W and current levels have reached 100 A in state-of-the-art designs. Meanwhile, supply voltage scaling to reduce power has led to V_{DD} values of 1.2 V and below. Fluctuations in the power system at these levels lead to timing variations and, eventually, design failure. Allowable voltage noise in the supply is budgeted at $\pm 10\% \times (V_{DD} - \text{Gnd})$ as a rule of thumb.

In order to manage fluctuations, the total impedance seen from the V_{DD} pads to the gate connection must be controlled based on the needed current. The target impedance of the power grid as seen from the pads is given by

$$Z = \frac{V_{DD} \times (\text{fractional noise budget})}{I_{DD}} \quad (11.1)$$

For a supply voltage of $V_{DD} = 1.2$ V and a supply current of $I_{DD} = 100$ A, $Z = 1.2 \text{ V} \times (10\%) / 100 \text{ A} = 1.2 \text{ m}\Omega$. This is an incredibly small number to achieve over the gigahertz frequency range and makes power distribution design very challenging.

The noise on the supply is primarily due to *IR* drop and *Ldi/dt* as will be explained later. At this stage, it is sufficient to realize that voltage drops occur on the power grid due to the resistance and inductance along the current path. These voltage drops affect clock skew, gate performance, and clock and PLL timing jitter.

There are also issues of power grid electromigration that affect long-term reliability. The metal migrates due to the eroding effects of large currents and eventually breaks. Ultimately, the power system design will limit the system performance and may lead to failures in the field. Therefore, it must be designed carefully to avoid such potential problems that are collectively called *power integrity* issues.

One approach that has been used in the past is to overdesign the system to avoid any potential failures. *Overdesign* is a term used often by designers when design risks

are removed by going well beyond the needed safeguards and recommended guidelines. One example of this is the use of decoupling capacitance on the chip to control IR and Ldi/dt . In the case of overdesign, most of the available area would be committed to decoupling capacitance rather than to devices. However, there is a large area penalty associated with this overdesign (and an even larger one for underdesign—nonworking chips!) so it is important to understand the key issues and make the proper tradeoffs wherever possible. With this brief overview, we now describe the main issues in power system design and then examine a few basic structures of the power distribution system.

11.2.1 IR Drop and Ldi/dt

The conditions contributing to the complexity of power distribution systems are primarily due to interconnect and its impact on chip performance. IR drop due to resistance on V_{DD} lines impacts overall timing and functionality. These effects are made worse by the presence of Ldi/dt voltage variations at package pins due to the increased rate of change of current in high-speed designs. Together, the total voltage drop at any point in the power grid is given by

$$V = IR + L \frac{di}{dt} \quad (11.2)$$

We consider each of these effects separately starting with IR drop. The basic concept of IR drop is illustrated in Figure 11.1, which depicts two large buffers connected to a resistive power supply. Initially, all voltage levels in the power grid are at V_{DD} . As the large driver, $inv2$, begins to switch, the demand for current reduces the voltage in the power grid. Specifically, the wire resistance creates voltage drops that increase as the current moves from the external supply toward $inv2$. The voltage remains relatively high near the V_{DD} connection at the periphery of the chip, and drops by ΔV at the connection to $inv2$. In practice, IR drop is caused by simultaneous switching of clock buffers, bus drivers, memory decoder drivers, etc. These simultaneous

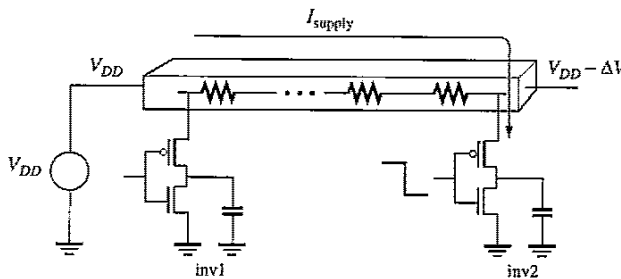


Figure 11.1
 IR drop in power distribution system.

switching events can occur anywhere on the chip and, therefore, all regions are susceptible to IR drop violations. The ground grid is subject to the same type of problem when the outputs switch low, except that the value will increase in voltage. This is sometimes referred to as *ground bounce*.

Voltage drops on the power grid primarily affect timing. The effect of IR drop on performance can be significant. IR drop reduces the drive capability of the gates and increases the overall delay. Typically, a 5% drop in supply voltage can affect delay by 10–15%, or more. Such an increase becomes serious when managing clock skews in the range of 100 picoseconds. In the previous chapter, we found that path delays are no longer predictable due to interconnect issues. This problem is exacerbated by IR drop. Ideally, timing calculations should account for worst-case IR drop to ensure a working design. The analysis can be carried out using process corner simulation, as described in Chapter 3.

IR drop also compromises the noise margins of logic gates, due not only to voltage drop in the power grid, but also to the increase in voltage in the ground grid. Once the noise margins drop below the budgeted amount, typically 10%, the design is not guaranteed to operate properly. Over the years, supply voltage has been shrinking to avoid transistor punch-through conditions, hot-electron effects, and device breakdown. More recently, V_{DD} scaling has been driven by the need to reduce power dissipation rather than the other issues. Either way, this has resulted in smaller and smaller noise margins. With IR drop, the margins are reduced even further which makes it more difficult to manage the effects of interconnect coupling noise in a multimillion-transistor design.

Now consider the Ldi/dt term of Equation (11.2). This is another source of voltage drop in the power supply due to package pin inductance—typically around 1–2 nH. The inductance arises from the bonding wire used to connect the chip I/O pads to the lead frame in a traditional dual-inline package (DIP), as in Figure 11.2a. While the inductance L has not changed significantly over the years, the value of di/dt has continued to increase and the supply voltage has been decreasing from 5 V to 3.3 V, 2.5 V, 1.8 V, and recently 1.2 V and below. These trends have reached a point where the Ldi/dt drop can contribute significantly to an overall voltage drop in the power grid, especially in peak demand situations.

Today, many companies have moved to a ceramic *ball-grid array* (BGA) packaging due to the high number of chip I/O and power and ground connections needed. In this packaging technology, solder balls or *bumps* replace the pins of the DIP as shown in Figure 11.2b. The bumps can be placed anywhere in the chip area and allow 300 to 500 I/O connections. This is a more expensive solution but must be used whenever the I/O count exceeds the capacity of the previous packaging technologies. The inductance of each solder bump is of the order of 0.1 nH.

Consider the example of Ldi/dt on the BGA package option shown in Figure 11.3. Here, the power supply is driving a few large inverters and we assume that the resistance of the line is zero. The inductances of interest are the ones due to the solder bump and via connecting from the bump and the package plane (providing clean power). A current level of 25 mA is supplied by V_{DD} and flows into the circuit over a 100 ps time interval. At this same time, some of the buffers are discharging

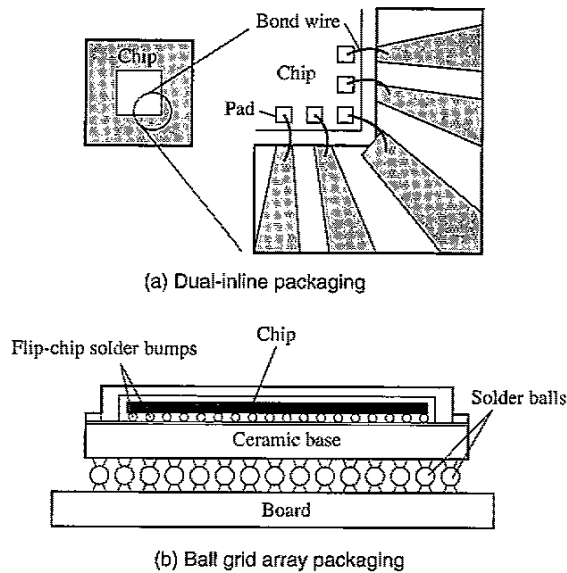


Figure 11.2
Chip packaging options.

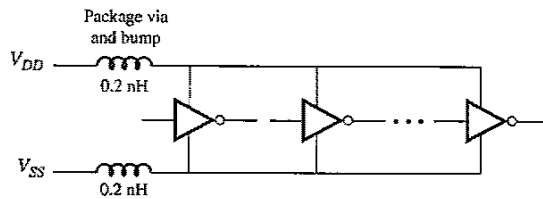
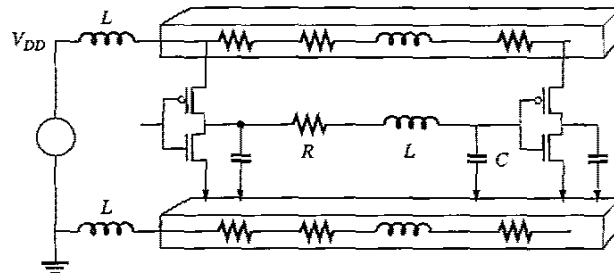


Figure 11.3
Effect of off-chip inductance.

through V_{SS} with roughly the same level of current in the same time period. If the bump and via generate 0.2 nH of inductance, then the total voltage drop due to the inductance on both rails is

$$V_L = 2 \times L \frac{di}{dt} = 2 \times 0.2 \text{ nH} \times \frac{25 \text{ mA}}{100 \text{ ps}} = 100 \text{ mV}$$

This is a significant drop considering that the supply voltage may only be 1.2 V. Furthermore, this is only one of the voltage drops to consider. If the inductance of the

**Figure 11.4**

Combined resistive and inductive effects of power/ground system.

power system is included, we will begin to observe $L di/dt$ effects due to the power grid itself, in addition to the pin inductance. We also neglected the resistance of the power grid, and the fact that many other buffers may be switching at the same time.

The overall voltage drop due to both $L di/dt$ and IR , which are both dynamic phenomena, requires the modeling approach shown in Figure 11.4. The packaging inductances, power grid inductances, and power grid resistances must all be considered together with an RLC model of interconnect. Of particular concern are situations that give rise to simultaneous switching noise (SSN) where many buffers switch together producing large IR drops and ground bounce.

11.2.2 Electromigration

The high currents experienced in the power grid also induce electromigration (EM) effects where metal lines begin to migrate and eventually break during the operation of the chip. The process involves the migration of metal molecules due to the high current densities and narrow line widths leading to a short or open in the metal line. Once this occurs, the chip may not function as expected or may no longer meet the timing specifications. Existing cracks or other imperfections in the metal lines are particularly prone to this problem since they are already weak links in the power system. Failures due to EM can be catastrophic because they occur in the field when the chip is in a system and in a customer's hands.

An example of an electromigration failure is shown in Figure 11.5. In the past, low current densities and wide metal lines, combined with special processing, reduced the likelihood of EM. As we exceeded speeds of 100 MHz, and reached geometries below $0.8 \mu\text{m}$, the potential for EM problems increased in aluminum. One of the reasons for switching to copper was its superior performance with respect to electromigration. Cu is approximately 10 times better in terms of EM than Al, but Cu diffuses quickly through oxide and it was not used in ICs because of this problem. However, advances in processing technology to solve this problem led to the eventual switch from Al to Cu. Initial EM results of Cu indicate that it is slightly better than Al, but large improvements are expected as the technology matures. Unfortunately, today the copper vias experience a greater degree of EM failures compared to the tungsten vias used with aluminum.

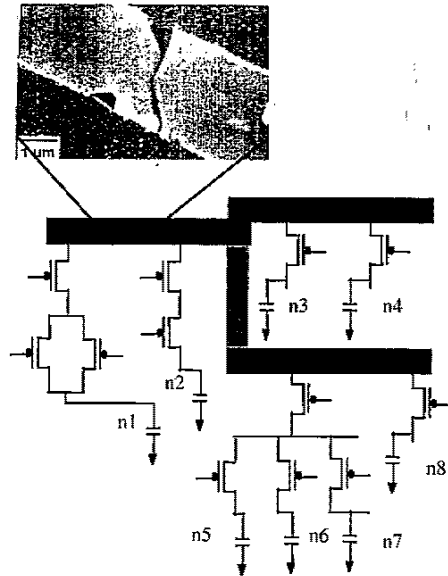


Figure 11.5
Electromigration failure in power bus.

Electromigration in the power grid is a function of the average current flow in metal lines and vias. Current flows primarily in one direction in each power bus, since it is delivering current from the power pad to the gate in the case of V_{DD} (see Figure 11.1). Therefore, it is called a dc phenomenon. To assess the EM tolerance of a piece of metal, we can examine its average dc current density, J , and compare it to some allowable level, $J_{\max}$, when EM failure occurs. Since metal lines vary in height and material properties at different levels in the design, each metal layer may have a different failure criteria.

It is useful to characterize the lifetime of a chip due to long-term reliability problems through a metric called the *mean time to failure (MTTF)*. If the value of MTTF is 5–10 times greater than the expected lifetime of the product in which the chip is embedded, then we can safely ship the design to customers. However, if the chip lifetime is less than the product lifetime, improvements to the design must be made. Black's law is used to estimate MTTF of a metal line using the average current density, J , and the activation energy¹ of the failure, ΔH :

$$\text{MTTF} = \frac{A}{J^2} \exp(\Delta H/kT) \quad (11.3)$$

¹ Electromigration is modeled as a diffusion process that is thermally activated at a given energy level. The activation energy depends on the diffusion mechanism, either through the lattice ($\Delta H = 1.4$ eV) or across the grain boundaries ($\Delta H = 0.6$ eV).

Here, MTTF is the median time to failure in an ensemble of samples and A is an empirical fitting coefficient. The MTTF is inversely proportional to J^2 and is very sensitive to temperature. For a target MTTF, the value of $J_{\max}$ can be determined from the equation in units of amperes/cm². A typical value of $J_{\max}$ is approximately 10⁶ A/cm², or 10 mA/μm². This value is compared with the actual current density in each segment to evaluate the potential for electromigration.

Example 11.1 Use of MTTF to Establish Maximum Allowable Current Density

Problem:

A design has a required lifetime of 10 years. Measurements for Al electromigration using an accelerated testing method indicates that $A = 2 \times 10^7$ hr-cm²/ampere and $\Delta H = 0.85$ eV. What is the maximum allowable current density in Al interconnect at 125 °C?

Solution:

The MTTF is the time at which 50% of the parts will fail, that is, t_{50} . For a desired lifetime of 10 years, the MTTF should be 5-10 times larger than this value. We will use a factor of 10 to be on the safe side:

$$t_{50} = \text{MTTF} = 10 \times 10 \text{ years} \times 365 \text{ days} \times 24 \text{ hours} = 8.8 \times 10^5 \text{ hours}$$

Using Equation (11.3), we can solve for $J_{\max}$:

$$\begin{aligned} J_{\max} &= \sqrt{\frac{A}{t_{50}} \exp\left(\frac{\Delta H}{kT}\right)} = \sqrt{\frac{2(10^7)}{8.8(10^5)} \exp\left(\frac{0.85 \text{ eV}}{8.62 \times 10^{-5} \text{ eV/}^\circ\text{K}(398^\circ\text{K})}\right)} \\ &= 8.95(10^5) \text{ A/cm}^2 \end{aligned}$$

This value can be used as the maximum tolerable current density for electromigration.

The current density in each wire segment is computed using the wire dimensions

$$J_{\text{avg}} = \frac{I_{\text{avg}}}{W \times T} \quad (11.4)$$

where W is the wire width and T is the wire thickness. The criterion for electromigration failure is

$$J_{\text{avg}} > J_{\max} \quad (11.5)$$

This comparison must be performed for every segment and via in the power grid to identify those nets that are prone to EM failure in the prescribed lifetime. Failing metal segments and vias must be modified, typically by adjusting the wire width according to Equation (11.4), until condition (11.5) is no longer violated. Not all

EM failures must be fixed since a break in the metal line does not necessarily mean that the part will fail. However, it may increase the likelihood of failure in other branches since there are usually many different current paths to each gate. The failures should be ordered in some manner, and then fixed in that order, until the probability of failure is below some predetermined threshold.

11.2.3 Power Routing Considerations

Integrated circuit power distribution systems are designed to provide the needed voltages and currents to the transistors that perform the logic functions of a chip. IC designers must design power systems with Equations (11.2) and (11.3) in mind for reliable operation. A complete picture of power grid integrity can only be obtained when effects such as IR drop, ground bounce, Ldi/dt , and electromigration are considered together. These are full-chip issues that must be addressed by interconnect verification tools that have the capacity and performance required to analyze detailed representations of the chip in a reasonable time. We will leave these higher-level issues to CAD tools and vendors. Instead we will focus on the methods needed to avoid these types of issues.

Initially, one can make estimates of the IR drop and Ldi/dt from R , C , and L calculations. Designers typically compute resistance along a conductor by counting the number of squares along the line and multiplying by the sheet resistivity. Similarly, the capacitance along the line is computed by multiplying the length of the line times the capacitance per unit length. Inductance can be estimated for the bonding wire or solder bumps, and crude estimates can be obtained for the power lines. Using this information and the expected current levels and rates of change, the anticipated voltage drops can be computed, along with electromigration lifetimes. Carrying out this type of analysis for a whole chip is difficult so rules of thumb have emerged to help guide the power grid design. These rules are associated with the metal width needed to keep the resistance low enough for the expected current levels and current densities.

A simplified method of power distribution is illustrated in Figure 11.6. The outer ring is V_{DD} and the inner ring is V_{SS} . The power and ground straps are routed

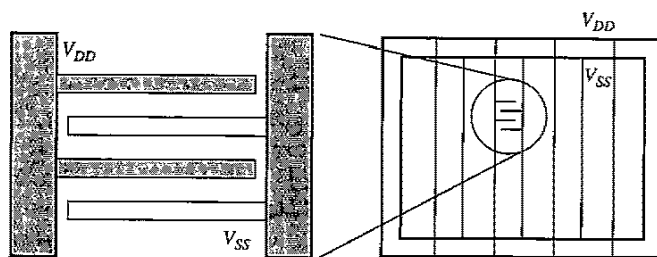


Figure 11.6
Large block level power distribution.

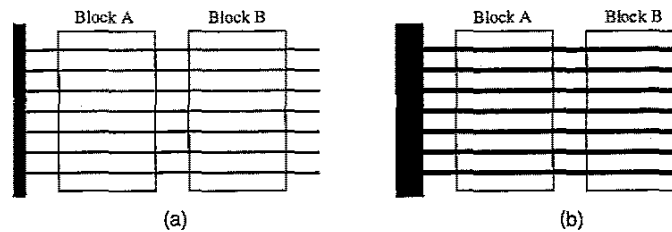


Figure 11.7
Power routing options.

vertically and interleaved so that at the lowest level they can be connected to the gates at the appropriate contact points. Of course, this type of routing does not fully account for the different types of blocks at the lower level, such as memory, logic, clock, and bus drivers, that may be connecting to the supply. However, the general idea of distributing power routes around the chip is conveyed in this diagram.

The power distribution system should be designed to route current to the gates, taking into account IR and EM . This requires a more careful consideration of the routing methodology. For example, consider the two blocks in Figure 11.7a. If power is routed through Block A to Block B, a larger IR drop will occur in Block B since power is also being consumed by Block A before it reaches Block B. As more and more blocks are added, the complex interactions between blocks determine the actual voltage drops. The placement of these blocks is typically based on the timing requirements of the system or the size and shape of blocks at the floor-planning stage. Therefore, sizing the power buses properly to minimize IR drop, while satisfying the required timing and area constraints, is a key design challenge.

The power busses could be widened to lower the resistance of the lines as in Figure 11.7b. The large metal trunks of power have to be sized to handle all the current for each block. Although routing power in this way makes it easier to manage IR drop, it also requires more area to implement. This forces designers to set aside large areas for power busing that takes away from the available global signal routing area. Since the same routing tracks are used for clocks, busses, and other global signals, the tradeoff between power and signal routing must be carefully managed.

Other alternatives are shown in Figure 11.8. In Figure 11.8a, current is delivered from both ends of the block thereby minimizing IR drop in the middle. Since the total IR drop is based on the resistance seen from the pin to the block, one could route around the block and feed power to each block separately as shown in Figure 11.8b. Ideally, the main trunks should be wide enough to handle all the current flowing through separate branches. However, the T-junctions have a high current density as current crowding occurs around the bends. It is important in this type of grid to examine the current density at all junctions, especially the corners, to ensure that EM problems do not exist.

Another approach to minimizing IR drop, depicted in Figure 11.9, is to have a grid of two metal layers. In the figure, assuming a five-layer metal process, Metal 4

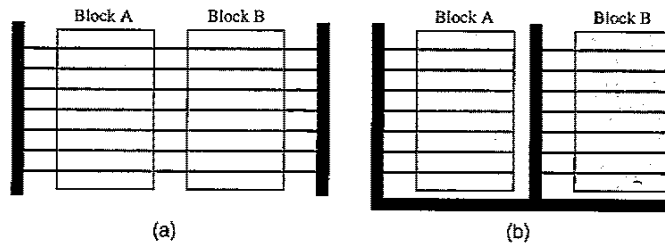


Figure 11.8
Simple techniques to reduce IR drop.

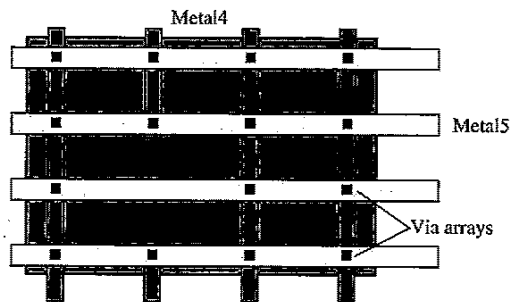


Figure 11.9
Vias in a mesh array methodology.

and Metal 5 use a via array to connect the two layers, effectively tying the whole grid to V_{DD} . However, part of the grid may have to be removed to route some signals. If you arbitrarily remove a strap that is conducting a large amount of current, the excess current must flow in adjacent straps that may push the current density in them beyond acceptable levels for electromigration. These examples all illustrate that design tradeoffs must be made with both EM and IR in mind.

11.2.4 Decoupling Capacitance Design

On-chip decoupling capacitances (decaps) are commonly used to keep the power supply within the noise budget, especially during peak demand periods in high-frequency switching applications. The decoupling capacitors are large-valued capacitances that hold a reservoir of charge located near the power pins and any large drivers, as shown in Figure 11.10. When large buffers switch from low to high, these decaps are the first line of defense for IR drop and Ldi/dt effects. The needed current for the switching process is obtained from the local decap. Later, current flows from the V_{DD} pad to refill the reservoir of charge for the next switching operation.