Chapter

16

Memory Circuits

In this chapter we turn our attention towards the design of semiconductor memory circuits. This includes the array design, sensing, and, finally, the operation of the memory cells themselves. The memories we look at in this chapter are termed random access memories or RAM because any bit of data can be accessed at any time. A block diagram of a RAM is shown in Fig. 16.1. At the intersection of a row line (a.k.a., word line) and a column line (a.k.a., a digit or bit line) is a memory cell. External to the memory array are

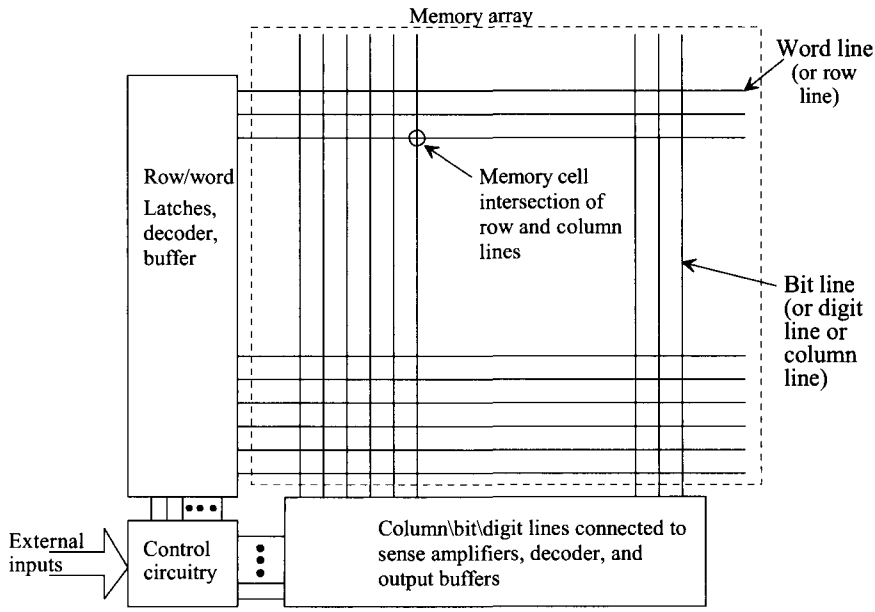


Figure 16.1 Block diagram of random access memory.

the row and column logic. Referring to the row lines, the row address is latched, decoded, and then buffered. A particular row line will, when selected, go high. This selects the entire row of the array. Since the row line may be long and loaded periodically with the capacitive memory cells, a buffer is needed to drive the line. The address is latched with signals from the control logic. After a particular row line is selected, the column address is used to decode which of the bits from the row are the addressed information. At this point, data can be read into or out of the array through the column decoder. The majority of this chapter presents the circuits used to implement a RAM.

16.1 Array Architectures

Examine the long length of metal shown in Fig. 16.2. Let's treat this metal line as one of the bit lines seen in Fig. 16.1. The parasitic capacitance of this bit line to ground (substrate) can be calculated using

$$C_{col1sub} = Area \cdot C_{1sub} \tag{16.1}$$

If the capacitance from the metal lines to substrate is 100 aF/μm^2, then

$$C_{col1sub} = (0.1)(100)(100\ aF) = 1\ fF \tag{16.2}$$

Not that significant of a capacitance. However, at the intersection of the bit line with every word line we have a memory cell. Let's say that we have a memory cell every 400 nm (250 total cells or word lines) and that each memory cell is connected through an NMOS device to the bit line. As seen in Fig. 16.3, this results in a periodic (depletion or junction) capacitance on the bit line from each MOSFET's source or drain implant. If this capacitance is 0.4 fF, then the total capacitance hanging on the bit line is

$$C_{col} = (\text{number of word lines}) \cdot (\text{capacitance of the MOSFET's source/drain}) + C_{col1sub} \tag{16.3}$$

or $C_{col} = 101\ fF \approx 100\ fF$ for this discussion. For the majority of the discussions in this chapter, we'll treat the column conductor as a capacitor, C_{col}. Note that increasing the number of word lines (the number of memory cells connected to a bit line) increases the bit line capacitance.

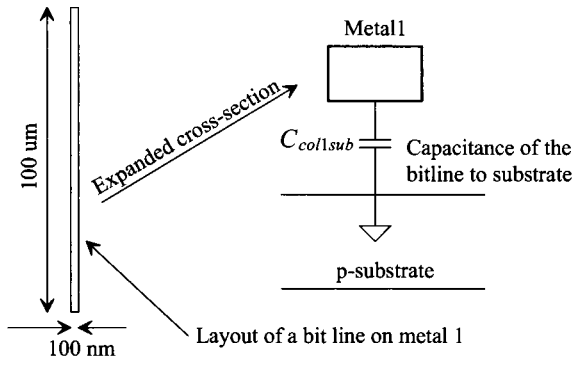


Figure 16.2 The parasitic capacitance of a bitline to ground (substrate).

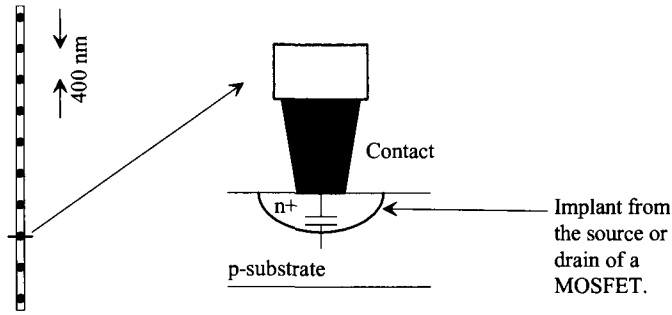


Figure 16.3 How a bit line is loaded with implants (depeletion capacitance).

16.1.1 Sensing Basics

In this section we discuss sensing the datum from a memory cell. Examine Fig. 16.4. When a memory cell is accessed (the word [row] line goes high), the datum from the cell (a charge) is placed on the bit line. The bit line voltage changes. At this point we know the bit line looks like a capacitor and that only one word line can go high at a time in a memory array (so that we don't have two memory cells trying to dump their data onto the same bit line at the same time). The voltage movement on the bit line, ΔV_{bit}, may be very small, e.g., 50 mV or less, and so determining if the voltage is moving upwards or downwards can be challenging. In addition, we would like our sense amplifier to drive the bit line to full, valid, logic levels (for speed reasons in some memories and to refresh the cell in a dynamic RAM, DRAM). Let's consider some sense amplifier topologies.

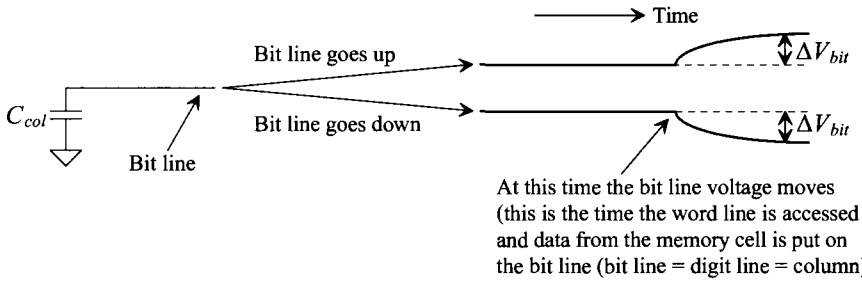


Figure 16.4 Sensing a change on the bit line in an array.

NMOS Sense Amplifier (NSA)

Consider the schematic of an NMOS sense amplifier (NSA) seen in Fig. 16.5. M1 and M2 form the NMOS portion of an inverter-based latch. The idea is to develop an imbalance on the gates of M1/M2 so that one MOSFET turns on harder than the other. Before we start sensing, we set the drains of M1/M2 to the same potential (an equilibrium potential). When *sense_N* goes high (indicating that we are starting the sense operation), the signal *NLAT* (N-latch) goes to ground (the sources of M1/M2 move to ground). While this circuit clearly can't pull the bit line high (we'll add a PMOS sense amplifier later to do this), we should see a problem. How do we get good sensing if the transistor loads are not balanced?

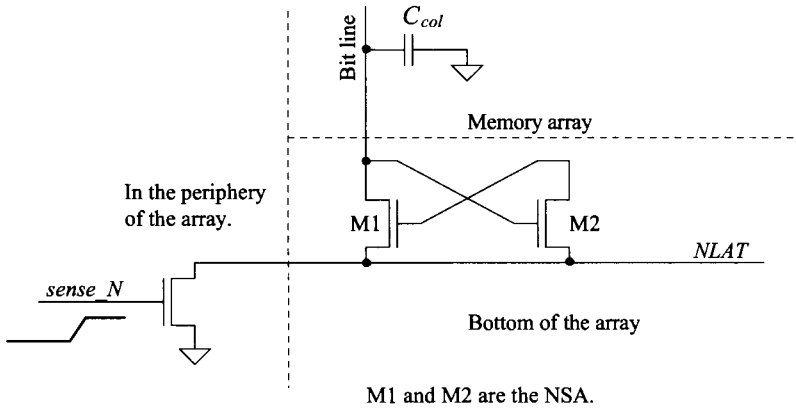


Figure 16.5 Development of an NMOS sense amplifier (NSA).

The Open Array Architecture

To provide the same load capacitance for each input of the NSA, two arrays can be used, as seen in Fig. 16.6. The array architecture in Fig. 16.6, from the side, looks like an open book and so it is called an "open array architecture." Let's use some numbers and the partial schematic seen in Fig. 16.7 to illustrate the sense amplifier's operation.

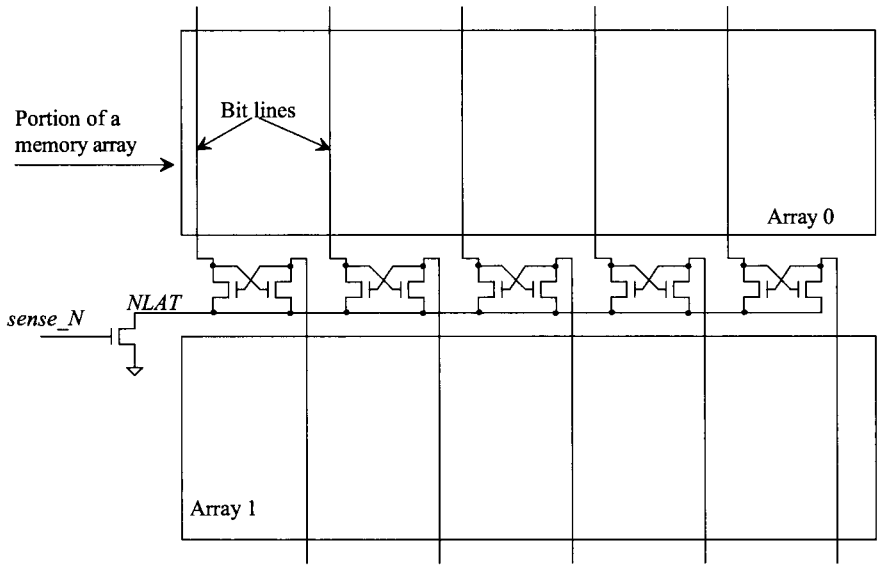


Figure 16.6 How the NSA is placed between two memory arrays in the so-called open memory array architecture.

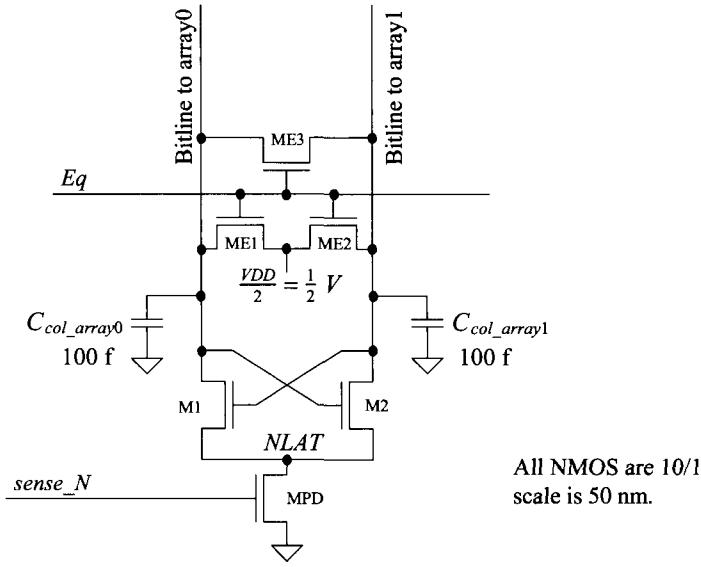


Figure 16.7 An NSA with equilibration circuitry and connections to two bit lines.

In the following we assume *VDD* = 1 V and the column capacitance is 100 fF. We start any sense operation by equilibrating the bit lines (the inputs to the sense amplifier). In Fig. 16.7 ME1–ME3 short the bit lines together and to *VDD*/2 (= 0.5 V). When the equilibrate signal, *Eq*, goes high, all of the word lines are low. We are not accessing any data in the array but, rather, getting ready for the sense (read) operation. Figure 16.8 shows how the *Eq* signal is asserted for a short period of time and how, during this time, the bit lines are equilibrated together and to 0.5 V.

During a sense operation, one of the bit lines will be pulled from *VDD*/2 to *VDD*. The other line will be pulled from *VDD*/2 to ground. The amount of power used during a sense depends on frequency and given by

$$P_{avg} = (\text{number of sense amplifiers}) \cdot C_{col} \cdot (VDD/2)^2 \cdot f \qquad (16.4)$$

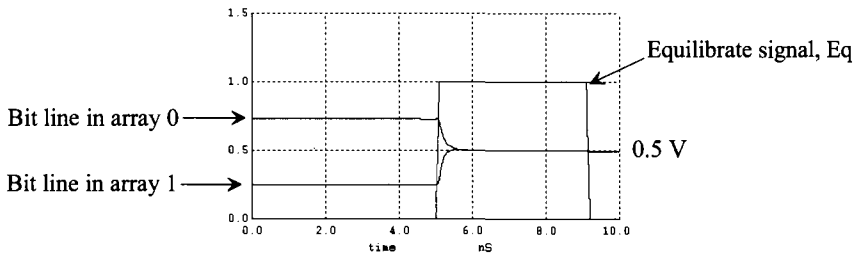


Figure 16.8 How the equilibrate circuitry operates.

Note, however, that when we equilibrate, no power is consumed. The two bit line capacitances simply share their charge (one going from ground to *VDD*/2 while the other goes from *VDD* to *VDD*/2).

To show a sensing operation, let's use the basic one-transistor, one-capacitor (1T1C) DRAM memory cell seen in Fig. 16.9. To fully turn on the access MOSFET, we need to drive the word line to $VDD + V_{THN}$ (with body effect). A typical value for the memory bit capacitance, C_{mbit}, is 20 fF. If the voltage on this capacitance is called V_{mbit} and the bit line is precharged to *VDD*/2, then we can write the total charge on both capacitors *before* the access MOSFET turns on as

$$Q_{tot} = C_{mbit}V_{mbit} + (VDD/2) \cdot C_{col\_array} \tag{16.5}$$

After the access MOSFET turns on, the voltage across each capacitor will be the same. We'll call this voltage, V_{final}. Since charge must be conserved

$$V_{final}(C_{mbit} + C_{col\_array}) = C_{mbit}V_{mbit} + (VDD/2) \cdot C_{col\_array} \tag{16.6}$$

or

$$V_{final} = \frac{C_{mbit}V_{mbit} + (VDD/2) \cdot C_{col\_array}}{C_{mbit} + C_{col\_array}} \tag{16.7}$$

If a logic one is stored on C_{mbit} (which means V_{mbit} is *VDD*, or here, 1 V) and C_{mbit} = 20 fF and C_{col_array} = 100 fF, then V_{final} = 0.583 V. The change in the bit line voltage is

$$\Delta V_{bit} = V_{final} - VDD/2 \tag{16.8}$$

or here $\Delta V_{bit} = 83\ mV$.

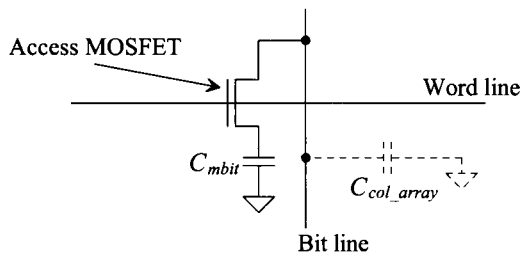


Figure 16.9 The one-transistor, one-capacitor (1T1C) DRAM memory cell.

Figure 16.11 shows the signals resulting from the operation of the sense amplifier in Fig. 16.10. The 1T1C DRAM (mbit) cell in array1 (the DRAM cell at the bottom of the schematic) has its word line held at ground when we sense the data in the cell in array0. The bit line from array1 is simply used as a reference. When we sense the data in array1, the bit line in array0 is used as the reference (and so the word line in array0 is held at ground).

For the simulation results in Fig. 16.11, we start out by equilibrating the bit lines (as in Fig. 16.8). Next our word line, in array0, goes to a voltage greater than $VDD + V_{THN}$. This causes charge sharing between C_{mbit} and the digit line capacitance. For the simulation results seen in Fig. 16.11, we set the voltage on C_{mbit} to zero so we are reading out zero.

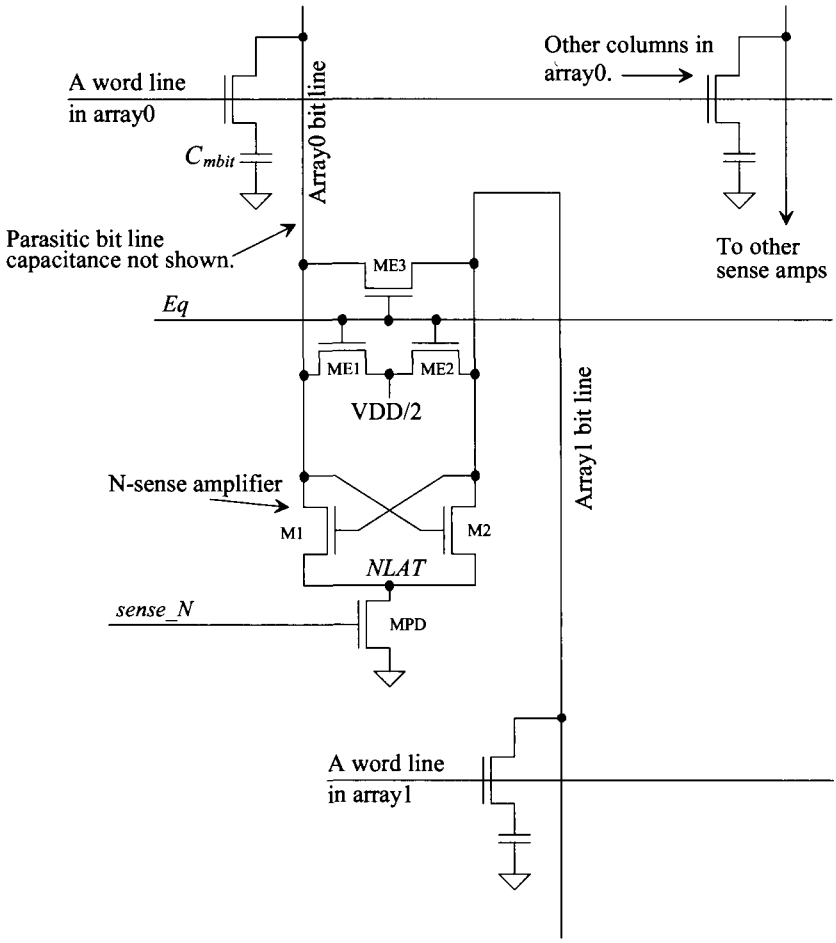


Figure 16.10 The connection of the NSA to the memory arrays.

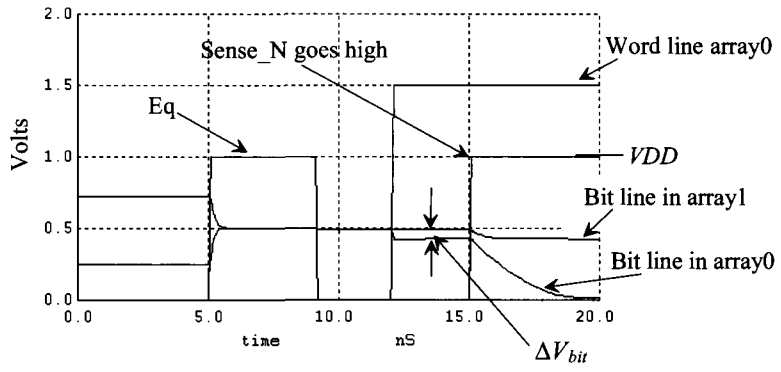


Figure 16.11 N-sense amp's operation.

When *sense_N* goes high, the NSA "fires" causing the bit line in array0 to move to ground. Note that our reference bit line (from array1) droops a little. Figure 16.12 shows the signals in a sensing operation if the mbit cell contains a "1." The bit line voltage in array0 now increases. The reference bit line in array1 is pulled to ground (this doesn't harm the data in array1). To pull the bit line in array0 high, we'll now add a PMOS sense amplifier (PSA).

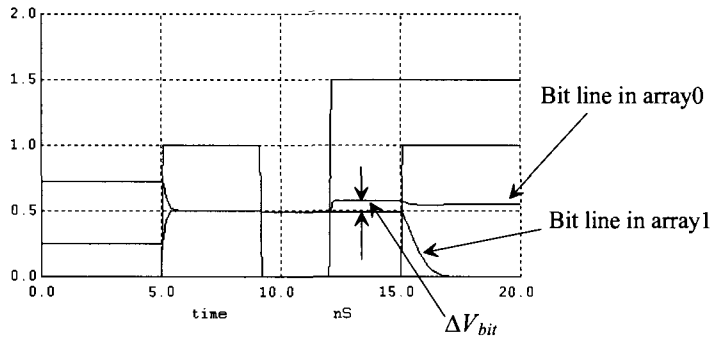


Figure 16.12 Reading out a "1" from the cell in array0.

PMOS Sense Amplifier (PSA)

To pull the bit lines up to *VDD*, we can add a PSA to the periphery of the array. Figure 16.13 shows the schematic of the PSA. The signal *ACT* (active pullup) is common to all of the PSAs on the periphery of the array (as is *NLAT* for the NSAs seen in Fig. 16.6). The signal $\overline{sense\_P}$ is active low and indicates that the PSA is firing. The PSA is usually fired after the NSA because the matching of the NMOS devices is, generally, better than

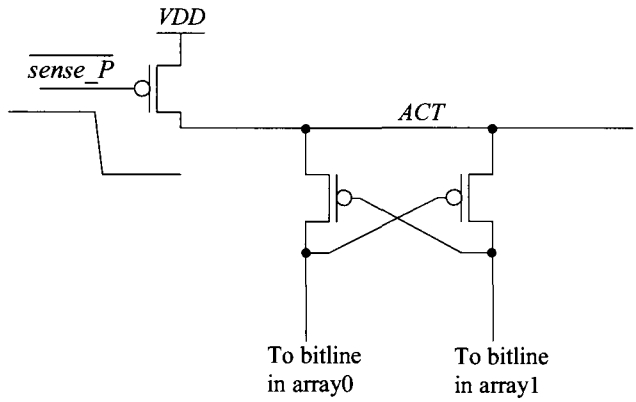


Figure 16.13 Schematic diagram of a PMOS sense amplifier.

the matching of the PMOS. It's not common to fire the sense amplifiers at the same time because of the potentially significant contention current that can flow. Figure 16.14 shows the full operation of a sense when the cell we are reading out has a one stored in it.

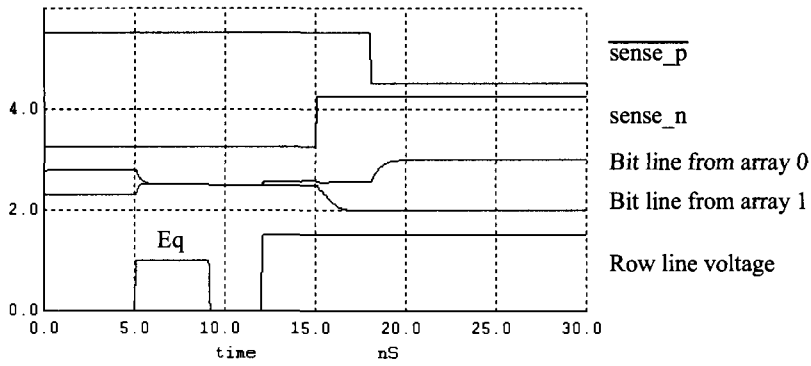


Figure 16.14 How the PSA pulls the bit line from array 0 high.

Refresh Operation

Our sensing operation determined whether a 1 or a 0 was stored in the memory bit. For this particular sense amplifier topology, the inputs and outputs are the same terminals. This is fundamentally important for DRAM operation where the charge can leak off the capacitor because of the finite subthreshold slope of the access transistors or leakage through the MOSFET's source/drain implant to substrate (hence the name "dynamic"). To ensure long-lasting data retention in a DRAM, the cells must be periodically refreshed. When we fire the sense amplifiers, with the access device still conducting, the mbit capacitor is refreshed through the access MOSFET (the bit line is driven to ground or *VDD* by the sense amp).

16.1.2 The Folded Array

The open array architecture seen in Fig. 16.6 has a memory cell at the location of every intersection of a word line and a bit line. The open array architecture results in the most dense array topology. Unfortunately, because of the physical distance between the bit lines used by the sense amplifier, this architecture is sometimes not used. Figure 16.15 shows the basic problem. Coupled noise (e.g., from the substrate) feeds unequal amounts

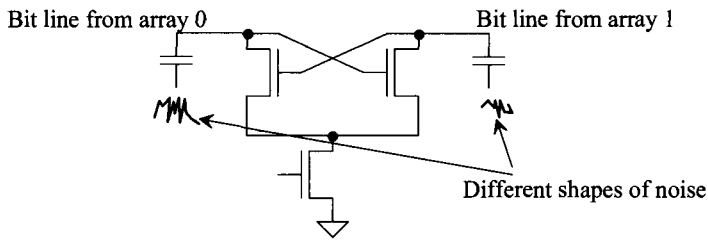


Figure 16.15 Different amplitudes of coupled noise into physically separated bit line.

of charge into the bit lines. This can cause the sense amplifier to make wrong decisions. To attempt to make each bit line see the same sources of noise, we can lay the bit lines out next to each other by folding array 1 on top of array 0 (see Fig. 16.6). The result, called a *folded array architecture*, is seen in Fig. 16.16. The bold lines in this figure are the bit lines from array 1.

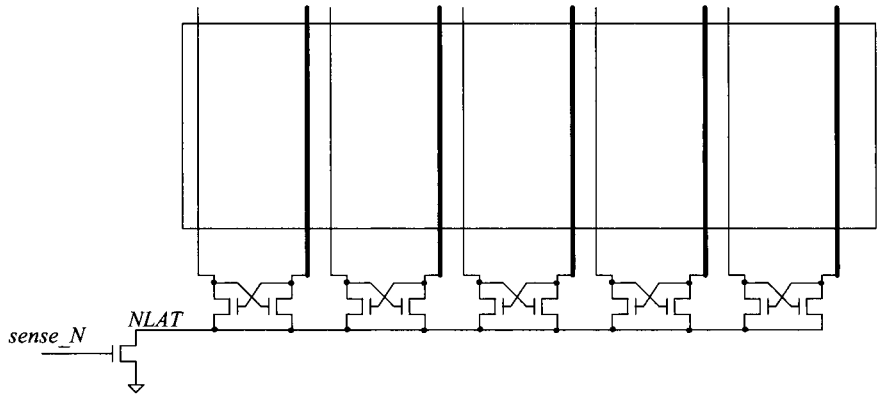


Figure 16.16 The folded array is formed by taking the open array architecture (open book) topology seen in Fig. 16.6 and "closing the book," that is, folding array 1 on top of array 0. Note that the bold lines indicate the bitlines from array 1 in the newly formed array.

What is the cost for this improved noise performance? We know that when a sense amplifier is used, one input is varied by the memory cell we are sensing, while the other input is simply used as a reference. What this means is that instead of having a memory cell at the intersection of every row and column, as in the open array, now we have a memory cell at the location of every other row and column, Fig. 16.17.

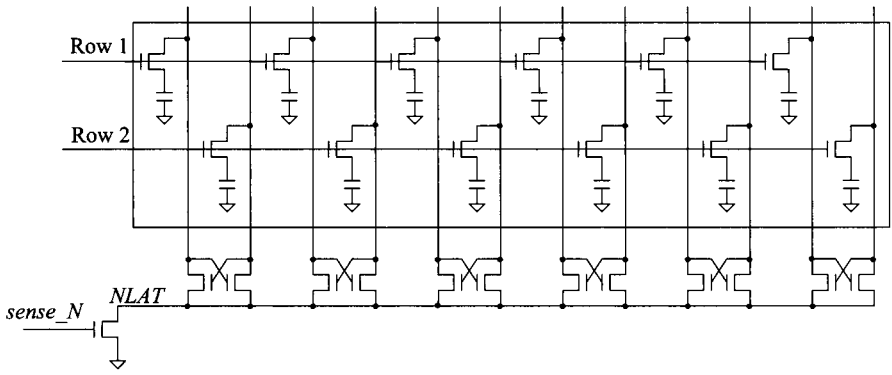


Figure 16.17 How a memory cell is located at every other intersection of a row line and a column line in a folded-area architecture.

Layout of the DRAM Memory Bit (Mbit)

Layout area, especially in a periodic array like a memory, is very important. Reducing the chip size increases the number of die on a wafer. Since the processing cost for a wafer is fixed, having more chips to sell per wafer increases the manufacturer's profit. To reduce the size of an mbit, often (always in DRAM) the contact to the bit line is shared between two mbit cells, Fig. 16.18. The word lines (row lines) are made using silicided polysilicon. Using polysilicon for the word lines can lead to significant delays. As seen in Fig. 16.17, for example, a signal applied to row 1 must propagate down a distributed R (the resistance of the polysilicon) C (the gate oxide capacitance of the MOSFET) line. The propagation delay through a word line can be estimated using

$$t_d = \overbrace{(\text{number of columns}) \cdot \left(WL\frac{\varepsilon_{ox}}{t_{ox}} + C_{parasitic} \right)}^{\text{total capacitance on the word line}} \cdot \overbrace{(\text{number of columns}) \cdot R_{gat}}^{\text{total resistance of the word line}} \qquad (16.9)$$

The number of columns is simply the number of MOSFETs in the row. The term $WL\frac{\varepsilon_{ox}}{t_{ox}}$ is C_{ox}, while R_{gate} is the resistance from one end of the polysilicon gate to the other end in the memory cell layout seen in Fig. 16.18. The term $C_{parasitic}$ is the parasitic capacitance associated with the cell (such as the capacitance from the word line to the bit line). If C_{ox} is 400 aF, $C_{parasitic}$ = 100 aF, R_{gate}= 4 Ω, and there are 512 bit lines in the array then the delay time through the word line is estimated as 500 ps. To fully turn the word line on (not just to the 50% point where delay is measured), we would probably want 3 ns.

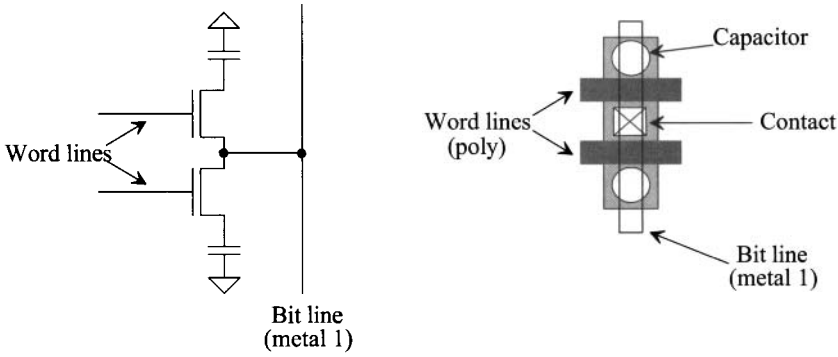


Figure 16.18 Two mbits sharing a contact to the bit line.

A section of the area layout for an open architecture DRAM is seen in Fig. 16.19. Notice that at the intersection of every bit and word line is a memory cell. A common term used to describe the density of, or distance in, a periodic array is *pitch*. As seen in Fig. 16.19, the pitch is defined as the distance between like points in the array. We used the distance between the right side of the digit lines to show pitch in this figure. A figure of merit for memory cell layout is its area. The area of the mbit DRAM cell seen in this figure is $6F^2$ where F = pitch/2. Question: How many MOSFETs are laid out in Fig. 16.19? Answer: Because polysilicon over active forms a MOSFET, we simply count the number of times poly crosses active (12 MOSFETs).

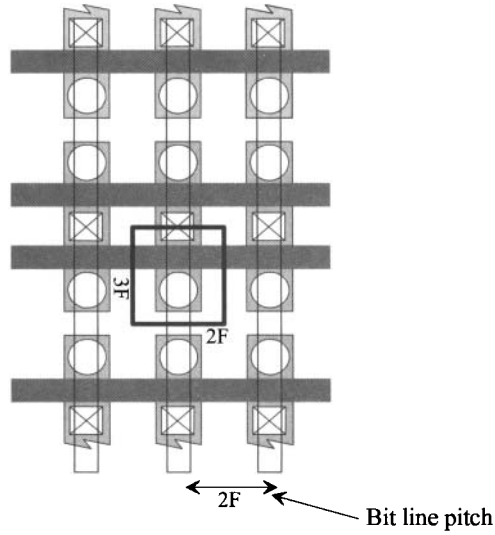


Figure 16.19 Layout of mbit used in an open bit line configuration.

Figure 16.20 shows the layout of the cell used in the folded area architecture. The schematic seen in this figure is different from the one seen in Fig. 16.17. In Fig. 16.20 our mbits share the contact to the bit line. This means that memory cells are located at the intersection of every other mbit pair with the bit lines. Figure 16.21 shows a section of the array layout in the folded architecture. The cell size is $8F^2$. The increase in the cell size is due to the needed poly interconnect between adjacent cells (the poly that runs over the field oxide, FOX).

Figures 16.22 and 16.23 show the process cross sectional views for mbits using trench capacitors and buried capacitors, respectively. The trench capacitor is formed by etching a hole in the substrate. For high-density memory, the aspect ratio (the ratio of the

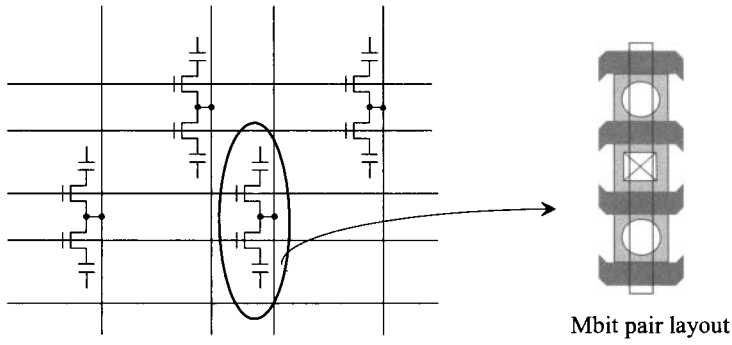


Figure 16.20 The mbit pair used for a folded architecture.

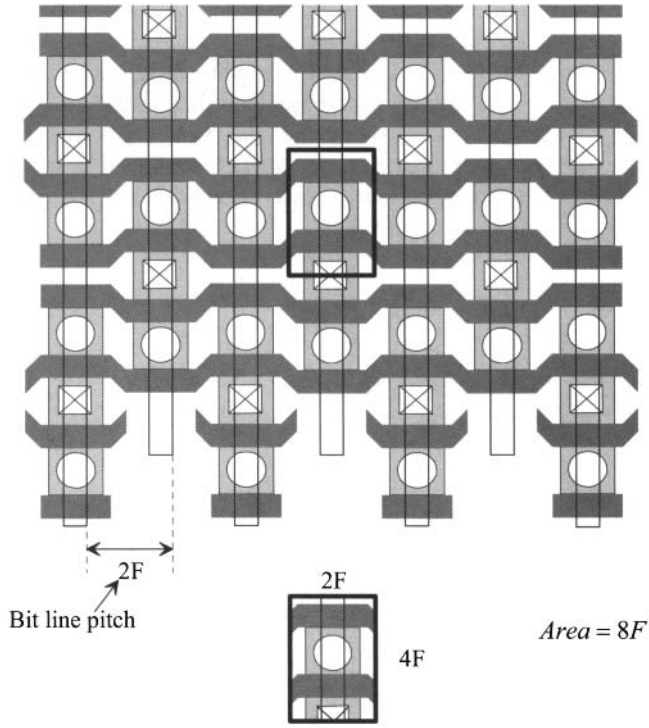


Figure 16.21 Folded architecture layout and cell size.

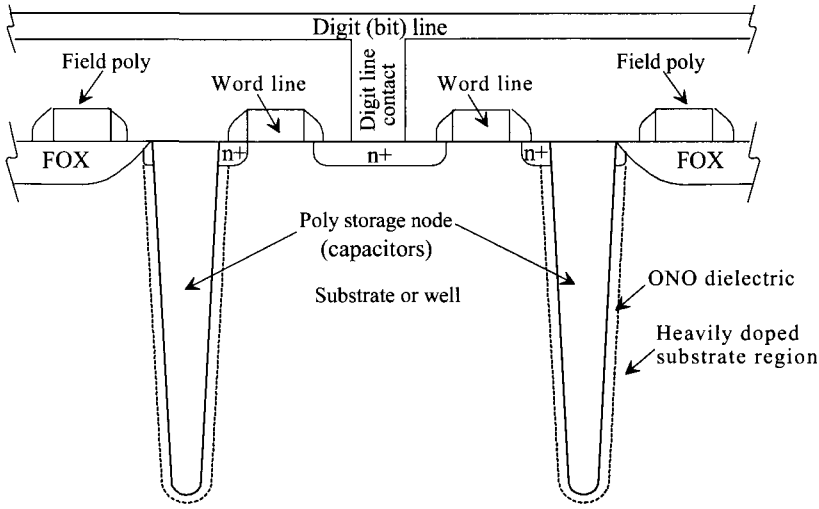


Figure 16.22 Cross-sectional view of a trench capacitor cell.

hole's depth to its diameter) can be quite high, which leads to processing concerns. In Fig. 16.23 the buried capacitor cell pair is seen. Unlike the trench capacitor-based cell that places the cell directly in the substrate, the capacitor is "buried" under the digit line but still above the substrate. The benefit of this cell is simpler processing steps. One problem with this cell, when compared to the trench-based cell, is that the parasitic capacitances are higher (such as the capacitance loading the (digit) bit line). Also, for a given bit capacitance, C_{mbit} , the area of the buried capacitor cell may need to increase, while the area of the cell using the trench capacitor remains constant. The depth of the trench capacitor is increased for more capacitance.

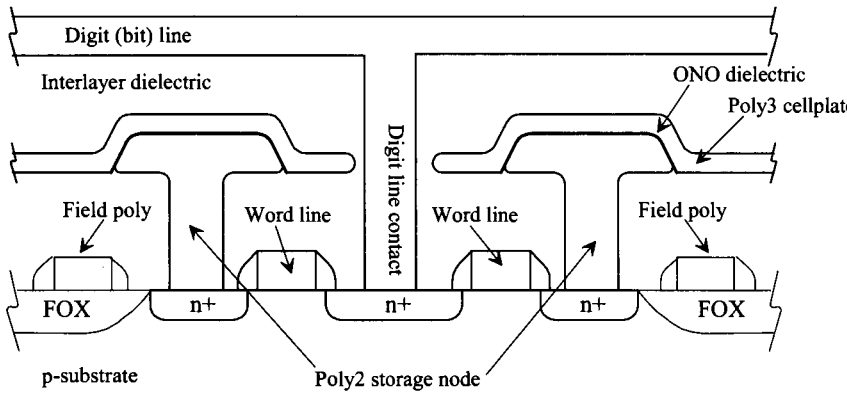


Figure 16.23 Cross-sectional view of a buried capacitor cell.

As we saw in Eq. (16.7), the value of C_{mbit} is very important. Thin dielectrics with high dielectric constants are used to implement C_{mbit}. Tricks like roughing up the dielectric to increase the surface area are often employed. To minimize the stress on the oxide, the common node of the capacitor is usually tied to *VDD*/2, as seen in Fig. 16.24. When a logic 1 (*VDD*) is written to the cell, the charge on the capacitor is $(VDD/2) \cdot C_{mbit}$. When a logic 0 (ground) is written, the charge on the capacitor is $-(VDD/2) \cdot C_{mbit}$. The difference in these charges is $VDD \cdot C_{mbit}$, which is the same difference we would have if the common node were connected to ground.

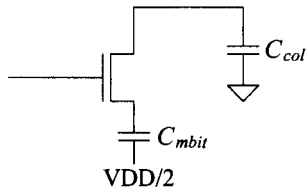


Figure 16.24 Holding the capacitor's common plate at VDD/2 to minimize oxide stress.

16.1.3 Chip Organization

The number of rows in a DRAM is limited because each additional row adds capacitance to the bit line. The power goes up, as seen in Eq. (16.4), with increasing C_{col}. The number of bit lines is limited by the delay through the row line (the length of the row line), as indicated in Eq. (16.9). For these reasons, the size of a DRAM array is limited. A typical array size is 512 word line pairs and 512 bit lines for a total memory size of 256 kbits. We use the term "pairs" to indicate that, as seen in Fig. 16.20, there are actually 1,024 row lines in a 256k array. We can arrange these 256k arrays to form a larger memory array. Figure 16.25 shows the basic idea for an 16-Mbit array block. A 1-Gbit DRAM uses 8,192 256k arrays. These arrays may be further subdivided into "banks" of memory. We might, for a 1-Gbit memory, have 8 banks of 128-Mbits. We subdivide the memory so that an operation (say a read) can be done in one bank while, at the same time, some other operation (say a refresh) can be performed in a different bank.

The *array efficiency* is defined as the ratio of the area of the memory blocks to the total chip area. A typical value ranges from 50 to 60%. The peripheral circuitry seen in Fig. 16.25 (the sense amplifier and row decoder blocks) take up a significant amount of chip real estate. Because of their importance, we'll devote the entire next section to the design of peripheral circuits including sense amplifiers, row drivers, and decoder circuits.

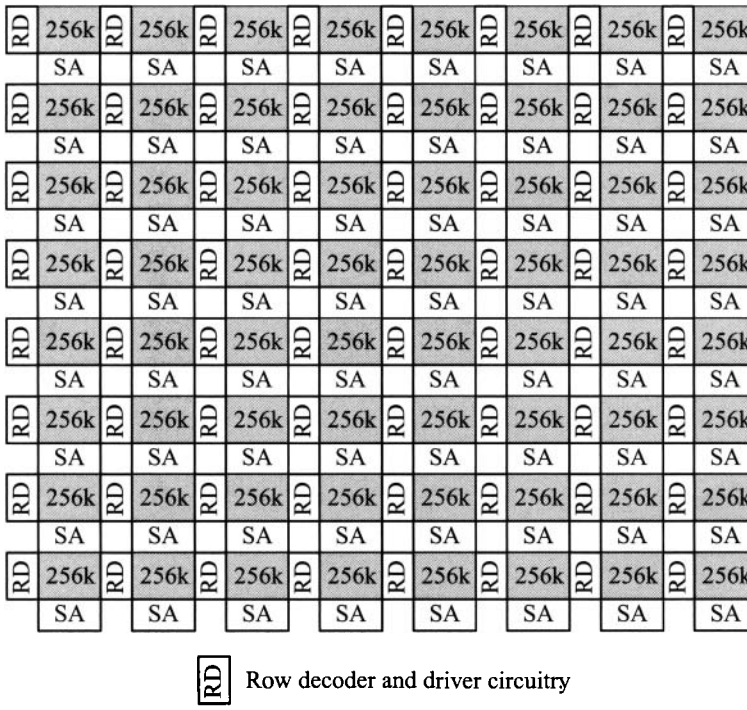


Figure 16.25 A 16-Mbit array block.

16.2 Peripheral Circuits

In this section we discuss the design of general sense amplifiers (for general use in memory design), row drivers (remembering, for a DRAM, the row voltage needs to be driven above *VDD*), and column and row decoder circuits.

16.2.1 Sense Amplifier Design

Examine the clocked sense amplifier seen in Fig. 16.26. When *clock* is low, MS3 is off while MS1 and MS2 are on. Our input signals can't go below V_{THP} (with body effect) without shutting MS1 and MS2 off. Assuming that the inputs stay above V_{THP}, the drains of M1/M3 and M2/M4 are charged to In+ or In– (creating an imbalance). When *clock* goes high, the imbalance causes the circuit to latch high or low depending on the state of the inputs. Figure 16.27 is the simulation output showing the typical operation of the circuit. When *clock* is low, the circuit outputs are not valid logic signals but rather, ideally, track the input signals. This circuit is plagued with problems including: kickback noise, memory, and significant potential contention current.

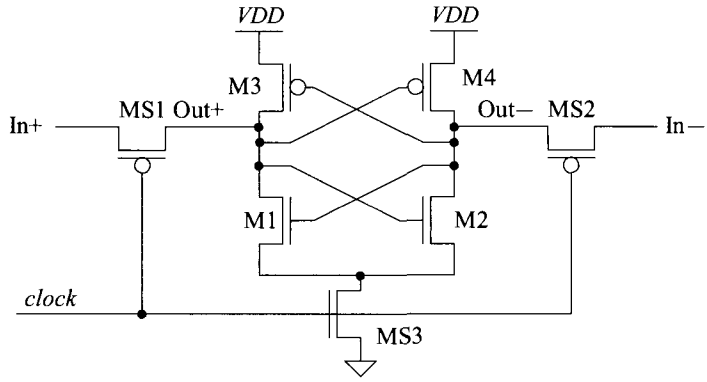


Figure 16.26 Clocked sense amplifier.

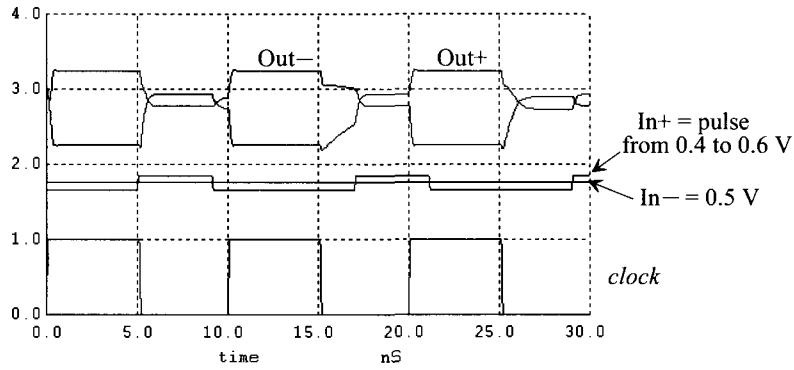


Figure 16.27 Simulating the operation of the circuit in Fig. 16.26.

Kickback Noise and Clock Feedthrough

In the simulation that generated Fig. 16.27 we used, for our inputs, voltage sources. In a real application, the inputs would come from some other logic or a bit line (a source with a finite driving impedance). Consider including logic for driving the sense amplifier seen in Fig. 16.28. We use long-length inverters to simulate a weak driver circuit (such as a decoder made with a series of transmission gates). Seen in the figure are simulation results at the time when the clock goes high (we get similar glitches when the clock goes low). The large glitch seen in these figures is called *clock feedthrough* noise. It is present when a clock signal has a capacitive path directly to the input of the sensing or comparator circuit. The clock feedthrough noise in Fig. 16.28 is close to 50 mV on both inputs.

Another type of noise, called *kickback noise*, is present and injected into the inputs of the circuit when the latch switches states. *It's important to simulate the operation of the sense amplifier with nonideal sources (with finite source resistance) to determine the significance of clock feedthrough and kickback noise.* Using voltage sources with 10k resistors is a reasonable source for general circuit simulations. This noise is an important specification for a sense amplifier. It can often be the *limiting factor* when making sensitive measurements. If the kickback noise, for example, is too large, it can interfere with sensing. A simple example of a potential problem occurs when 256 sense amps are all being clocked at the same time (e.g., on the bottom of a memory array) and they are directly adjacent to each other. We don't want one of the amplifiers interfering with any of the others by feeding noise into the others' inputs.

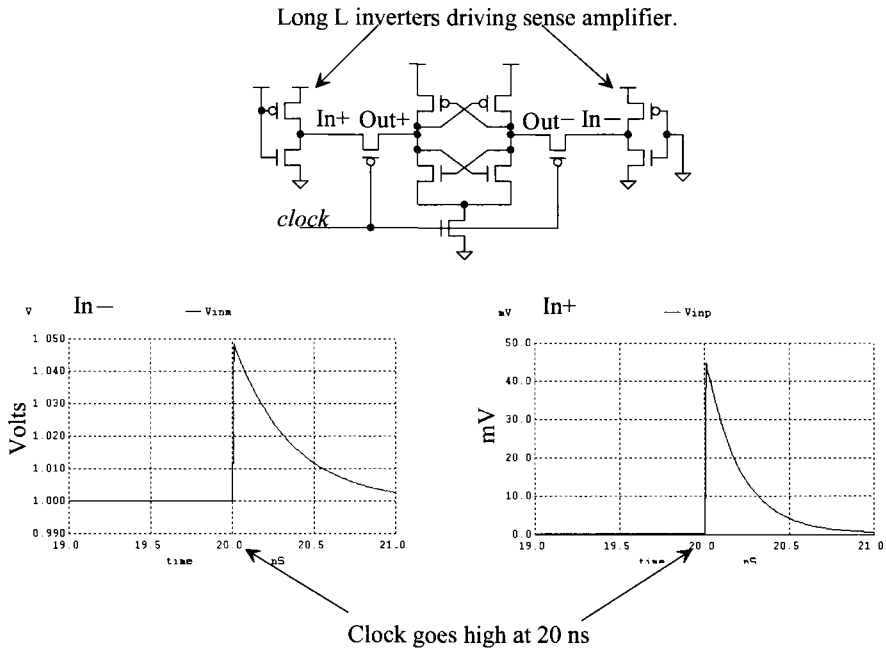


Figure 16.28 Clock feedthrough noise.

Memory

For a good comparison, the sense amplifier shouldn't have a memory of previous sensing operations. For the sense amplifier in Fig. 16.26, the outputs are actively driven by the input signals prior to clocking. However, the drain of MS3 (sources of M1/M2) is floating. This node is a dynamic node that floats to a voltage dependent on the previous decision and the input signals. *For a precision sense operation, all nodes in the sense amplifier must be driven or equilibrated, prior to clocking, to a known voltage.*

Current Draw

Because there may be thousands of sense amplifiers operating on a chip at the same time, it is extremely important to minimize the power used by these circuits. Let's take a look at the current supplied by *VDD* to the circuit in Fig. 16.26 (with the signals seen in Fig. 16.27). Because the inputs actively drive the gates of M3 and M4, it is possible M3/M4 source a significant current into the inputs. In this case, Fig. 16.27, the V_{SG} voltages are roughly 0.5 V (and so both M3 and M4 are conducting a drain current). The current supplied by *VDD* is seen in Fig. 16.29. *Clock* is high during 0 to 5ns, 10ns to 15ns, etc. (when the current supplied by *VDD* is small). If the average current is 50 μA and there are 1,000 sense amplifiers operating on the chip, then *VDD* supplies 50 mA of current (just to the sense amplifiers). *For minimum power, it's important that there are no DC paths from VDD to ground except during switching times.*

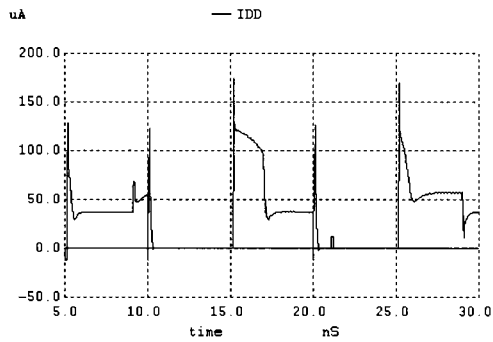


Figure 16.29 The current that flows in the sense amplifier in Fig. 16.26.

Contention Current (Switching Current)

When switching takes place in the sense amplifier of Fig. 16.26, both M3 and M4 are conducting. When *clock* goes high, both M1 and M2 turn on and conduct current as well. In this situation there is a direct path from *VDD* to ground. If the inputs are at relatively the same voltages, the time that M1–M4 remain conducting can be very long (resulting in significant current being pulled from *VDD*). To eliminate this *metastable* condition and force the comparator to make a decision (switch to valid logic levels), the gain in series with the input signals can be increased. A simple example of increasing input signal gain is to place an amplifier on the inputs of the latch.

Note that if our input signals amplitudes are within a V_{THP} of *VDD*, the PMOS devices are off and we don't have this problem. The inputs can generate an imbalance on

the drains/gates of the NMOS transistors when clock goes high. Neither PMOS device will turn on until either M1 or M2's drain moves below $VDD - V_{THP}$. *For low power (low contention current), try to keep one side of the latch shut off until a significant imbalance is present in the circuit.*

Removing Sense Amplifier Memory

Figure 16.30 shows how the sense amplifier's memory can be erased. M1–M4 form a latch (Fig. 13.16). To remove the sense amplifier's memory, *all* nodes in the sense amplifier must be actively driven to a known voltage (no floating or dynamically charged nodes). When *clock* is low, the sense amplifier's outputs are pulled to *VDD* through MS3 and MS4. The MOSFETs MS1 and MS2 are off, breaking the connection between *VDD* and ground (so no current flows in the latch). The gates of M1 and M2 are at *VDD* so their drains are actively driven to ground. In other words, when *clock* is low, all nodes in the circuit are either pulled high to *VDD* or low to ground.

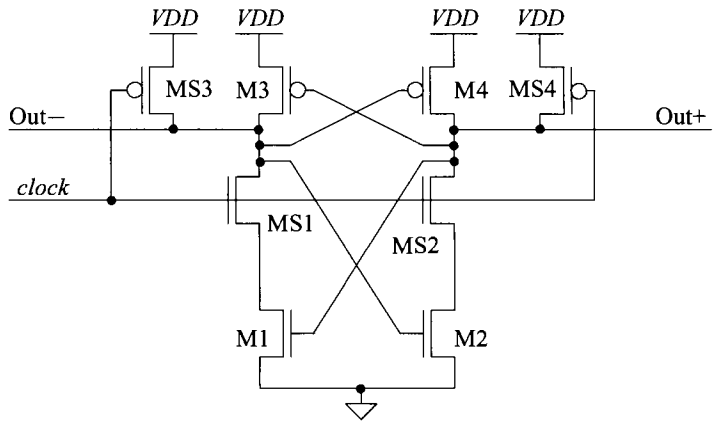


Figure 16.30 Removing memory in a sense amplifier.

Creating an Imbalance and Reducing Kickback Noise

When *clock* goes high, the latch action can take place. For the comparison to function as desired, we need to generate an imbalance. This, alone, isn't too difficult. What is difficult is creating an imbalance for a wide range of input signals while not causing an excessive amount of current to flow in the circuit. Consider the additions to our sensing circuit seen in Fig. 16.31. MB1/MB2 can be connected to either the drain or the source of MS1/MS2, as seen in the figure. Let's first consider connecting them to the drains. When *clock* is low, the drains of MB1 and MB2 are pulled to *VDD*. If the input voltages are significant, a *large* current can be drawn from *VDD* (when *clock* is low). If the input voltages are relatively small, then the current pulled from *VDD* may be tolerable (keeping in mind that the sense amplifier doesn't function correctly when the input voltages are less than V_{THN}). The benefit of connecting to the drains is high gain. Very small voltage differences can cause quick sensing (the outputs swing quickly to valid logic voltages). Note that the possibility for large current flowing through either MB1/MB2 is present independent of the state of *clock*.

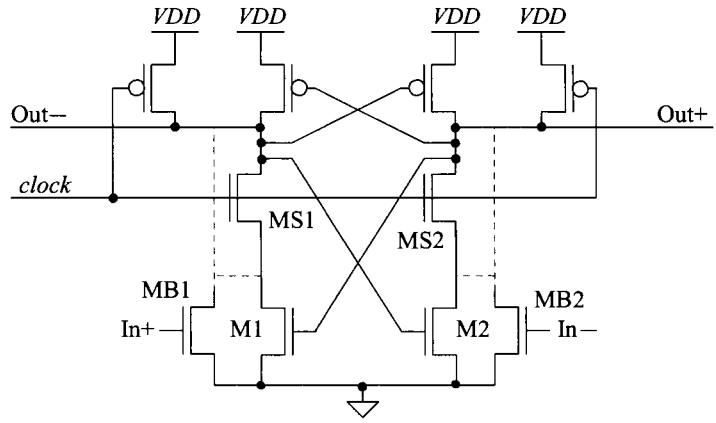


Figure 16.31 Creating an imbalance in a sense amplifier.

Next let's consider connecting MB1/MB2 to the sources of MS1/MS2 (drains of M1/M2). When *clock* goes low, the drains of MB1/MB2 are pulled low through M1/M2. No current flows in the comparator when *clock* is low (unlike when the drains of MB1/MB2 are connected to the drains of MS1/MS2). The gain is much lower because now MB1/MB2 are operating in the triode region when *clock* goes high. The potential for large current flow is still present when *clock* is high but, now, the maximum value on the drain of MB1/MB2 is $VDD - V_{THN}$. And so the potential for MB1/MB2 to move into the triode region can result in a smaller output current.

We seem to have a problem. To reduce the possibility of metastability, we need to increase the gain in series with the signal path (increase the *W/L* ratio of MB1 and MB2; that is, reduce their switching resistance). However, this results in larger power dissipation. How can we keep power down while maximizing the sensitivity of our sense amplifier? While we can use long *L* devices for MB1 and MB2 so they never pull significant current (at the cost of sensitivity and speed) or have a separate pre-amplifier to generate the input signals, let's look at some other ideas (noting that no design is perfect but a trade-off between power, speed, and sensitivity).

Figure 16.32 shows one idea for creating an imbalance without the possibility of significant current flow. MB1 and MB2 are used to create an imbalance in the gate-source voltages of M1/M2. Since, prior to switching, the voltage on the gates of M1/M2 is *VDD*, we can still have good sensitivity even though MB1 and MB2 are operating in the triode region (look like resistors). The large voltage dropped across $V_{GS,M1}$ and $V_{DS,MB1}$ ensure that good gain. It is important, however, that the inputs signals are $> V_{THN}$ to ensure that the circuit operates properly. The sensing fails if this isn't the case because neither MB1 or MB2 can turn on and provide the needed path to ground for proper logic level signals. As long as the inputs are above the threshold voltage, there aren't any dynamic nodes (*no memory in the sense amplifier*). The sources of MS1 and MS2 are still pulled to ground when *clock* is low. *Kickback noise* is reduced using this topology because the inputs are isolated from the latch by MB1/MB2. Kickback noise is still present and must be considered when designing and simulating this sense amplifier.

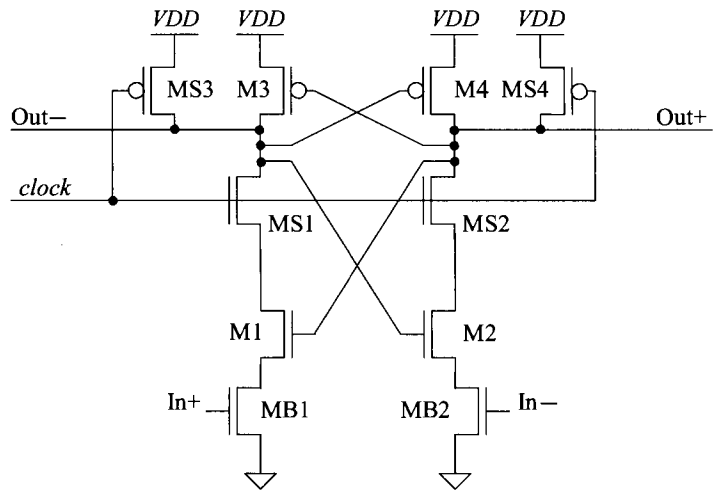


Figure 16.32 Reducing power in a sense amplifier.

Note that we might try to design our comparator to operate over a wide range of input voltages by adding PMOS devices in series with the sources of M3/M4, as seen in Fig. 16.33. We know that MB1–MB4 must be capable of turning on when *clock* goes high. If either pair can't turn on, then the latch won't be able to generate full logic levels. For the design in Fig. 16.33, the input signals would have to fall within V_{THN} and $VDD - V_{THP}$ (more restricted than the circuit in Fig. 16.32). Also, and perhaps more importantly, MB3/MB4 can't generate an imbalance in the latch. If the sources of M1/M2 are connected to ground so that MB1/MB2 don't generate the imbalance, then when *clock* goes high, MB3 and MB4 are off. They won't turn on until the NMOS pair M1/M2 turns on and drops the gates of M3/M4 below $VDD - V_{THP}$. For great differences in the input signals, the comparator may still function correctly. However, for the majority of the input signal differences, the resulting output logic levels is determined by the matching between M1/M2, e.g., the one with the lower "actual" V_{THN} will turn on faster.

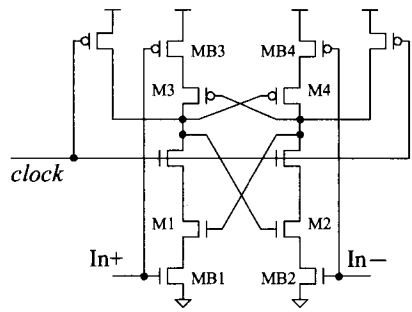


Figure 16.33 A bad design: how not to get wide-swing operation.

Example 16.1

Suppose that it is suggested that the gates of M3/M4 be tied to the drains of M1/M2 or MB1/MB2, respectively, in the sense amplifier (clocked comparator) seen in Fig. 16.32. Comment on the concerns.

If the gates of M3/M4 are tied to the drains of M2/M1, then, when *clock* goes high, a significant contention current flows in the latch. As discussed earlier, it is desirable that one side of the latch (either the NMOS or PMOS pairs) is off when the latch is clocked in order to reduce power. The same concerns are present when these gates are connected to the drains of MB1/MB2.

We might try to eliminate MS3/MS4 when we tie the gates of M3/M4 to the drains of MB1/MB2. Because the drains of MB1/MB2 always move to ground when *clock* is low (again assuming the inputs are > V_{THN}), M3/M4 will turn on and pull the outputs high. When *clock* goes high, though, it's impossible to shut M3/M4 off. Because the maximum voltage on the sources of MS1/MS2 is *VDD* – V_{THN}, the gates of M3/M4 can't go to *VDD* to shut one of the devices off. ■

Increasing the Input Range

Figure 16.34 shows adding, to our basic sense amplifier in Fig. 16.32, MOSFETs MB3–MB8 so that the circuit will function with input signals ranging from 0 to *VDD* (actually it will operate correctly with input signals beyond the power supply rails). We know current can only flow out of the sources of M1/M2, so our additional circuitry, for creating the imbalance, must be capable of sinking current from these sources (and so we'll add MB3 and MB4 for this reason). Next, we know that MB1/MB2 function fine for creating the imbalance if the input signals are above V_{THN}. However, if the input signals are below V_{THN}, they turn off. To level-shift the input, let's use the MB5–MB8. If the input signals are < V_{THN}, then MB7 and MB8 are on. A difference in the input voltages causes different currents to flow in MB5 and MB6. The different currents flowing in these transistors results in different voltages across each one. This voltage difference is then used to create an imbalance in MB3 and MB4. Unfortunately, the addition of MB5–MB6 contradicts our earlier statement that, "*For minimum power it's important that there are no DC paths from VDD to ground except during switching times.*" There is a DC path through MB8/MB6 and MB7/MB5. To lower the power, we can increase the length of MB5 and MB6 so that the continuous current flowing through the DC path is lessened.

Simulation Examples

Let's simulate the operation of the circuit in Fig. 16.32. We know that the outputs of the sense-amp go high every time *clock* goes low. To make the outputs change only on the rising edge of *clock*, we can use the SR latch discussed in Ch. 13 on the output of the circuit, Fig. 16.35. Now, with the addition of the NAND gates, when the sense amplifier's outputs are high, the outputs of the latch don't change from the previous decision (made on the rising edge of clock).

Figure 16.36 shows the current supplied by *VDD* for the circuit in Fig. 16.35. While the current seen in this figure includes the current supplied to the NAND gates, it still should be compared to Fig. 16.29. Notice in Fig. 16.36 that current flows only during the times the clock changes states (unlike the current flow in Fig. 16.29).

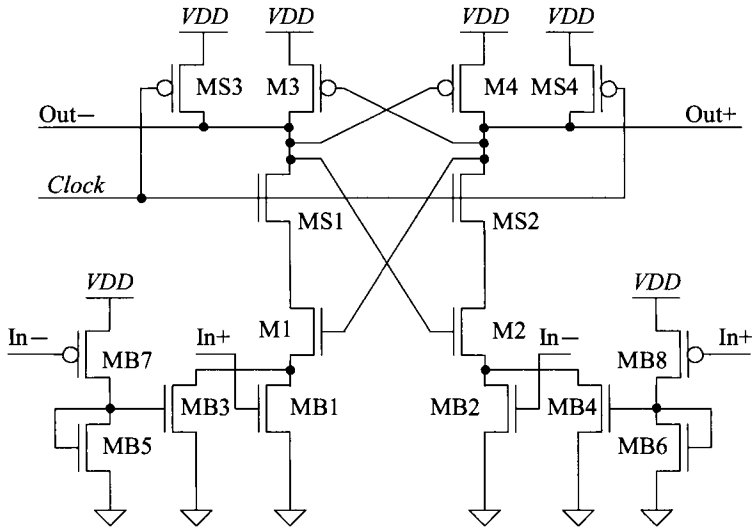


Figure 16.34 Rail-to-rail input signal range. See also Fig. 27.16.

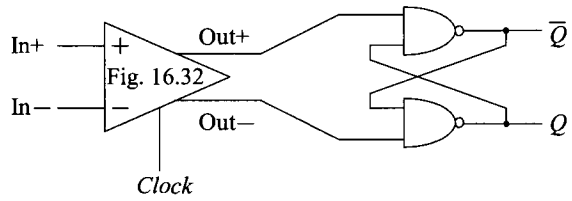


Figure 16.35 Adding an SR latch so output only changes on rising edge of the clock.

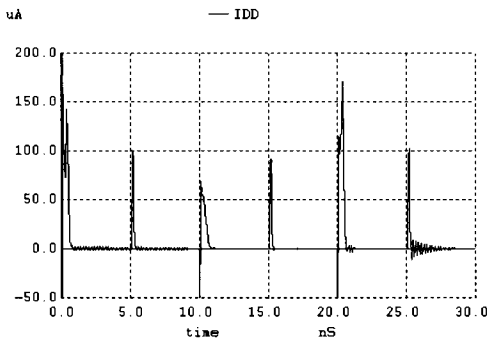


Figure 16.36 Regenerating the plot seen in Fig. 16.29 with the sense amplifier from Fig. 16.35. The current includes the switching current from the NAND gates.

Next let's look at the kickback noise. The clock feedthrough noise is small for this topology because the *clock* is isolated from the input by three transistors. Let's put the sense amplifier in the topology seen in Fig. 16.28 where the inputs are driven to the rails by long-length inverters. The simulation results are seen in Fig. 16.37. The kickback noise is approximately 5 mV.

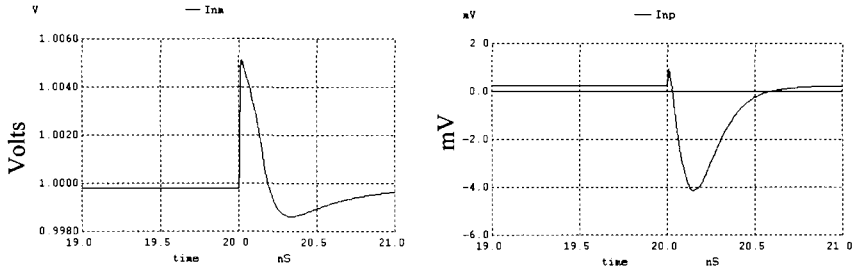


Figure 16.37 Kickback noise.

Figure 16.38 shows the operation of the circuit in Fig. 16.35 where the outputs only change on the rising edge of *clock*. The input signals are the same ones that were used in generating Fig. 16.27 (although the simulation was 20 ns longer in Fig. 16.38). The sensitivity of this sense amplifier is better than 10 mV. The sensitivity can be improved by increasing the lengths of MB1/MB2 (so that they have a large resistance). The drawback of the improved sensitivity is the increased time it takes to pull one of the outputs to ground (the output transition time). The sensitivity can be increased without this penalty by adding a preamplifier (e.g., a differential amplifier) between the sense amplifier and the inputs, see Fig. 27.16.

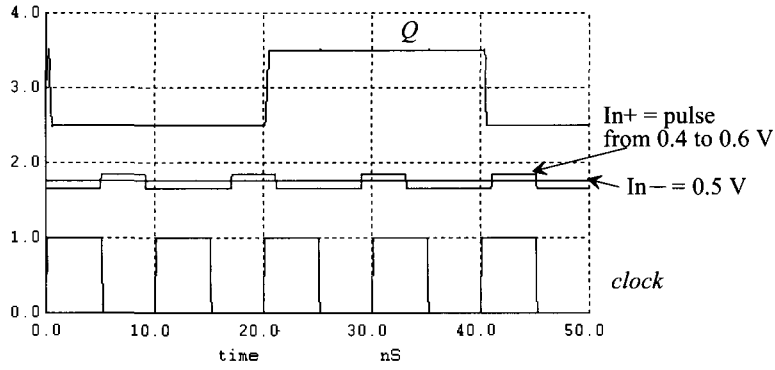


Figure 16.38 The operation of the circuit in Fig. 16.35.

16.2.2 Row/Column Decoders

Reviewing the memory block organization in Fig. 16.25, we may wonder how to address one or more of the 256kbit subarrays. When we say "address," for the row lines, what we mean is that we are driving one, and only one, of the row lines in one or more of the 256k subarrays to a voltage greater than $VDD + V_{THN}$ (to fully turn the access MOSFET on), while holding all other row (word) lines at ground. To accomplish this selection, a *row address decoder* is used. Once the word line has transitioned high and the sensing is complete, the data from the memory cells is present on the bit lines (as discussed in detail in Sec. 16.1). To select the addressed data from one or more of these bit lines (in one or more memory arrays), a *column decoder* is used.

In a large (capacity) memory the chip's address pins are multiplexed, that is, the same set of pins are used for both the row and column addresses. A separate clock signal is used to strobe in either the row or the column addresses, at different times, into some hold registers. In older DRAMs, for example, the falling edge of $\overline{RAS}$ (row address strobe) was used to clock in the row address and the falling edge of $\overline{CAS}$ was used to clock in the column address. The outputs of the row address hold register (column address hold register) are decoded and used to select a row (column) line, Fig. 16.39.

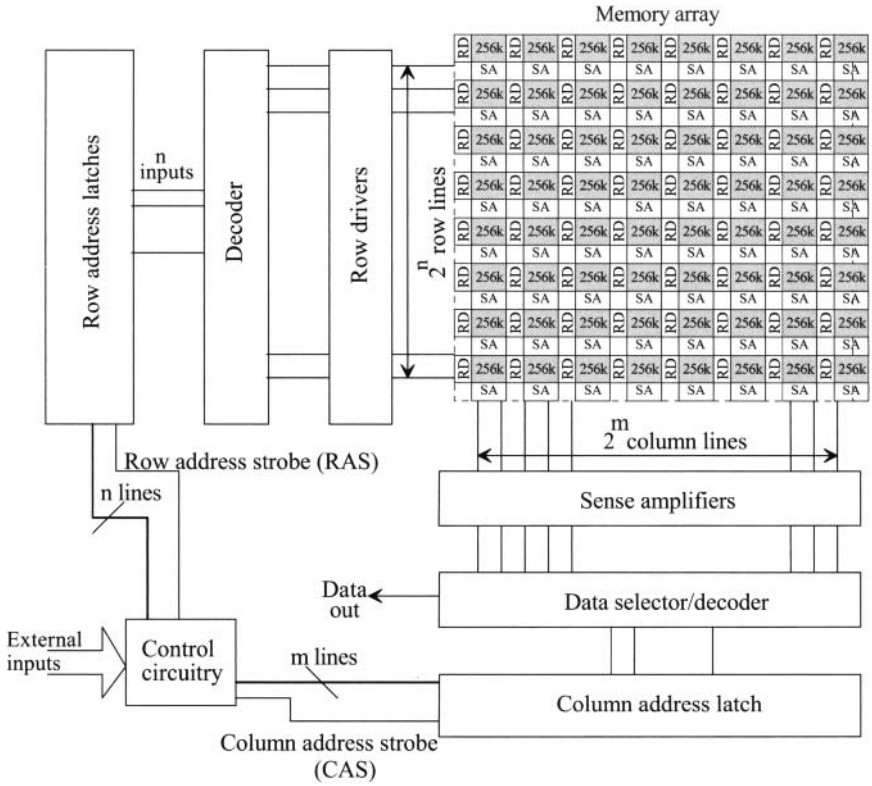


Figure 16.39 Detailed block diagram of a RAM.

The length of the column (m-bits in the Fig. 16.39) and row (n-bits in Fig. 16.39) address words depends on the size of the memory and the word size. For example, a 1-Gbit memory can be organized in x1 (by one), x2 (by two), x4 (by four), etc., configuration. The "by four" for example, simply signifies the word size used in the memory's input/output data path is 4-bits. A 1-Gbit memory in a x4 configuration simply indicates that 256-Mwords can be addressed where each word is 4-bits. In a x1 configuration 1-Gbits of data can be accessed (2^{30} = 1-G = 1,073,741,824). We would then have 15 address pins for 2^{15} different row and column addresses (remembering that the address pins are shared). The next question then becomes how to decode the addresses and select the appropriate row(s) and column(s).

Global and Local Decoders

Looking at Fig. 16.39, we see that we decode the outputs of the row address latches and the column address latches and then feed the outputs to the subarrays. This is called a *global decode*. If the memory size is, once again, 1-Gbit (in a x1 configuration), then 2^{15} (= 32,768) row line wires and 2^{15} column line wires are fed to the memory arrays. If each memory array is 256-kbits (512 word lines and 512 column lines), then we need 4,096 arrays to get a 1-Gbit memory (64 by 64 memory arrays in Fig. 16.39). When the output of the row decoder goes high, it turns on a word line in the 64 memory arrays of the addressed row. A total of 32k columns then have data sitting on them. Because our memory is going to take only 1 bit of data and feed it to the output, we have, perhaps, wasted a lot of power. (In many topologies this large amount of data is called a "page." Once the row is opened it can be quickly read out of the memory by simply changing the column address.) The other issue with the global decoder is that a separate layer of metal is needed to route the decoded signals throughout the entire chip (adding process complexity). The big benefit of global decoding is reduced chip size.

A *local decoder* takes the input addresses and, as the name says, decodes them locally at the memory array. The 15 bits, for the example given above, are routed to each subarray so that they can be decoded locally. This takes up a considerable amount of space on the chip (having a decoder at each 256k array, for example) but doesn't call for increased process complexity. In most practical designs a combination of local and global techniques are used. Some of the address bits are decoded globally while others are decoded locally. The global decoder enables an array or (arrays), while the local decoder selects the array's row line and column line. A static decoder is seen in Fig. 16.40. Figure 16.41 shows how the 10-bit address (for 1,024 word or row lines) is connected to each decoder element.

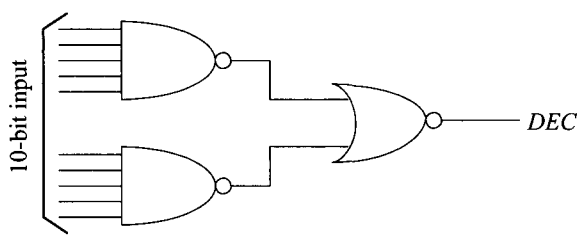


Figure 16.40 A static decoder.

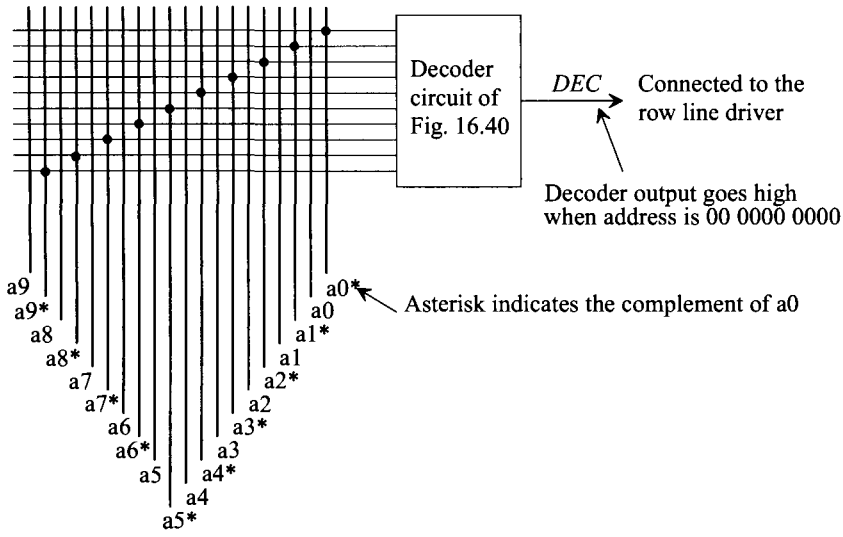


Figure 16.41 How the address lines are connected to a decoder element.

For the column decoder, we know that we have to be able to write or read data to the memory cells (not just select a column). For this reason pass transistors are used on the output of the decoder, as seen in Fig. 16.42. The pass transistor doesn't pass a logic one to full levels. We lose a V_{THN} with body effect when writing or reading a one. The sense amplifier may be used to pull the bit line up to a full logic one. Similarly, on the output of the column decoder a sense amplifier may be used to pull the output line to full logic levels. Finally, note that we drew the column decoder and its outputs going horizontally. In practice, the outputs of the column decoder run in the same direction as the columns themselves in order to save space.

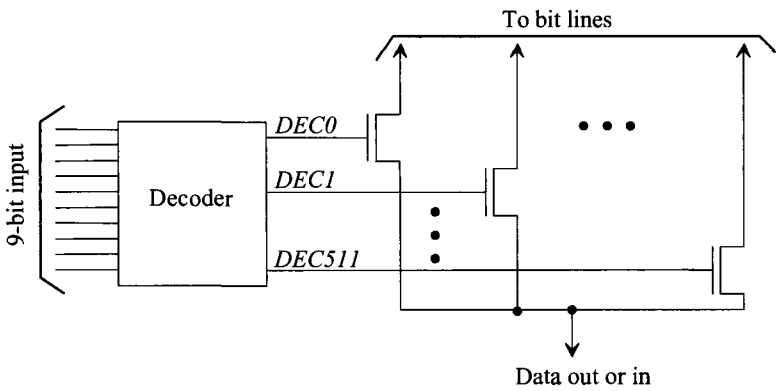


Figure 16.42 Addition of pass transistors to implement a column decoder.

Reducing Decoder Layout Area

To reduce the layout area of address decoders, a pass-transistor based decoder, see Fig. 16.43a, can be used. The concerns with using this decoder are that the unselected outputs are not actively driven high or low (they are floating) and, once again, as NMOS devices they don't pass a logic one to a full logic level (that is, *VDD*). This means that on the output of the decoder, as part of our row driver, we need a circuit that, when not selected, pulls the output of the decoder low. At the same time, to maximize the noise margins, the switching point of the driver needs to be reduced. For example, with $VDD = 1$ V and $V_{THN} = 0.35$ V (high because of body effect), the selected output of the decoder swings from 0 to 0.65 V. The switching point of the row driver should lie in the middle of this swing at 0.3125 (ideally), to maximize the noise margins, Fig. 16.43b.

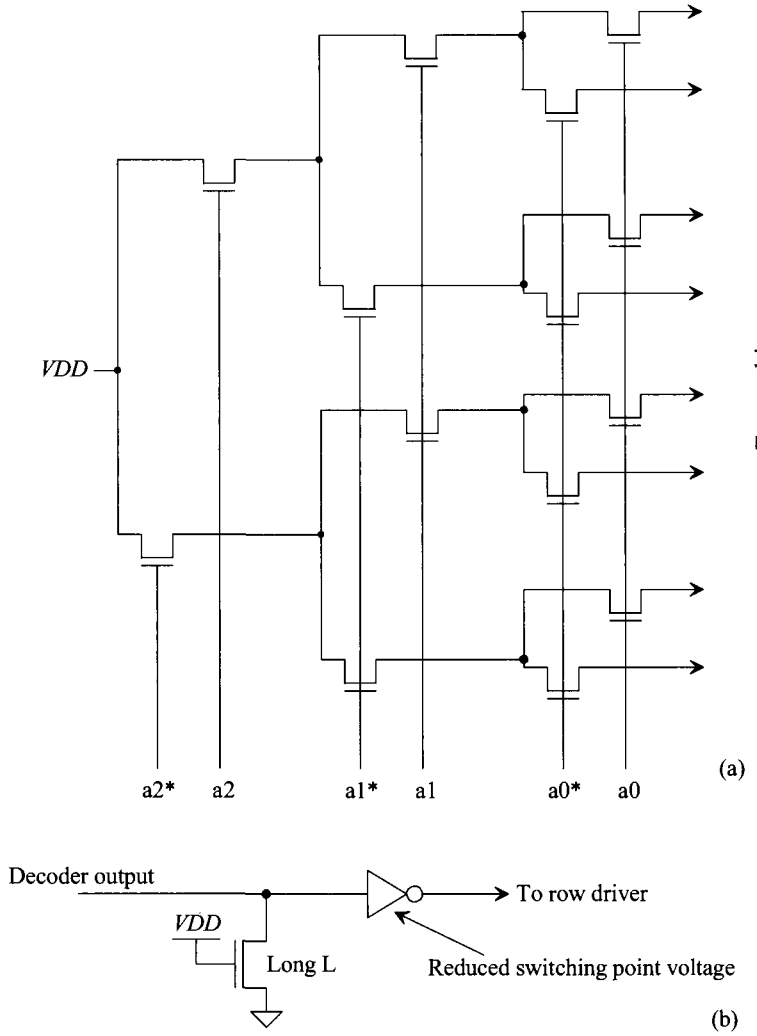


Figure 16.43 A tree decoder used to reduce layout area.

Another technique to reduce decoder layout size uses the precharge-evaluate (PE) logic, Fig. 16.44, discussed in Ch. 14. This figure shows a 3-bit input decoder. The output of the circuit goes high when the input address is 000. Using the long-length keeper MOSFET makes the dynamic circuit operate as a static circuit. Variations of this circuit are common in commercially available memories.

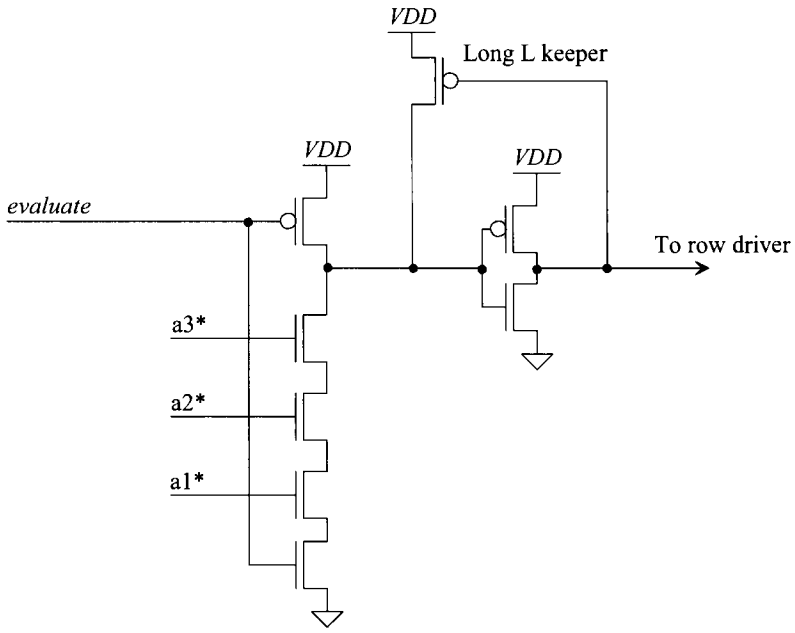


Figure 16.44 3-bit PE decoder.

16.2.3 Row Drivers

In most memories a single NMOS pass transistor switches data into or out of the memory cell. To fully turn this pass transistor on, the row line is driven to a pumped voltage (a voltage outside of the power supply rails that is generated using the on-chip charge pump circuit discussed in Ch. 18). We'll call this pumped voltage *VDDP*. For our purposes this voltage must be greater than $VDD + V_{THN}$ (with body effect). In the following discussion we'll assume a *VDDP* of 1.5 V (since our *VDD* is 1 V and V_{THN} is 280 mV without body effect).

Examine the inverter circuit seen in Fig. 16.45. The input to the inverter connected to the word line swings from 0 to *VDD* (1 V). Note that the PMOS is sitting in its own well. Both its source and body are tied to *VDDP*. When the input to this inverter is low (0 V), the PMOS is on and the NMOS is off. The word line is driven to *VDDP* (which is what we want). Unfortunately when this inverter input is high (= *VDD*), the NMOS turns on but the PMOS can't shut off. The source gate voltage of the PMOS is 0.5 V (above the PMOS's threshold voltage). In this section we discuss word-line drivers, knowing that a simple inverter can't be used.

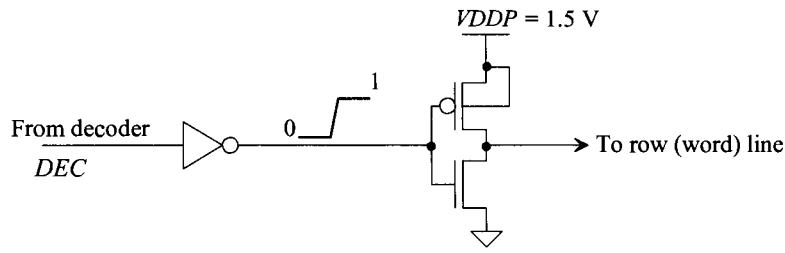


Figure 16.45 Problems with using an inverter for a row driver.

A row driver based on the inverter is seen in Fig. 16.46. When the decoder output is low, M1 is off and M2 is on. This causes M3 to turn on, driving the gate of M4 to *VDDP* so that it shuts off. When the decoder output goes high, M1 turns on and M2 shuts off. The gate of M4 is pulled to ground turning it on. This causes the word line to move to *VDDP* and M3 to shut off. This circuit works well but there can be a significant contention current when M3/M4 are turning on. To avoid this, the pumped voltage can be a clocked signal that goes high *after* the output of the decoder has transitioned. The idea is seen in Fig. 16.47.

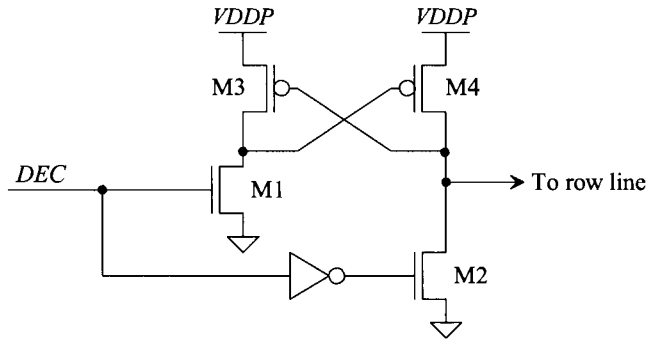


Figure 16.46 A CMOS word line driver.

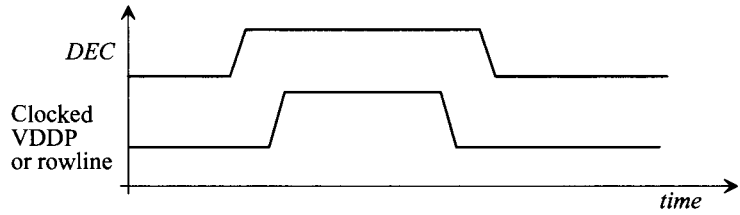


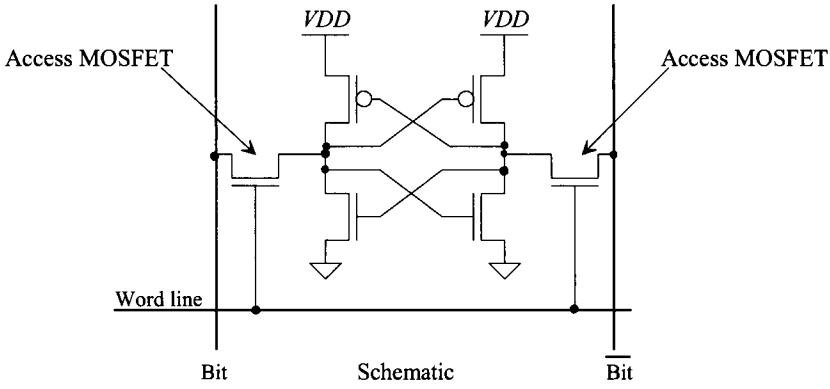
Figure 16.47 Using a clocked VDDP to reduce contention current in a row line driver.

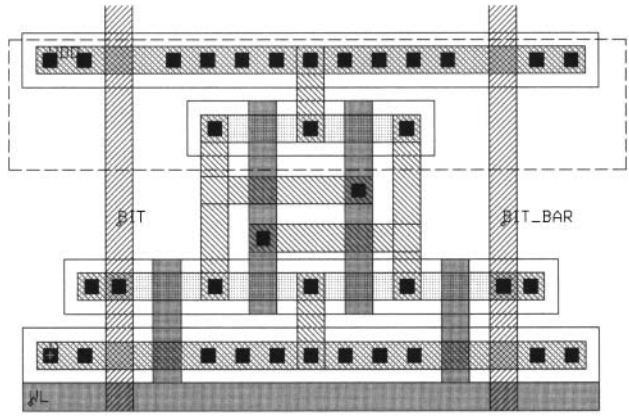
16.3 Memory Cells

We've already looked at the DRAM memory cell in detail. In this section we'll take a look at the static RAM cell (SRAM), the erasable programmable read-only memory (EPROM) cell, the electrically erasable programmable read-only memory (EEPROM) cell, and the Flash memory cell.

16.3.1 The SRAM Cell

The schematic and layout of a six-transistor SRAM memory cell is seen in Fig. 16.48. This is, as its name implies, static, meaning that as long as power is applied to the cell it will remember its contents (unlike the DRAM cell, which loses its memory after a short time). The basic cross-coupled inverter latch should be recognized in the topology. To access the cell, the word line goes high and turns on the access MOSFETs. The bit lines are driven in complementary directions with a "strong" driver for writing. When the word line goes low, the datum is latched in the cell.





Layout

Figure 16.48 The six-transistor SRAM memory cell.

The sizing of the access MOSFETs is important. If they are too weak, the bit lines won't be able to flip the state of the cell when writing. If they are too strong, the layout area can be large (noting that bit lines are often precharged high in an SRAM so that the access devices are initially off during sensing). To reduce the layout area, a cell that doesn't use PMOS devices can be used, Fig. 16.49. The cell size is decreased by eliminating the n-well and using high resistance poly n+/p+ resistors. The layout of the polysilicon resistor is also seen in Fig. 16.49. The resistor can be thought of as a leaky bipolar transistor. Typical resistance values approximately 10 MΩ (or more). The CMOS SRAM cell dissipates little static power, while the resistor/n-channel SRAM cell dissipates VDD^2/R_{poly}.

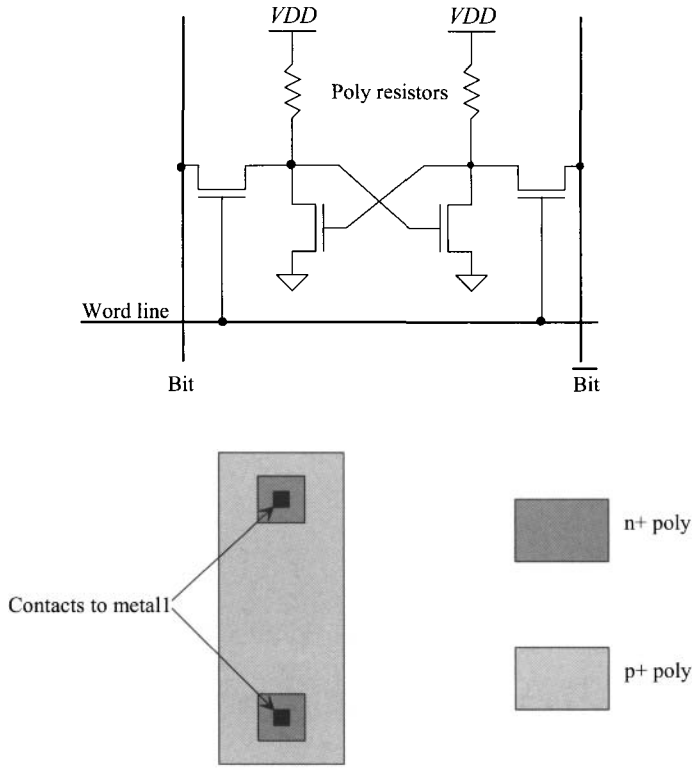


Figure 16.49 SRAM memory cell with poly resistors.

16.3.2 Read-Only Memory (ROM)

ROM is the simplest semiconductor memory. It is used primarily to store instructions or constants in a digital system. The basic operation of a ROM can be explained with the ROM memory schematic shown in Fig. 16.50. Remembering that only one word line (row line) can be high at a time, we see that R_1 going high causes the column lines C_1, C_2, and C_4 to be pulled low. Column lines C_3 and C_5 are pulled high through the long L

MOSFET loads at the top of the array. If the information that is to be stored in the ROM memory is not known prior to fabrication, the memory array is fabricated with an n-channel MOSFET at every intersection of a row and column line (Fig. 16.51a). The ROM is programmed (PROM) by cutting (or never fabricating) the connection between the drain of the MOSFET and the column line Fig. (16.51b). Because it is not easy to program ROM, it is limited to applications where it is mass produced.

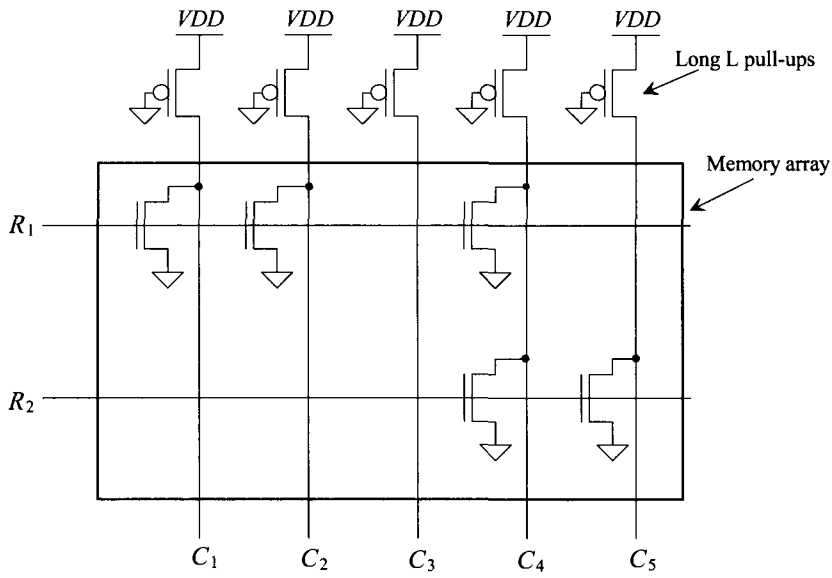


Figure 16.50 A ROM memory array.

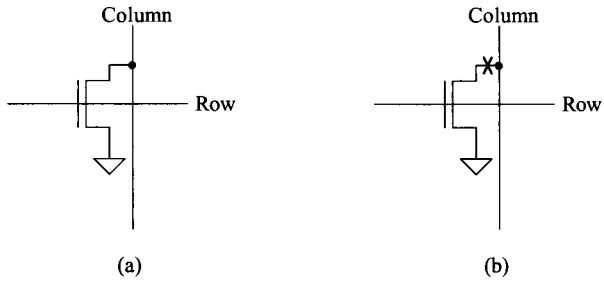


Figure 16.51 (a) n-channel MOSFET at the intersection of every column and row line and (b) eliminating the connection between the drain and column line to program the ROM.

16.3.3 Floating Gate Memory

A MOSFET made with two layers of poly is seen in Fig. 16.52 (along with its schematic symbol and a typical layout). As seen in the figure, the poly1 is floating, that is, not electrically connected to anything. A dielectric surrounds this floating island of poly1. With the gate oxide on the bottom, a thin oxide insulates the MOSFET from the poly2 (word/row line) above. Poly2 is the *controlling gate* of the transistor (the terminal we drive to turn the MOSFET on). We are going to make a memory element out of this cell by changing, adding or removing, the charge stored on the floating gate.

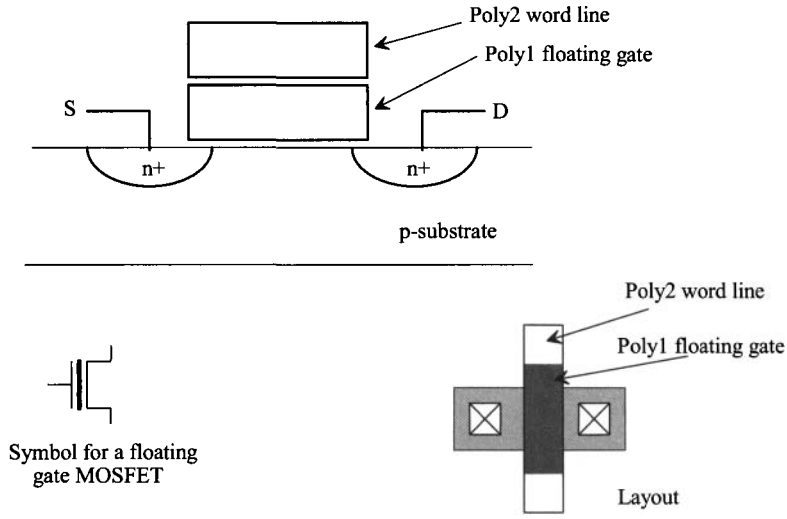


Figure 16.52 A floating gate MOSFET, its symbol and layout.

Figure 16.53 shows the difference between the *erased* state and the *programmed* state in a floating gate memory. The erased state, the state of the memory when it is fabricated (that is, before we force charge onto the floating gate), shows normal MOSFET behavior. Above the threshold voltage, the device turns on and conducts a current. When the cell is programmed, we force a negative charge (electrons) onto the floating gate (how we force this charge will be discussed shortly). This negative charge attracts a positive charge beneath the gate oxide. The result is that a larger controlling gate voltage must be applied to turn the device on (the threshold voltage increases).

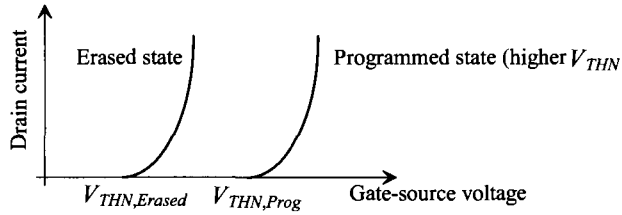


Figure 16.53 Programmed and erased states of a floating gate memory.

The Threshold Voltage

From Ch. 6 we can write the threshold voltage of a single-poly gate MOSFET as

$$V_{THN} = -V_{ms} - 2V_{fp} + \frac{Q'_{b0}}{C'_{ox}} \tag{16.10}$$

where we haven't included the shifts due to the threshold voltage implant, Q'_c, or any unwanted surface state charge, Q'_{ss} (to keep the equations shorter). Looking at Fig. 16.54, we see that the effective oxide capacitance from the controlling gate to the channel has decreased from C'_{ox} for a single-poly gate MOSFET to $C'_{ox}/2$ for a dual poly gate MOSFET. Our threshold voltage, for a floating gate MOSFET, can then be written as

$$V_{THN,Erased} = -V_{ms} - 2V_{fp} + 2 \cdot \frac{Q'_{b0}}{C'_{ox}} \tag{16.11}$$

If the term Q'_{b0}/C'_{ox} is approximately 50 mV (a typical oxide thickness used in a floating gate memory is 100 Å), then the erased threshold voltage, $V_{THN,Erased}$ (see Fig. 16.53) is only 50 mV larger than the V_{THN} of a normal (single poly gate) MOSFET.

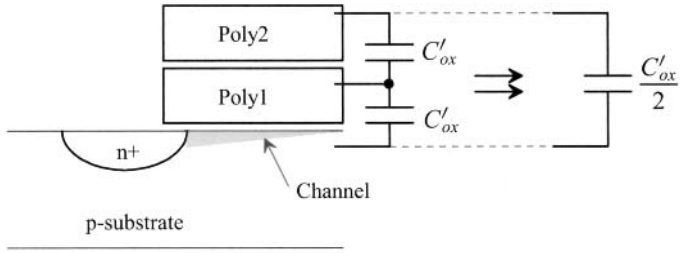


Figure 16.54 Oxide capacitance estimation for calculating threshold voltage..

Next consider what happens if we trap a negative charge on the floating poly1 gate, Fig. 16.55. The threshold voltage with this trapped charge, Q'_{poly1}, is shifted to

$$V_{THN,Prog} = -V_{ms} - 2V_{fp} + 2 \cdot \left(\frac{Q'_{b0}}{C'_{ox}} + \frac{|Q'_{poly1}|}{C'_{ox}} \right) \tag{16.12}$$

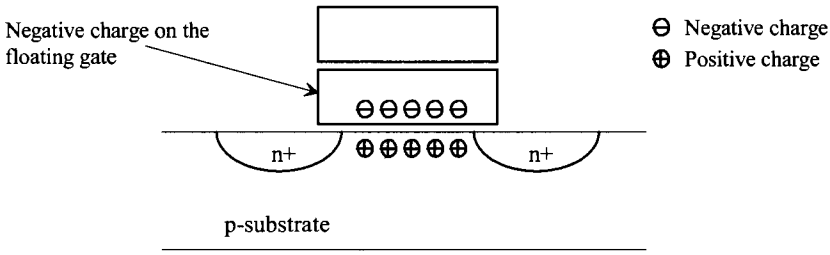


Figure 16.55 Trapped negative charge on the floating gate.

Erasable Programmable Read-Only Memory

Erasable programmable ROM (EPROM) was the first floating gate memory that could be programmed electrically. To erase (return the cell to its fabricated state, that is, leave no trapped charge on the floating gate), the cell is exposed to ultra-violet light through a quartz window in the top of the chip's package. The ultra-violet light increases the conductivity of the silicon-dioxide surrounding the floating gate and allows the trapped charge to leak off. The inability to erase the EPROM electrically has resulted in its being replaced by Flash memory (discussed later).

Programming the EPROM relies on *channel hot-electron* (CHE) injection. CHE is accomplished by driving the gate and the drain of the MOSFET to high voltages (say a pumped voltage of 25 V), Fig. 16.56. The high voltage on the drain of the device causes hot electrons (those with significant kinetic energy) to flow in the channel. A large positive potential applied to the gate attracts some of these electrons to the floating gate. The electrons can penetrate the potential barrier between the floating gate and channel because of their large energies.

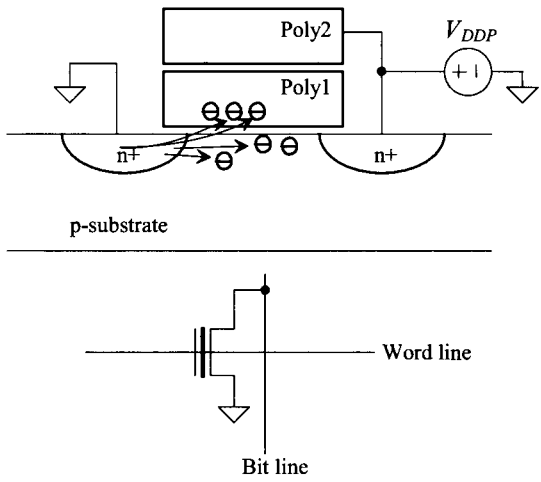


Figure 16.56 How charge is trapped on the floating gate using channel hot electron (CHE) injection.

Two Important Notes

When we are programming the floating gate device, the accumulation of electrons on the floating gate causes an increase in the device's threshold voltage. The more electrons that are trapped the higher the threshold voltage. This increase in threshold voltage causes the drain current to decrease. The decrease in drain current then reduces the rate at which the electrons are trapped on the floating gate oxide. If we apply the programming voltages for a long period of time, the drain current drops to zero (or practically a small value). Because of this feedback mechanism, the programming is said to be *self-limiting*. We simply apply the high voltages for a long enough time to ensure that the selected devices are programmed.

Next examine Fig. 16.57. When we are programming a row of cells, we drive the word line to a high voltage. If the cell is to remain erased, we simply leave the corresponding bit line at ground. If the cell is to be programmed, we drive the bit line to the high voltage. For CHE, both the drain and gate terminals of the floating device must be at a high voltage. This is important because it removes the need for a select transistor (something we'll have to use in other programming methods to keep from programming an unselected cell).

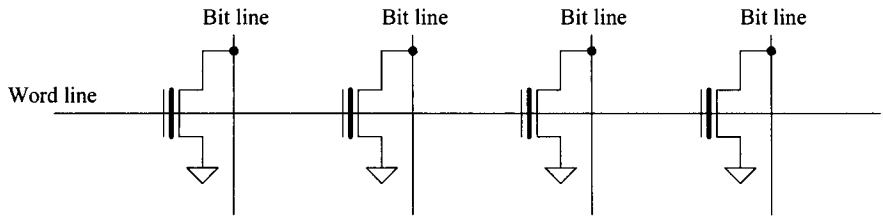


Figure 16.57 Row of floating gate devices. When programming the word line is driven to a high voltage. To program a specific cell, its bit line is also driven to a high voltage. To leave the cell erased, the bit line is held at ground.

Flash Memory

By reducing the thickness of the oxide from, say, 300 Å (a typical value used in an EPROM) to 100 Å, Fowler-Nordheim tunneling (FNT) can be used to program or erase the memory cell. Floating gate memory that can be both electrically erased and programmed is called electrically erasable programmable ROM (EEPROM). Note that this name is an oxymoron. If we can electrically *write* to the memory, then it isn't a *read*-only memory. For this reason, and because the rows of memory are generally erased in a *flash* (that is, large amounts of memory, say a memory array, are erased simultaneously and, when compared to the EPROM method of removing the chips from the system and exposing to ultra-violet light to erase, very quickly), we call floating gate memory that can be electrically programmed and erased *Flash* memory.

While CHE and FNT can be used together (CHE for the programming and FNT for erasing) to implement a memory technology, we assume FNT is used for both programming and erasing in the remaining discussion in this chapter.

Figure 16.58 shows the basic idea of using FNT to *program* a device (recall that this means to trap electrons on the floating gate so that the devices threshold voltage increases). The control gate (poly2) is driven to a large positive voltage. For a 100 Å gate oxide this voltage is somewhere between 15 and 20 V. Note that we are assuming that our NMOS devices are sitting in a p-well that is sitting in an n-well (so we can adjust the p-well [body of the NMOS] potential). The electrons tunnel through the thin oxide via FNT and accumulate on the floating gate. Like programming in an EPROM device, this mechanism is self-limiting. As electrons accumulate on the floating gate, the amount of tunneling current falls. Note that if we didn't want to program the device when poly2 is at 20 V we could hold the drain at a higher voltage (not ground) to reduce the potential across the thin gate oxide (aka tunnel oxide).

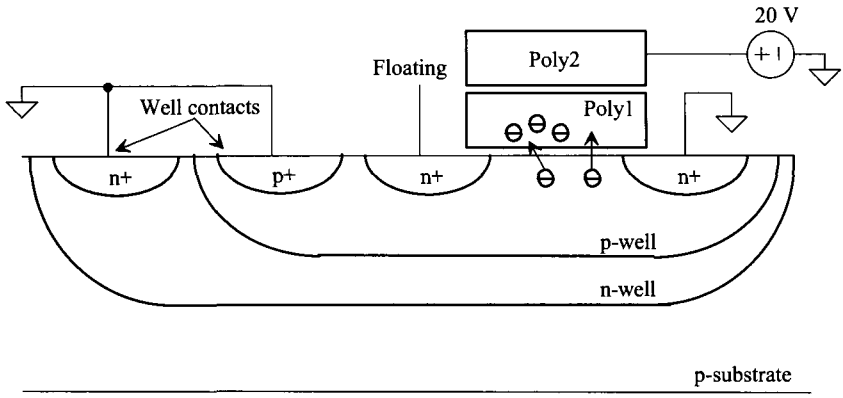


Figure 16.58 FNT of electrons from the p-well to a floating gate to increase threshold voltage (showing programming).

To *erase* the device using FNT, examine Fig. 16.59. Both the p-well and the n-well are driven to 20 V while the control gate (poly2) is grounded. Electrons tunnel via FNT off of the floating gate (poly1) to the p-well. The source and drain contacts to the device are floating (to accomplish this we will float the bit line and the source n+ outside of the array). Again, the movement of charge is self-limiting (however, there are device issues that can result in over erasing). The tunnel current drops as positive charge accumulates on the floating gate. If the erasing time is long, a significant amount of positive charge can accumulate on the floating gate. This will *decrease* the threshold voltage of the MOSFET. Figure 16.60 shows the programmed and erase states for a Flash memory where a positive threshold voltage indicates that the device is programmed and a negative threshold voltage indicates that the device is erased (we show ± 3 *V* as typical values).

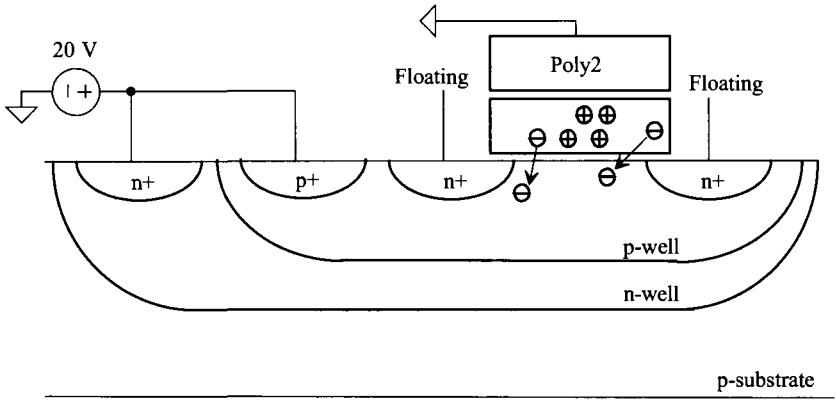


Figure 16.59 FNT of electrons from the floating gate to p-well to decrease threshold voltage (showing erasing).

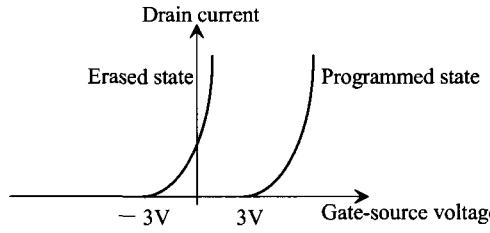


Figure 16.60 Programmed and erased states of a flash memory.

The schematic and layout of a 4-bit NAND Flash memory cell is seen in Fig. 16.61. The select transistors are made using single poly (normal) MOSFETs. When the cell (all four bits) is erased, the p-well and the n-well are driven to 20 V external to the memory array via the p+ implant. The bit lines and the n+ source connection at the bottom of the layout are floated. The four control gates and the two select MOSFET gates are pulled to ground (so all six poly gates, aka, word lines, are at ground for an erase).

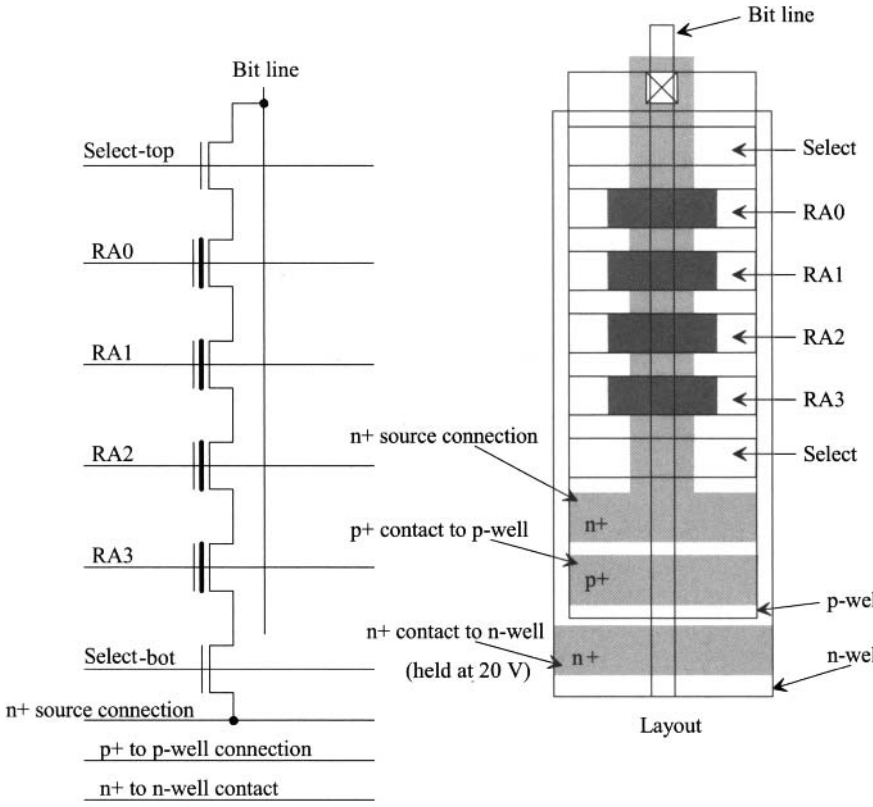


Figure 16.61 A 4-bit NAND Flash memory cell.

To illustrate programming the floating gate MOSFET connected to RA0 (row address 0), examine the connections to the NAND cell seen in Fig. 16.62. The bit line is driven to ground. A voltage, say 20 V, is applied to the gate of the top select gate. The gates of the floating gate MOSFETs connected to RA1–RA3 are driven to 5V. This 5 V signal turns on these devices but isn't so big that FNT will occur in them. The bottom select MOSFET remains off so that there is no DC path from the bit line to ground, Fig. 16.58. The p-well (which is common to all of the cells in the memory array, that is, not just the four in the memory cell) is pulled to ground external to the memory array via the p+ implant. Because the gate, RA0, is pulled to 20 V, as seen in Fig. 16.58, and the drain implant is pulled to ground through the bit line, the device will be programmed (electrons will tunnel through the oxide and accumulate on the floating gate).

The next thing we need to look at before talking about reading the cell is how we keep from programming the adjacent floating gate devices, those also connected to *RA*0, if they are to remain erased. What we need to do is ensure that no FNT occurs in these unselected devices. Figure 16.63 shows how we keep from programming a device. The bit line of the cell that is to remain erased is driven to a voltage that is large enough to keep FNT from occurring. The bottom select MOSFET is off, so there won't be a DC path from the bit line to ground (this is important because all other MOSFETs in the memory cell will be on).

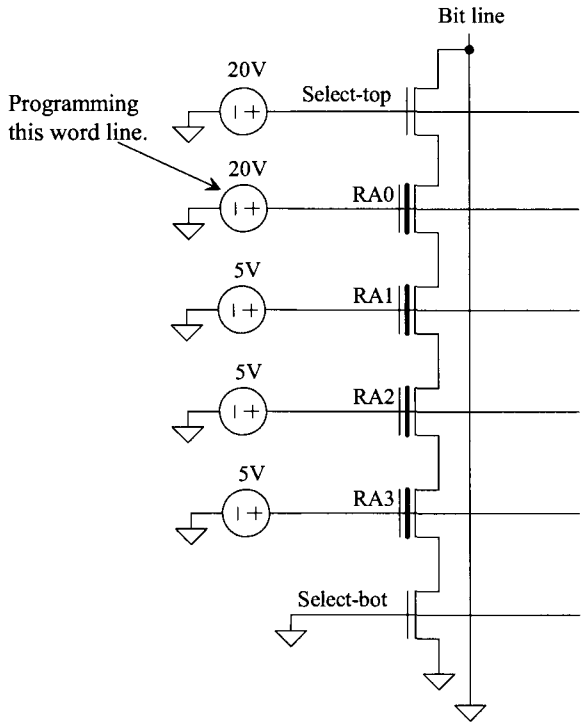


Figure 16.62 Programming in a Flash NAND cell.

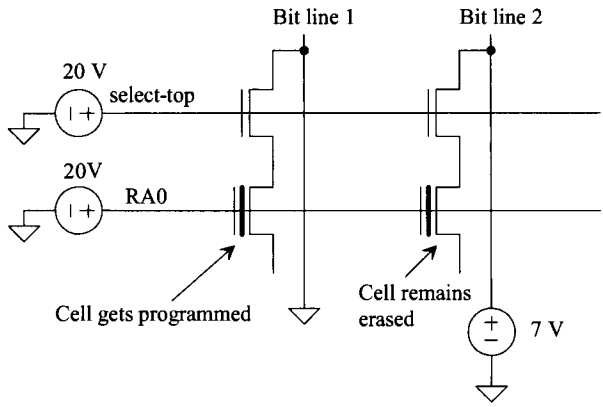


Figure 16.63 How cells are programmed (or not) in a NAND Flash memory array.

To understand reading a NAND Flash cell, consider the expanded view of Fig. 16.60 shown in Fig. 16.64. If both select transistors are turned on (say 5 V on their gates), the unselected row lines are driven high (say, again, to 5 V), and the selected row line is held at zero volts, then the current difference between I_{erased} and I_{prog} can be used to determine if the (selected) floating gate MOSFET is erased or programmed. If an average current, that is, $(I_{erased} + I_{prog})/2$ is driven into the bit line, an erased cell will keep the bit line at a low voltage. The selected (erased) MOSFET will want to sink a current of I_{erased} when its gate is zero volts. A programmed cell won't be able to sink this current (it will want to sink a current of I_{prog}) and so the bit line will go high.

Table 16.1 shows a summary of erasing, programming, and reading a NAND Flash memory cell. Notice that in Figs. 16.58 and 16.59 (during either erasing or programming the cell) the source of the MOSFET is floating. In Fig. 16.58 we can float the source by shutting off the bottom select MOSFET in the NAND stack. However, in Fig. 16.59 if we try to isolate the cell by shutting off the select transistors we see an issue, that is, when 20 V is applied to the p-well, the n+ source/drain implants can forward bias. This is why we must float both the bit line and the n+ source external to the array. Also, note that the top select MOSFET is used to provide better isolation between the memory cell and the bitline (it lowers the capacitive loading on the bit line). Going through the

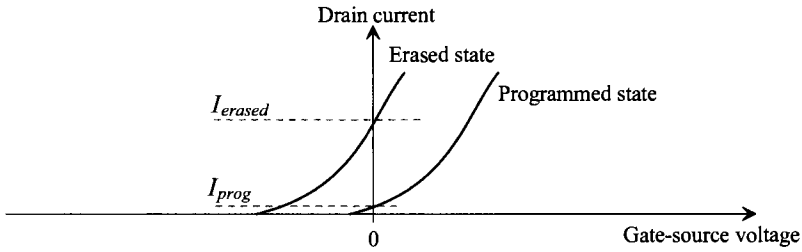


Figure 16.64 Expanded view showing erased and programmed IV curves.

operation of the NAND memory cells, we see that the lower select transistor, from a logical point of view, is all that is needed for basic cell operation.

Table 16.1 Summarizing NAND Flash cell operation

Inputs	Erase	Program	Read
Bit line	Floating	0 V	High or low
select_top	0 V	20 V	5 V
RA0	0 V	20 V	0 V
RA1	0 V	5 V	5 V
RA2	0 V	5 V	5 V
RA3	0 V	5 V	5 V
select_bot	0 V	0 V (so the source of the cell floats)	5 V
n+ source	Floating	0 V	0 V
p+ well tie-down	20 V	0 V	0 V
Comments	Erases entire array since well and word lines are common (Flash). The p-well at 20V will forward bias the n+ source and drain regions. This requires the bit line and n+ source float external to the array.	Programming the RA0 cell. Bit lines of the cells not to be programmed, but on RA0, are driven to 7 V to avoid FNT. Unused word lines driven to 5V.	Reading the contents of RA0. An average current is put into the bit line.

Before leaving this topic, let's use our short-channel process, which, of course, has only a single poly layer to show how direct tunneling gate current varies with terminal voltages. For our 50 nm, short-channel process used in this book (t_{ox} = 14 Å), the gate current density is (roughly) 5 A/cm^2. A 500 nm by 50 nm device will have a gate current of (typically) 50 pA. However, if the potentials of either the drain or source terminals of the MOSFET move further away from the gate's potential, the gate tunnel current will increase. Figure 16.65 shows how the gate current changes in a 10/1 NMOS device (actual width of 500 nm and a length of 50 nm) with the drain and bulk held at ground while the gate voltage is swept from 0 to 2 V. Notice that at a V_{GS} above *VDD* (= 1 V), the gate current becomes much more significant than the 50 pA we calculated for a typical value a moment ago. Figure 16.66 shows that if we hold the drain at 1 V, the gate current is significantly less than the values seen in Fig. 16.65. This is important because we can keep from programming an erased cell (if we were using a floating gate in this technology with oxide thickness of 14 Å) simply by driving its drain (the bit line) to *VDD*. Finally, Fig. 16.67 shows the current when erasing the device. The problem with using direct tunneling for Flash memory is retention. While it is easy to program/erase the floating gate device, it is equally easy for trapped charge to tunnel back off of a floating gate over time.

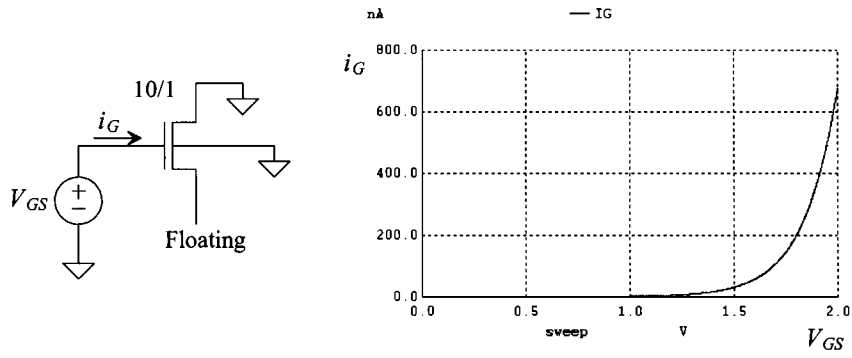


Figure 16.65 How gate current changes with gate voltage with either the gate or source grounded.

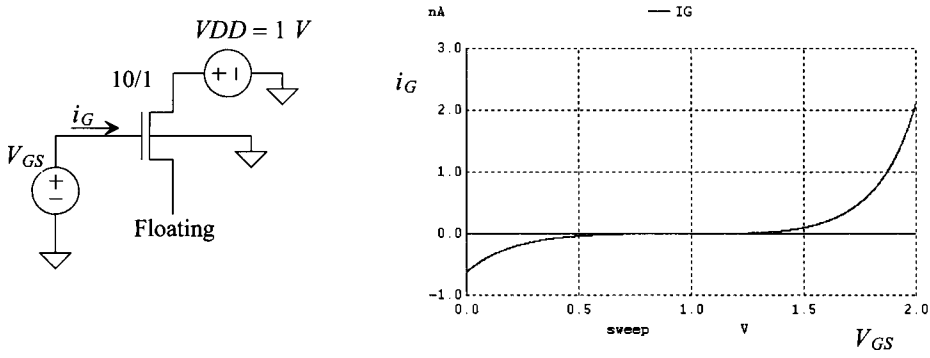


Figure 16.66 How gate current changes with gate voltage when drain is floating and drain is at VDD.

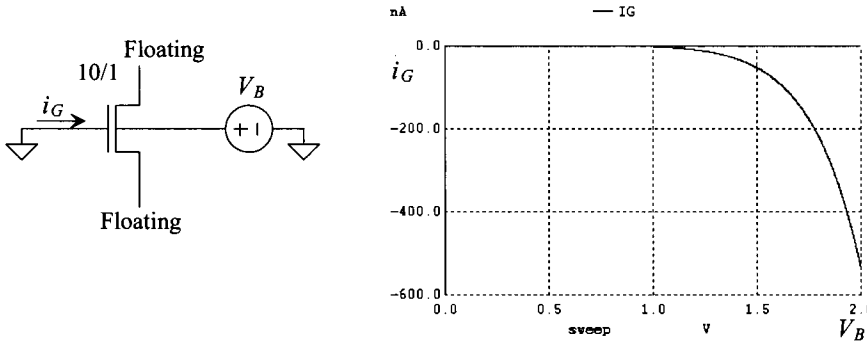


Figure 16.67 Erasing. Since electrons are being removed from the gate (they are tunneling through the gate oxide), we know the gate current will flow out of the device.

An interesting use for floating gate MOSFETs is in the design of analog circuits. Being able to adjust the threshold voltage of a MOSFET can be very useful for low-power or low-voltage design. For example, the input common-mode range of a differential amplifier can be widened by reducing the threshold voltage of the input diff-pair. Also, trimming (removing offsets) can be accomplished when floating gate MOSFETs are used. The interested reader is referred to the references on the following page for additional information.

ADDITIONAL READING

[1] B. Keeth, R. J. Baker, B. Johnson, and F. Lin, *DRAM Circuit Design: Fundamental and High-Speed Topics, Second Edition*, Wiley-IEEE, 2008. ISBN 978-0-470-18475-2

[2] B. Jacob, S. W. Ng, and D. T. Wang, *Memory Systems: Cache, DRAM, Disk*, Morgan Kaufmann, 2008. ISBN 978-0123797513

[3] J. E. Brewer and M. Gill, *Nonvolatile Memory Technologies with Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices*, Wiley-IEEE, 2008. ISBN 978-0471770022

[4] K. Itoh, *VLSI Memory Chip Design*, Springer-Verlag Publishers, 2001. ISBN 3-5406-7820-4

[5] W. D. Brown and J. E. Brewer (eds.), *Nonvolatile Semiconductor Memory Technology: A Comprehensive Guide to Understanding and Using NVSM Devices*, John Wiley and Sons Publishers, 1998. ISBN 0-7803-1173-6

[6] A. K. Sharma, *Semiconductor Memories: Technology, Testing and Reliability*, John Wiley and Sons Publishers, 1997. ISBN 0-7803-1000-4

[7] T. Mohihara, et al., "Disk-Shaped Capacitor Cell for 256Mb Dynamic Random-Acess Memory," *Japan Journal of Applied Physics*, vol. 33, part 1, no. 8, pp. 4570–4575, August 1994.

[8] K. Sagara, et al., "Recessed Memory Array Technology for a Double Cylindrical Stacked Capacitor Cell of 256M DRAM," *IEICE Trans. Electron.*, vol. E75-C, No. 11, pp. 1313–1322, November 1992.

[9] T. Hamada, "A Split-Level Diagonal Bit-Line (SLDB) Stacked Capacitor Cell for 256Mb DRAMs," 1992 *IEDM Technical Digest*, pp. 799–802.

[10] J. H. Ahn et al., "Micro Villus Patterning (MVP) Technology for 256Mb DRAM Stack Cell," 1992 *Symposium on VLSI Technical Digest of Technical Papers*, pp. 12–13.

[11] M. I. Elmasry, *Digital MOS Integrated Circuits II,* IEEE Press, 1992. ISBN 0-87942-275-0

[12] B. Prince, *Semiconductor Memories: A Handbook of Design Manufacture, and Application,* 2nd ed., John Wiley and Sons Publishers, 1991. ISBN 0-471-92465-2

[13] R. D. Pashley and S. K. Lai, "Flash Memories: The Best of Two Worlds," *IEEE Spectrum,* 1989, pp. 30–33.

[14] D. Frohman-Bentchkowsky, "FAMOS-A New Semiconductor Charge Storage Device," *Solid-State Electronics,* vol. 17, pp. 517–529, 1974.

[15] E. H. Snow, "Fowler-Nordheim Tunneling in SiO_2 Films," *Solid-State Communications,* vol. 5, pp. 813–815, 1967.

Additional Readings Covering Analog Applications of Floating Gate Devices

[16] C. T. Charles and R. R. Harrison, "A Floating Gate Common Mode Feedback Circuit for Low Noise Amplifiers," *Proceedings of the Southwest Symposium on Mixed-Signal Design*, Las Vegas, NV, pp. 180–185, February 23–25, 2003.

[17] F. Adil, G. Serrano, and P. Hasler, "Offset Removal Using Floating-Gate Circuits for Mixed-Signal Systems," *Proceedings of the Southwest Symposium on Mixed-Signal Design*, Las Vegas, NV, pp. 190–195, February 23-25, 2003.

[18] R. R. Harrison, J. A. Bragg, P. Hasler, B. A. Minch, and S. P. Deweerth, "A CMOS programmable analog memory-cell array using floating-gate circuits," *IEEE Transactions on Circuits and Systems-II,* vol. 48, no. 1, pp. 4–11, 2001.

[19] F. Munoz, A. Torralba, R. G. Carvajal, J. Tombs, and J. Ramírez Angulo, "Floating-Gate based tunable CMOS low-voltage linear transconductor and its application to HF g_m-C filter design," *IEEE Transactions on Circuits and Systems*, vol. 48, no. 1, January 2001, pp. 106–110.

[20] E. Sánchez-Sinencio and A. G. Andreou, "Low-Voltage/Low-Power Integrated Circuits and Systems: Low-Voltage Mixed-Signal Circuits," *IEEE Press*, 1999. ISBN 0-7803-3446-9

[21] P. M. Furth and H. A. Ommani, "A 500-nW floating-gate amplifier with programmable gain," 41st Midwest Symp. Circuits and Systems, South Bend, IN, August 1998.

[22] J. Ramírez-Angulo, "Ultracompact Low-voltage Analog CMOS Multiplier Using Multiple Input Floating Gate Transistors," 1996 European Solid State Circuits Conference, pp. 99–103.

[23] J. Ramírez-Angulo, S.C. Choi, and G. Gonzalez-Altamirano, "Low-Voltage OTA Architectures Using Multiple Input Floating gate Transistors," *IEEE Transactions on Circuits and Systems*, vol. 42, no. 12, pp. 971–974, November 1995.

[24] C. G. Yu and R. L. Geiger, "Very Low Voltage Operational Amplifiers Using Floating Gate MOS Transistors," *IEEE Symp. Circuits Sys.*, vol. 2, pp. 1152–1155, 1993.

[25] L. R. Carley, "Trimming Analog Circuits Using Floating-Gate Analog MOS Memory," *IEEE Journal of Solid-State Circuits*, vol. SC-24, pp. 1569–1575, 1989.

[26] J. Sweeney and R. L. Geiger, "Very High Precision Analog Trimming Using Floating Gate MOSFETs," *Proceedings of the European Converence on Circuit Theory and Design*, Brighton, United Kingdom, September 1989, pp. 652–655.

PROBLEMS

Unless otherwise indicated use the short-channel CMOS BSIM4 models for all simulations.

16.1 Estimate the bit line capacitance if there are 256 word lines and we include the gate-drain overlap capacitance from each MOSFET, as seen in Fig. 16.68.

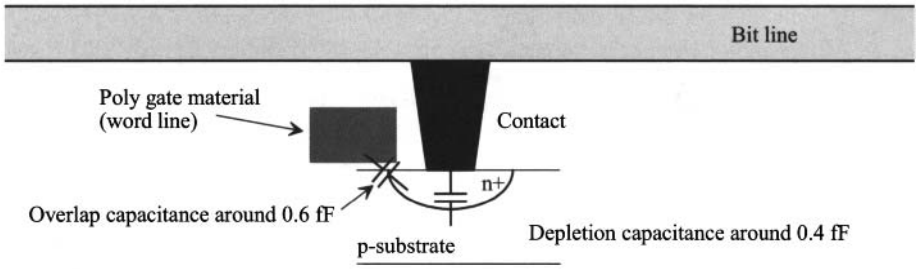


Figure 16.68 How the gate-drain overlap capacitance loads the bit line. Note that this cross-sectional view is rotated 90 degrees from the one seen in Fig. 16.3.

16.2 Consider the NSA seen in Fig. 16.69. Suppose that the load capacitance is mismatched by 20%, as seen in the figure. Assuming that both caps are equilibrated to 0.5 V prior to sensing, which capacitor will fully discharge to ground (which MOSFET will fully turn on)? What voltage difference on the capacitors is needed to cause metastability? Verify your answers with SPICE.

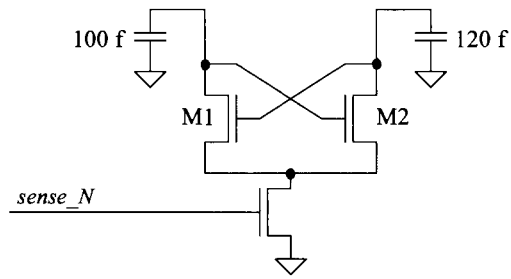


Figure 16.69 Problem 16.2 showing how a mismatch in the load capacitance of an NSA can result in in sensing errors.

16.3 Repeat Problem 16.2 with the circuit in Fig. 16.70. In this figure M1 and M2 experience a threshold voltage mismatch (modeled by the DC voltage source seen in the figure).

16.4 Examining Fig. 16.17 we see that there will be a voltage drop along the metal line labeled *NLAT* when the sense amplifiers fire. Re-sketch this metal line as resistors between each NSA. If the voltage drop along the line is significant (the length of

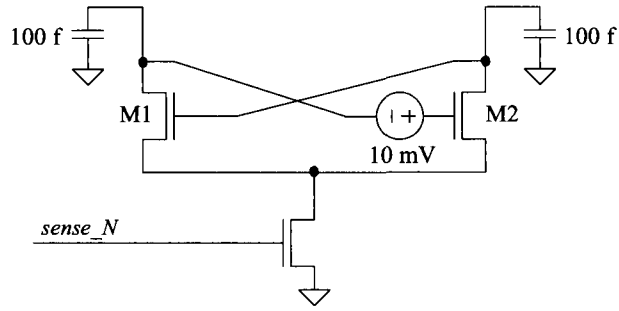


Figure 16.70 Circuit for Problem 16.3.

the line is very long), errors can result. Would we make things better by using individual MOSFETs on the bottom of each NSA connected to *sense_N*? Why or why not?

16.5 Suppose a memory array has 1024 columns. If the word line resistance of one cell is 2 Ω and the capacitance per cell (to ground) is 500 aF, estimate the delay to open a row line. Sketch the equivalent RC circuit of the word line.

16.6 In Fig. 16.71, if the top plate of capacitor C_1 is initially charged to V_1 and the top plate of capacitor C_2 is initially charged to V_2, estimate the final voltage, V_{final}, on the top plates of the capacitors after the switch closes.

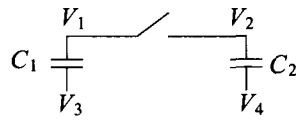


Figure 16.71 Circuit for Problem 16.6.

16.7 Figure 16.72 shows a clocked comparator topology based on the topology seen in Fig. 16.32. The location of the imbalance MOSFETs has been moved in this figure. Comment on the merits and disadvantages of this topology compared to the one in Fig. 16.32. Simulate the operation of this circuit.

16.8 Figure 16.73 shows the addition of I/O (input/output) transistors to the memory array and NSA seen in Fig. 16.17. Sketch a column decoder design using static logic. Show how the 3-bit input address is connected to each stage.

16.9 The I/O lines in the previous problem won't swing to full logic levels. To restore a full *VDD* level on these lines and to speed up the signals, a *helper flip-flop* is used. Sketch a possible implementation of the helper flip-flop. Why can it be used to speed up the signals?

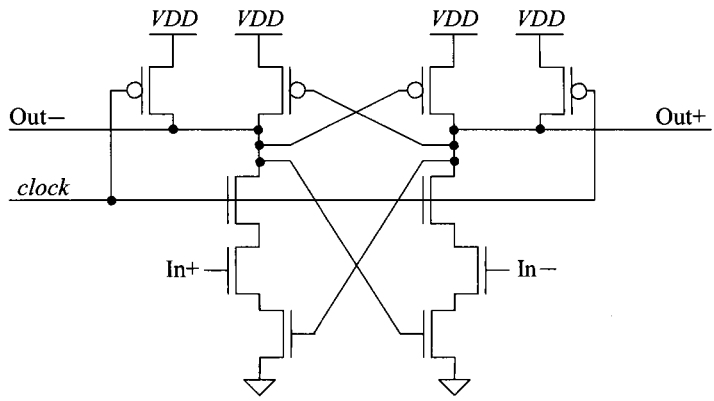


Figure 16.72 An alternative clocked sense amplifier.

16.10 Suppose a 2Mbit memory is to be designed as a x2 part (2-bit input/output words). Further suppose that 10 address pins are available to access the memory. Sketch a block diagram of how the row and column addresses are multiplexed together and stored in separate registers. Use $\overline{RAS}$ and $\overline{CAS}$, as discussed in the chapter to clock in the addresses. Comment on the validity of using 20-bits to access 2Mbits of data.

16.11 Suppose that the memory in Problem 16.10 is made up of 8-256k memory arrays (assume 1024 row lines, a folded array, and 512 column lines). How many I/O lines are needed for each array? If three bits of the address are used to select one of the memory arrays (so that only a single row is open in a single memory array

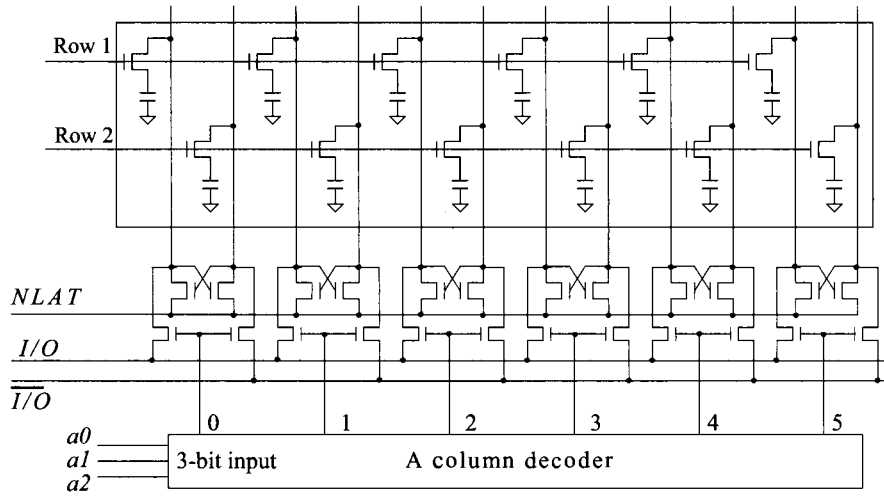


Figure 16.73 Design of a column decoder for problem 16.8.

at a time), how big is a page of data? Sketch a block diagram of a possible decoding scheme for the chip. Assume that only the three bits of the address are globally decoded and that the remaining 17 bits are locally decoded. How many of these bits must be routed to each array assuming an enable (one for each of the eight memory arrays) signal from the global decoder is routed to each memory array.

16.12 Suppose that it is suggested that the word line driver in Fig. 16.46 be modified to simplify it as seen in Fig. 16.74. What is the problem with this design? Use SPICE to illustrate the driver not functioning properly.

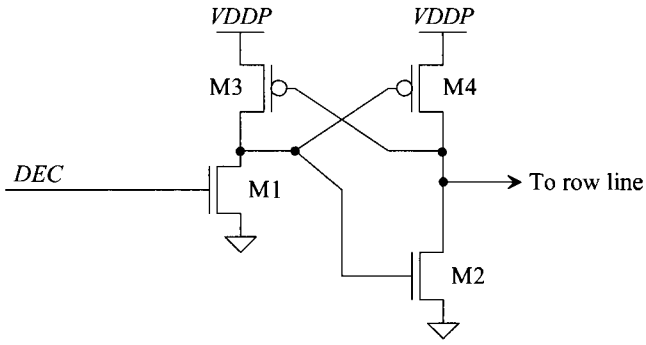


Figure 16.74 A (bad) CMOS word line driver, Problem 16.12.

16.13 Simulate the operation of the SRAM cell seen in Fig. 16.48. Use the 50 nm process with NMOS of 10/1 and PMOS of 20/1. Is it wise for the access MOSFETs to be the same size as the latch MOSFETs? Why or why not? Use simulations to verify your answers.

16.14 Suppose an array of EPROM cells, see Fig. 16.57, consisting of two row lines and four bit lines is designed. Further assume $V_{THN,Erased} = 1\ V$ and $V_{THN,Prog} = 4\ V$. Will there be any problems reading the memory out of the array if an unused row line is grounded while the accessed row line is driven to 5 V ? Explain why or why not.

16.15 Suppose that it is suggested that the n-well in a NAND Flash memory cell should be grounded at all times except when the array is being erased. Is this OK? What would be a potential benefit?

16.16 Explain in your own words and with the help of pictures why a top select MOSFET is needed in a NAND Flash memory cell.

16.17 Suppose that the transistor connected to RA3 in Fig. 16.61 is to be programmed. Further suppose that the n+ source implant seen in the layout is to remain grounded as indicated in Table 16.1. How do we float the source of the transistor connected to RA3 so that it can be programmed as seen in Fig. 16.58?

16.18 Reviewing Fig. 16.62, is it necessary that the gates of the floating gate MOSFETs connected to RA1 – RA3 be driven to 5 V? Can the gates of these MOSFETs remain grounded? Why or why not? If the RA1 MOSFET is to be programmed, in this figure, must the gate of RA0 be driven to 5 V?

16.19 If the I_{erased} of a NAND Flash memory cell is 20 μA and the I_{prog} is 2 μA, explain what the bit line voltage will do when reading out the cell in the following configuration, Fig. 16.75. Will the bit line go all the way to *VDD*? All the way to ground? Explain. If the bit line capacitance is 200 fF, estimate the length of time it will take the data on the bit line to settle before it can be read out. Assume that the *VDD* is 5 V and the bit line is equilibrated to 2.5 V prior to sensing.

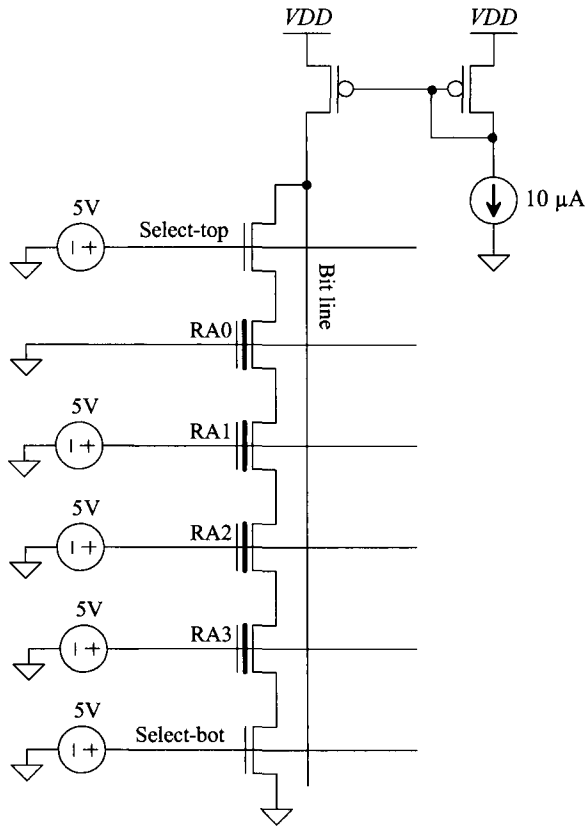


Figure 16.75 Reading the contents of RA0 in a NAND Flash cell.

16.20 Show, using a configuration similar to the one seen in Fig. 16.66, if it is possible to get negative gate tunnel gate current (used for erasing) by grounding the gate and raising the potential on the drain of the MOSFET. Explain why it works or why it doesn't work.