VLSI Design EE 523 Spring 2025

Shahid Masud

Lecture 27

Topics for lecture 27

- **ROM DESIGN**
- **Floating Gate**
- **FN Tunneling**

- **Example Design: Crosspoint Switch, Barrel Shifter**

Lab and Quiz in Last Lecture

ROM CELL CONSTRUCTION – NON-VOLATILE MEMORY TECHNOLOGY

Non-Volatile Read-Write Memory

Non-volatile Read-Write Memories

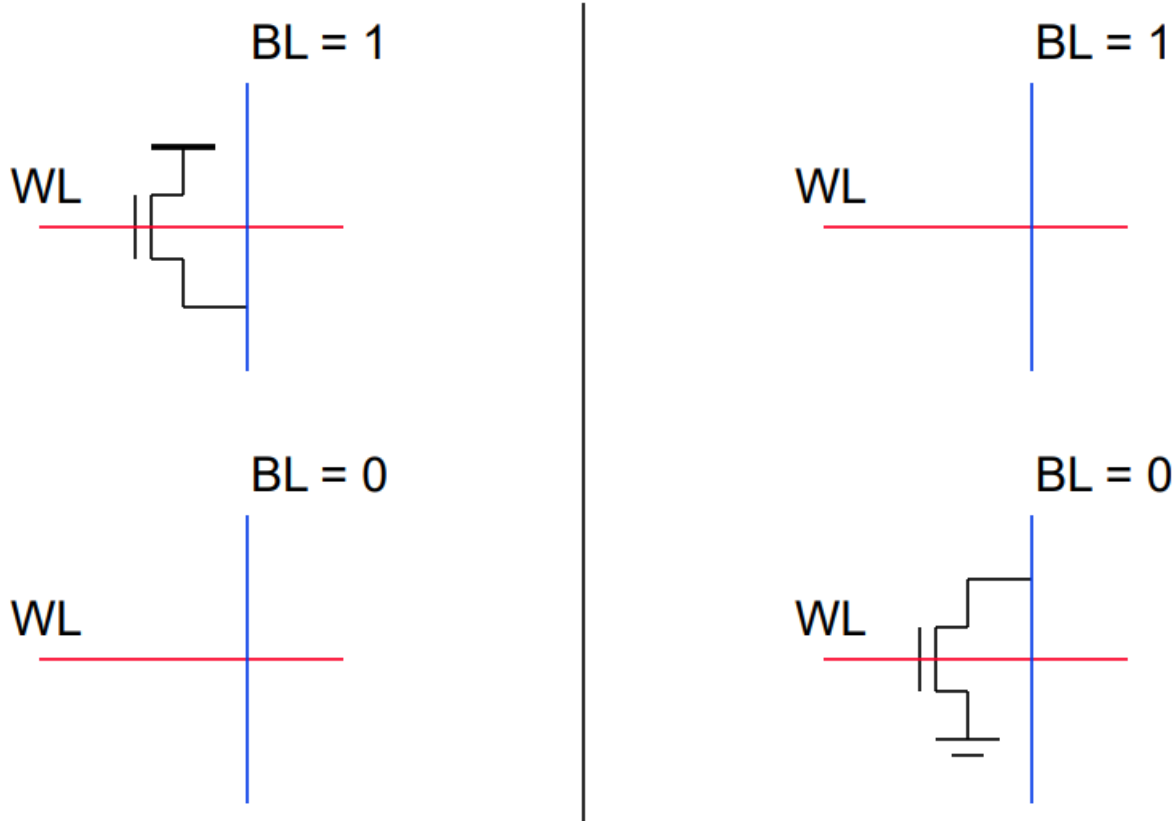
The method of erasing is the main differentiating factor between the various classes of reprogrammable nonvolatile memories.

- ***EPROM:***
  - UV light renders oxide slightly conductive.
  - Erase is slow (seconds to several minutes).
  - Programming is slow (5-10 microsecs per word).
  - Limited number of programming cycles - about 1000.
  - Very dense - single transistor functions as both the programming and access device.

What is ROM?

Read Only Memories (ROMs)

- A memory that can only be read and never altered
  - Programs for fixed applications that once developed and debugged, never need to be changed, only read
  - Fixing the contents at manufacturing time leads to small and fast implementations.



Basic ROM Cell

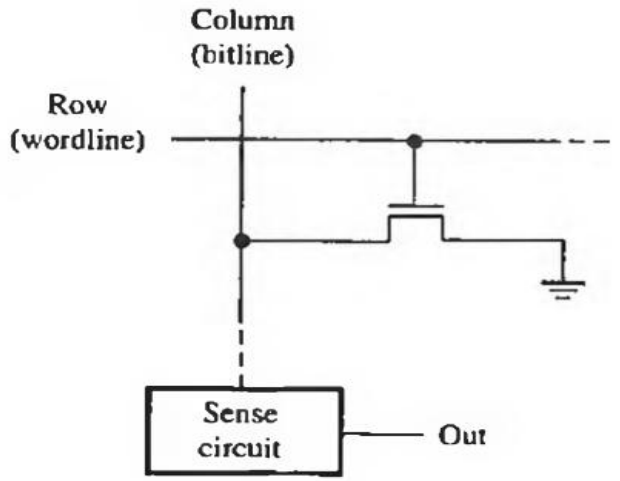
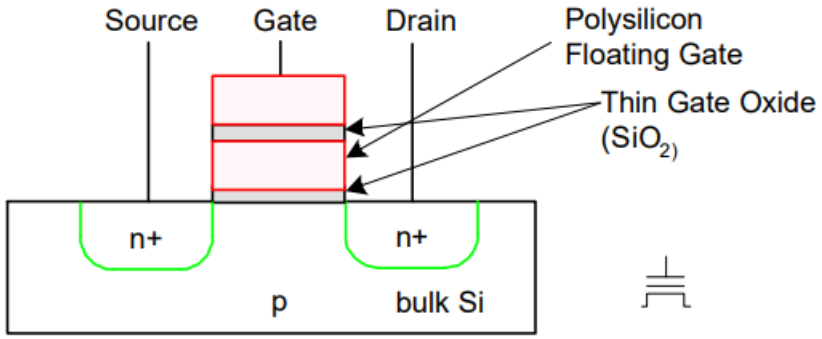


Figure 9.19
Basic ROM cell.

joining row and column. Figure 9.19 shows a single ROM cell with the gate attached to the wordline and the drain attached to the bitline. If the transistor is present at this cross point, it will pull the bitline low when the wordline goes high. However, if it is absent, the bitline will remain at its precharged voltage. This is the key concept in single transistor ROMs and provides the high density levels of the DRAM. This is different from DRAM cells since a transistor is present in every DRAM cell position and its operation relies mainly on charge sharing. In the case of the ROM, the cell may or may not contain a transistor and does require a capacitor for data storage.

PROMs and EPROMs

- Programmable ROMs
  - Build array with transistors at every site
  - Burn out fuses to disable unwanted transistors
- Electrically Programmable ROMs
  - Use floating gate to turn off unwanted transistors
  - EPROM, EEPROM, Flash



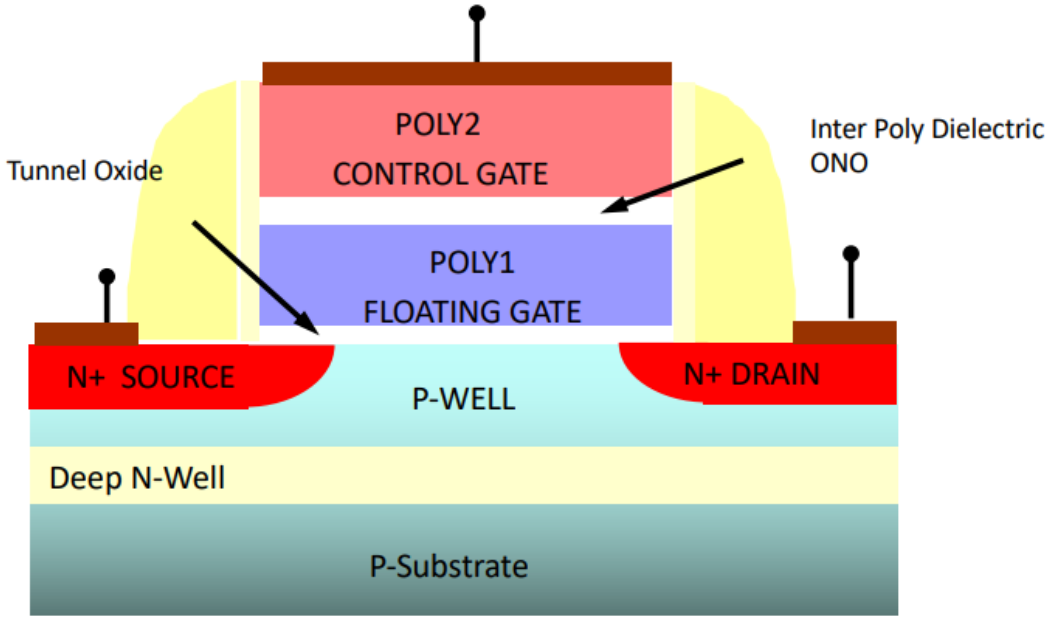
PROM and EPROM

PROMs and EPROMs

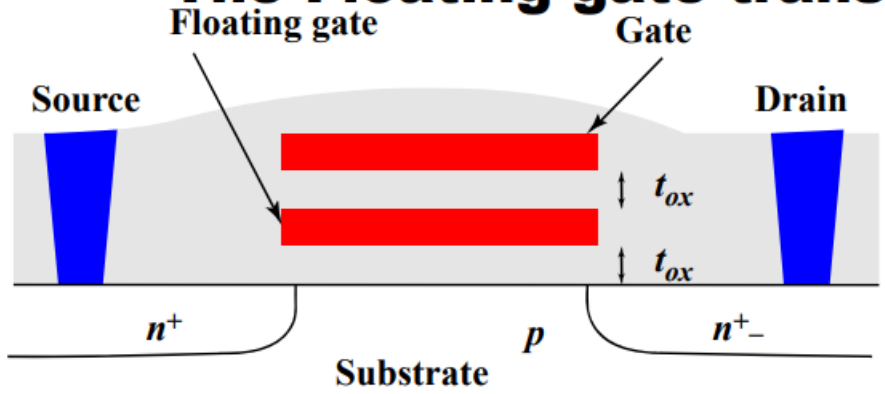
- **Programmable ROMs**
  - Build array with transistors at every site
  - Burn out fuses to disable unwanted transistors
- **Electrically Programmable ROMs**
  - Use floating gate to turn off unwanted transistors
  - EPROM, EEPROM, Flash

Stacked Gate NMOS Transistor

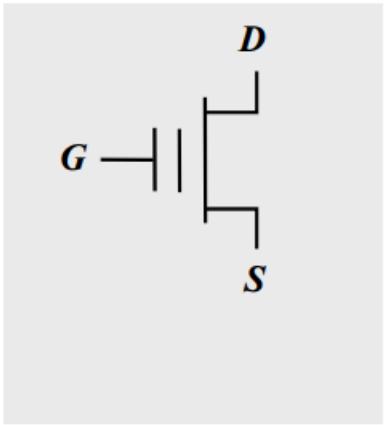
- Poly1 Floating Gate for charge storage
- Poly2 Control Gate for accessing the transistor
- Tunnel-oxide for Gate oxide
- Oxide-Nitride-Oxide (ONO) for the inter Poly Dielectric
- Source/Drain Junctions optimized for Program/Erase/Leakage



Non-Volatile Memories
The Floating-gate transistor (FAMOS)



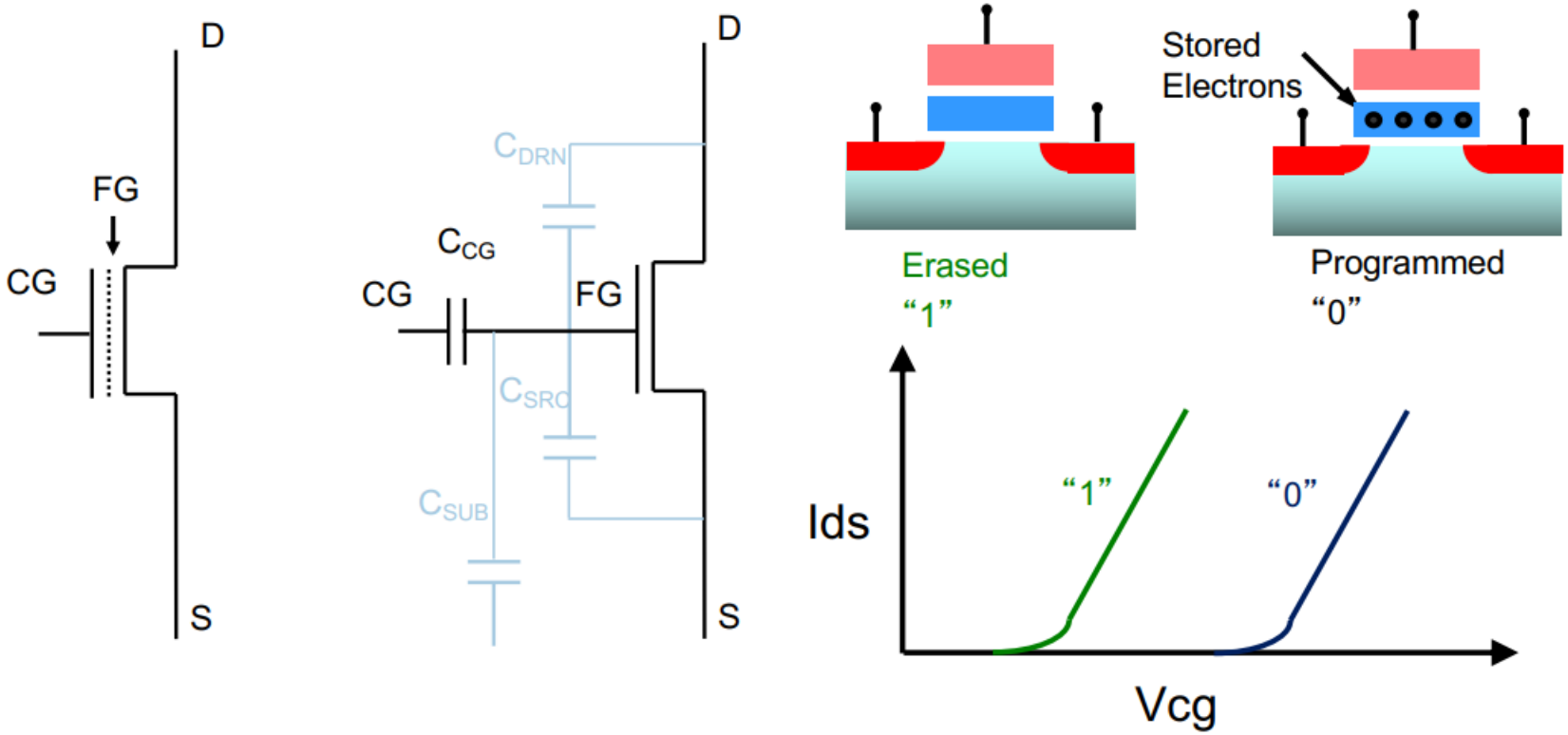
Device cross-section



Schematic symbol

- Storage determined by charge on the floating gate
  - "0" = negative charge (extra electrons)
  - "1" = no charge
- Negative charge on floating gate "screens" normal gate, raising threshold
- Charge can take years to "leak off" once placed there
- **Multi Level** flash: different charge levels represent different values
  - We are "programming" Vt of the transistor

Flash Memory Device – Basic Operation

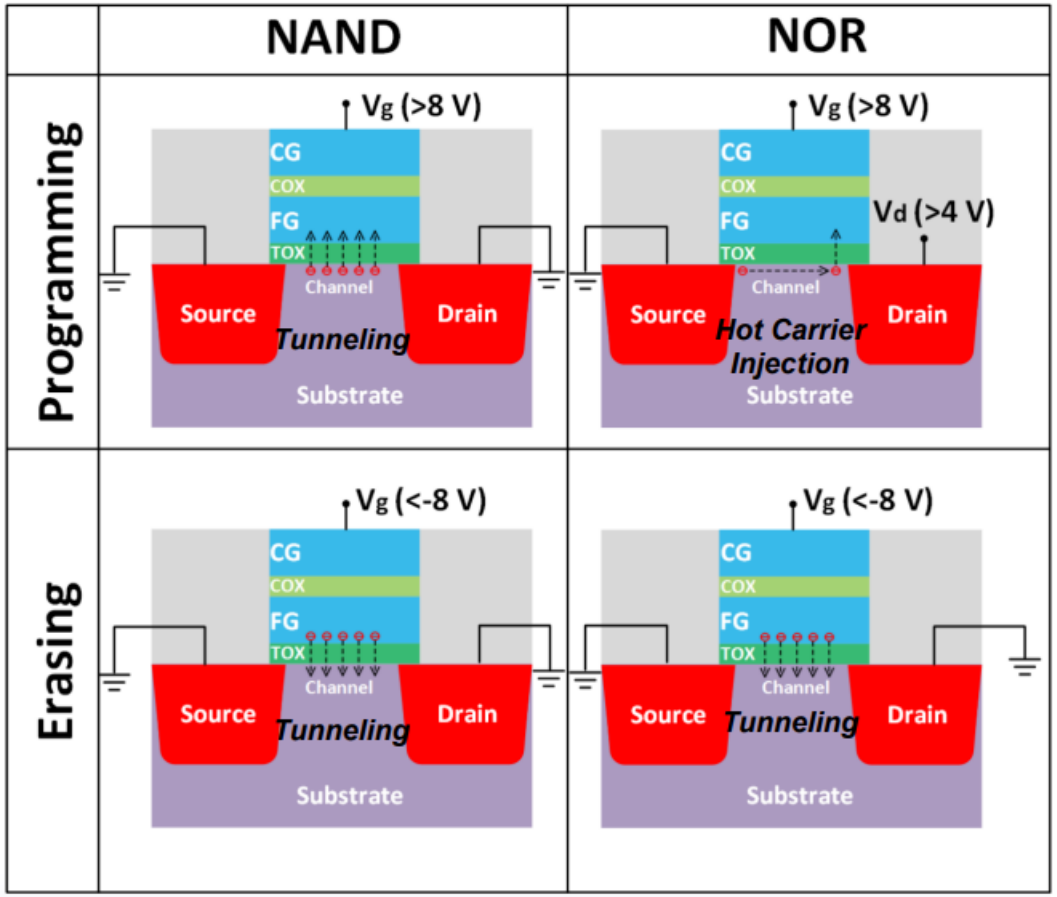


Programming = Electrons Stored on the FG = High Vt

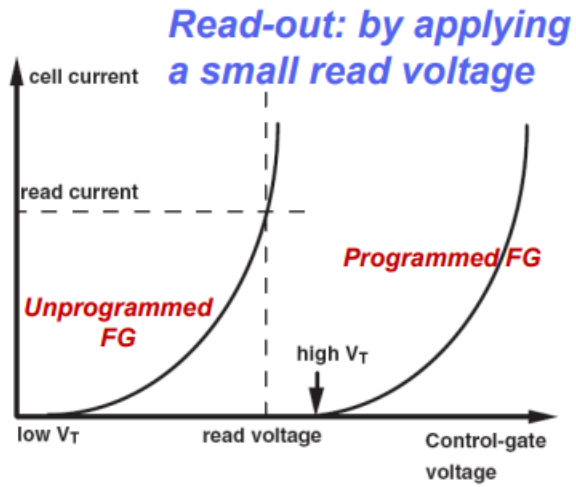
Erasing = Remove electrons from the FG = Low Vt

Threshold Voltage shift = $\Delta Q_{FG}/C_{CG}$

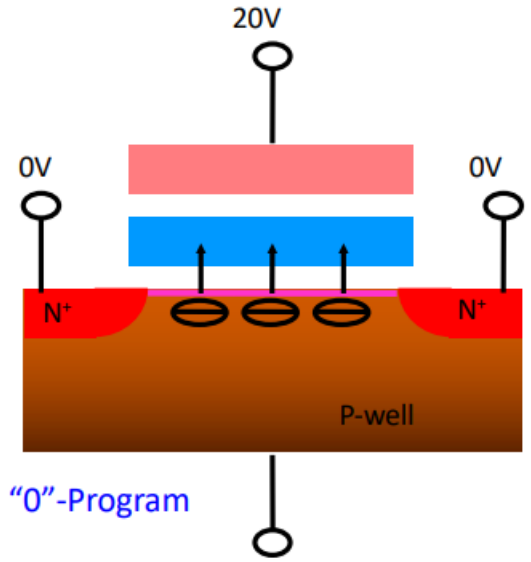
Flash Memory (with Floating Gate (FG) Transistor)

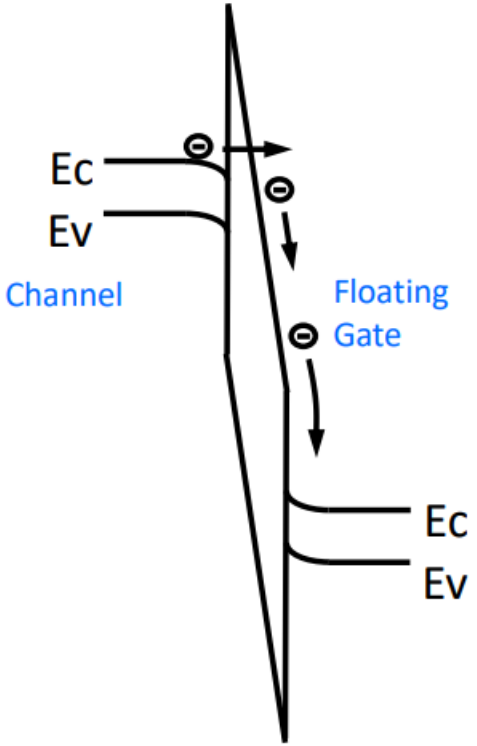


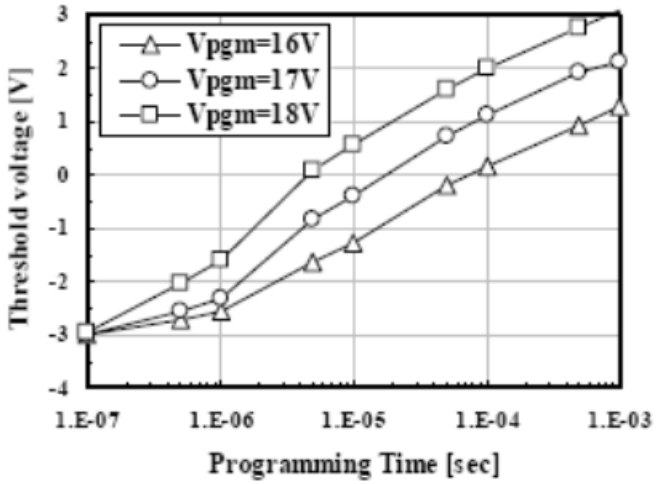
N-type device
CG: control gate
COX: control oxide
FG: floating gate
TOX: tunnel oxide



Nand Flash Programming – FN Tunneling







Y. S. Yim, et al. IEDM 2003

- Tunnel Programming from channel by biasing the Top Gate positive with respect to the ground
- Program Time ~300us
- Program current ~ Displacement plus Tunneling current. Low current allows large parallelism.

EPROM Cell Structure

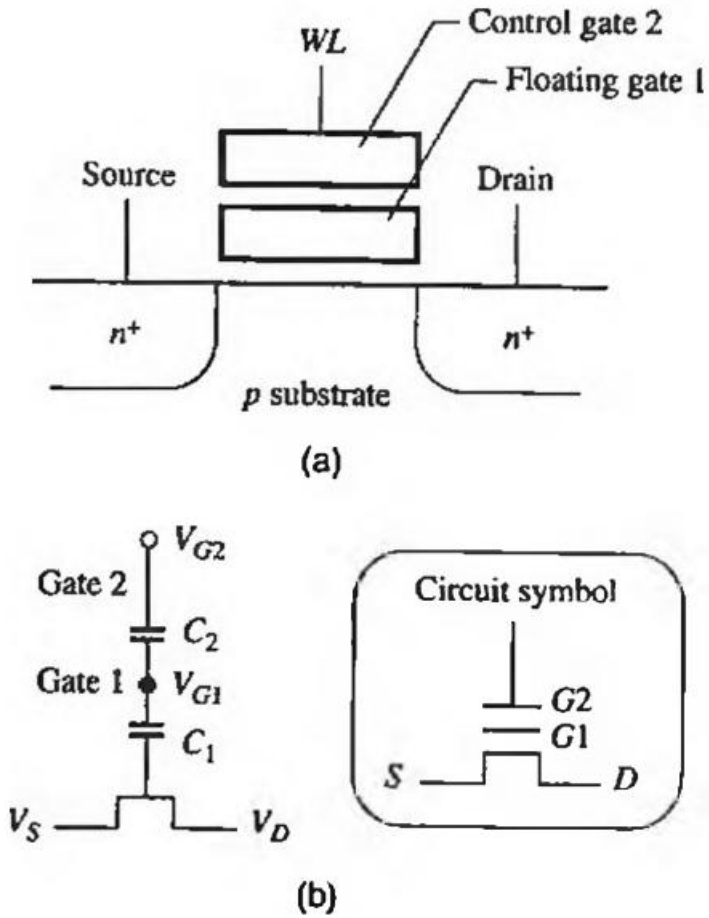
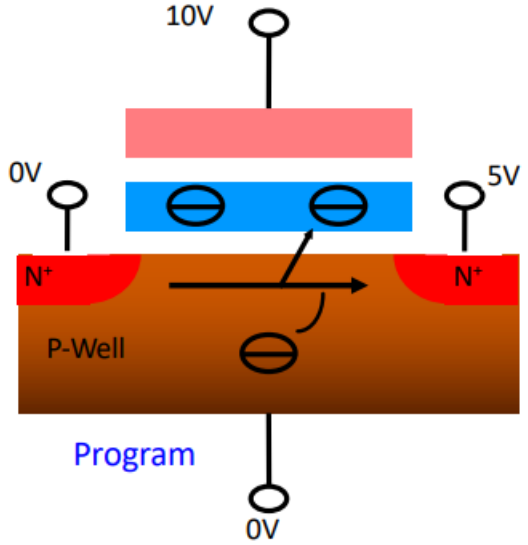
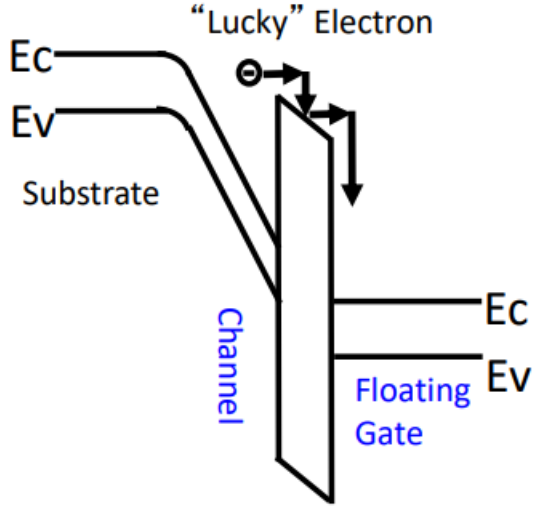
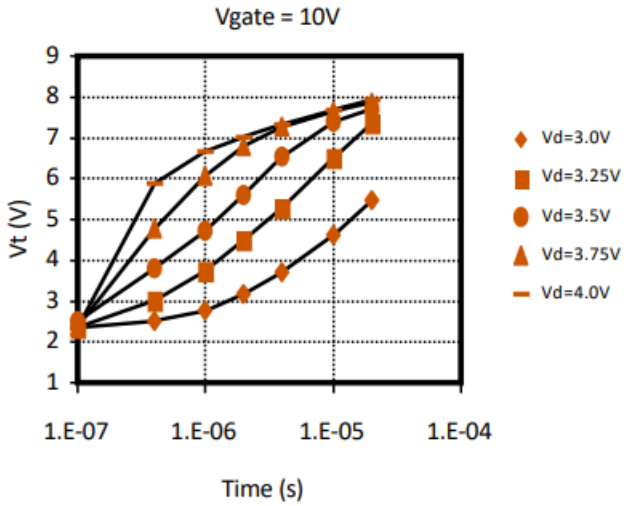


Figure 9.22
EPROM cell structure.

NOR Flash Programming – Channel Hot Electron







Channel Hot Electron Programming - Gate voltage inverts channel; drain voltage accelerates electrons towards drain; gate voltage pulls them to the floating gate

In Lucky Electron Model, electron crosses channel without collision, gaining > 3.2eV, hits Si atom, bounces over barrier

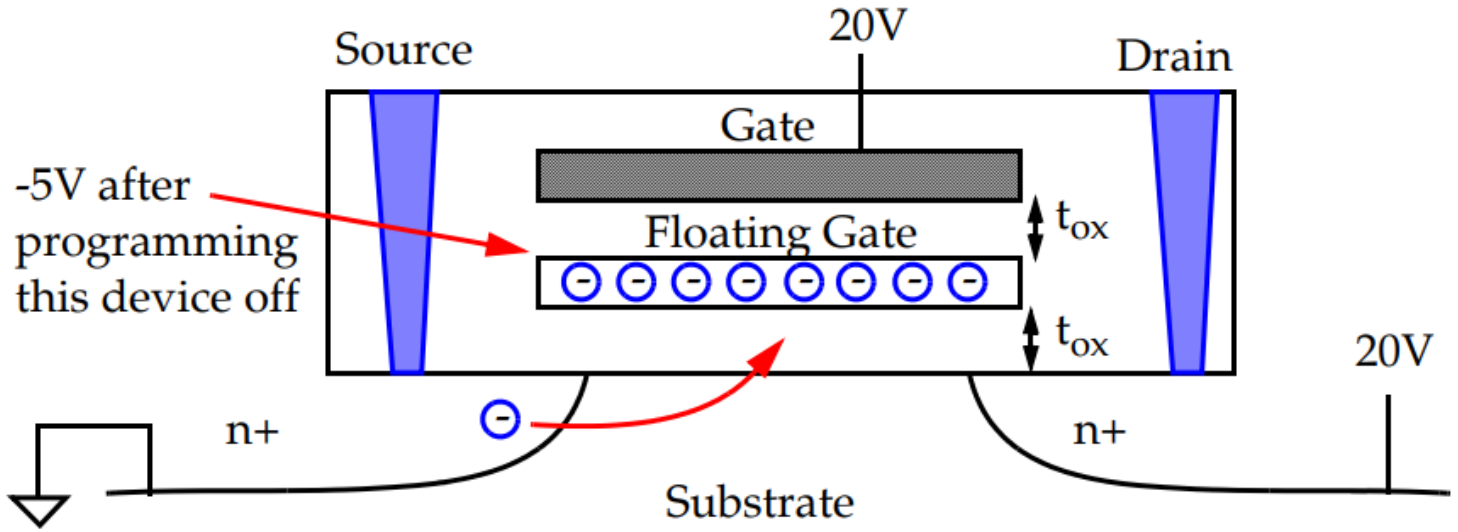
Program Time ~ 0.5-1ms. Program current ~50mA/cell

Non-Volatile Read-Write Memory

Non-volatile Read-Write Memories

Virtually identical in structure to ROMs.

Selective enabling/disabling of transistors is accomplished through modifications to **threshold voltage**. This is accomplished through a floating gate.



Applying a high voltage (15 to 20 V) between source and gate-drain create high electric field and causes avalanche injection to occur.

Hot electrons traverse first oxide and get trapped on floating gate, leaving it negatively charged.

This increases the threshold voltage to ~7V. Applying 5V to the gate does not permit the device to turn on.

Flash EEPROM

Non-volatile Read-Write Memories

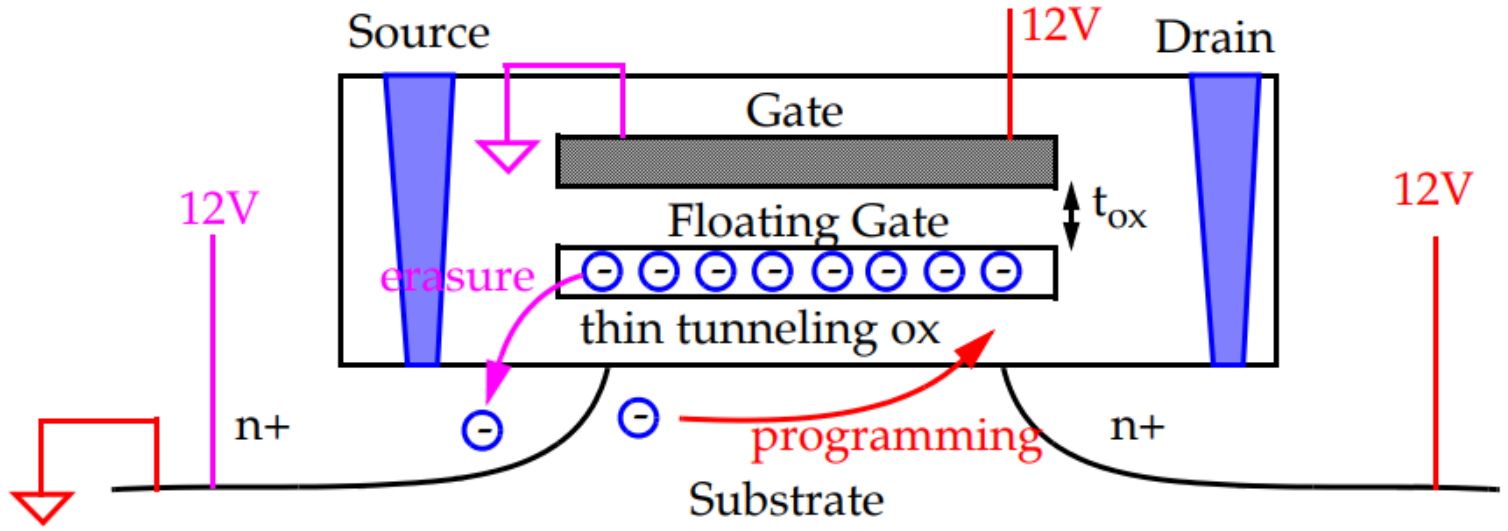
- ***Flash EEPROM:***

Combines density adv. of EPROM with versatility of EEPROM.

Uses avalanche hot-electron-injection approach to program.

Erasure performed using Fowler-Nordheim tunneling.

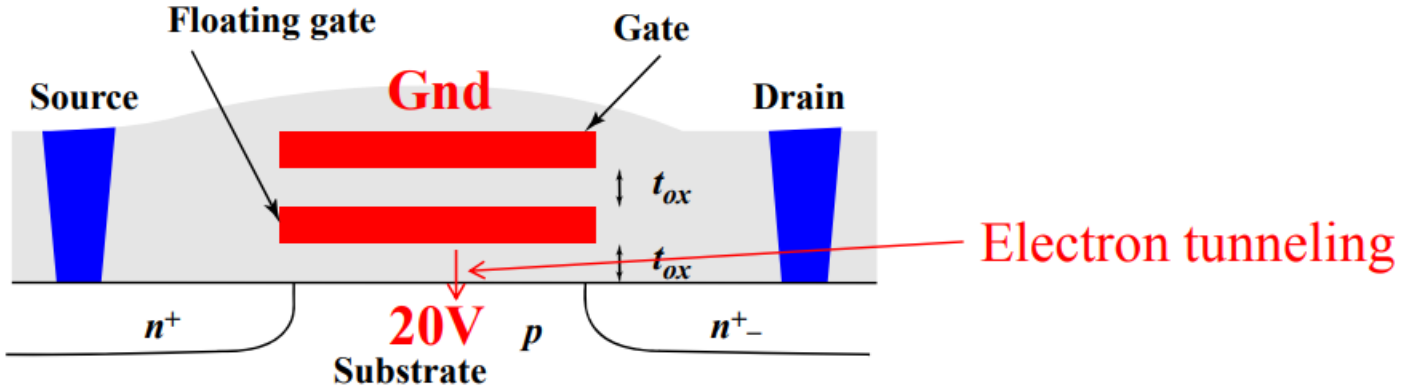
Monitoring control hardware checks the value of the threshold during erasure - making sure the unprogrammed transistor remains an enhancement device.



Programming performed by applying 12V to gate and drain.

Erasure performed with gate grounded and source at 12V.

Block Erasure



- If we can only write "0"s, how do we store "1"s?
- Answer: we "erase" all cells to 1 before writing and write only 0s
- **Erasing process**:
  - Set substrate very high (e.g. 20V)
  - Set all control gates to ground
  - Over time (ms), electrons on floating gates tunnel to substrate
- Cannot control substrate voltage of single transistors, so erase ***all cells in a block*** at the same time

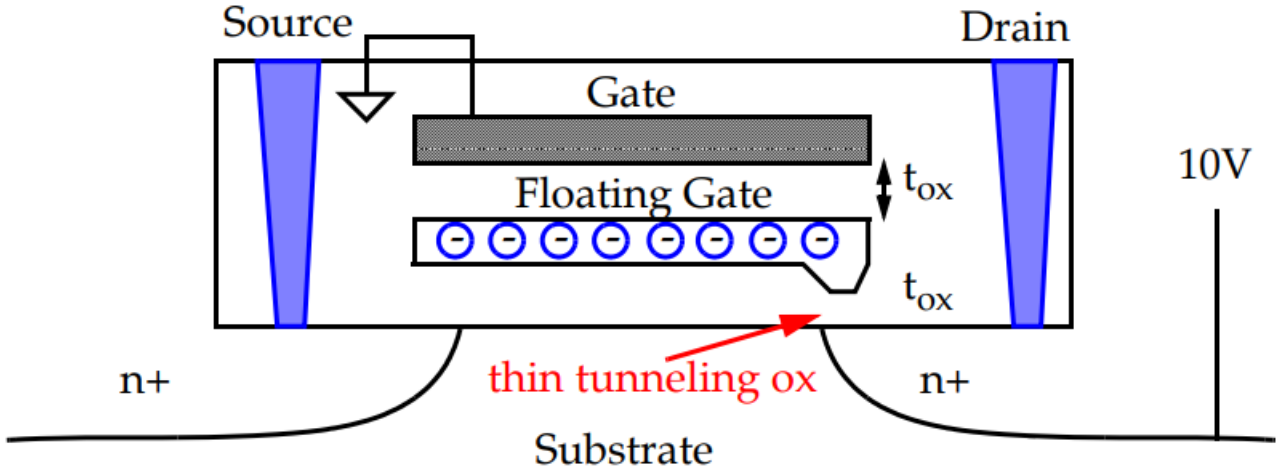
EEPROM

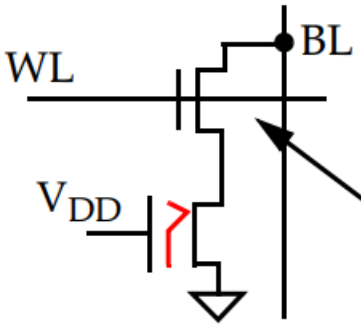
Non-volatile Read-Write Memories

- ***EEPROM or E^2PROM:***

Very thin oxide allows electrons to flow to and from the gate via Fowler-Nordheim tunneling with V_{GD} applied.

Erasure is achieved by reversing the voltage applied during writing.



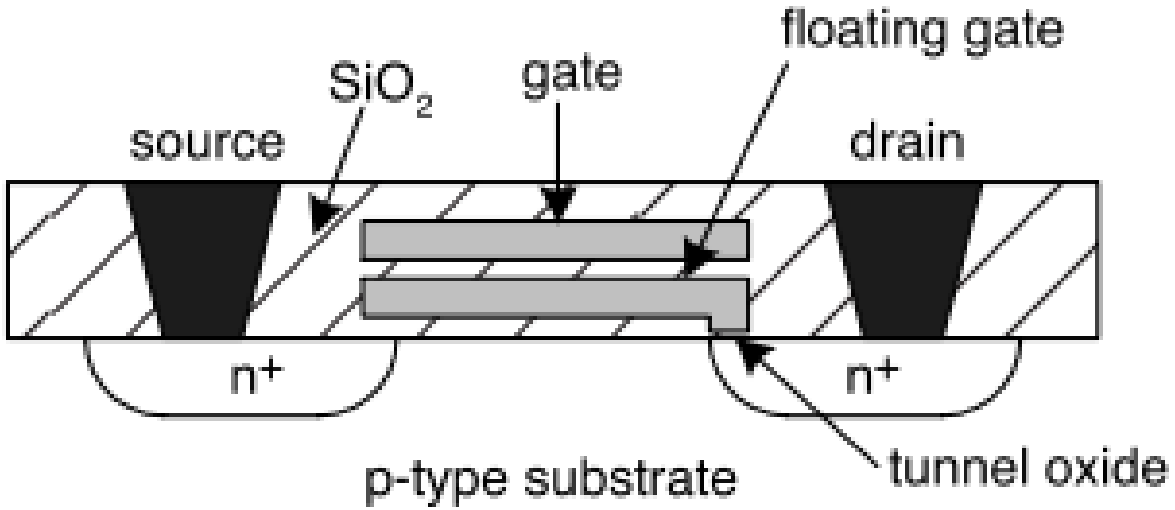


Threshold control becomes a problem:

Removing too much charge results in a depletion device that cannot be turned off.

Remedy: Add an access transistor.

FLOTOX Transistor



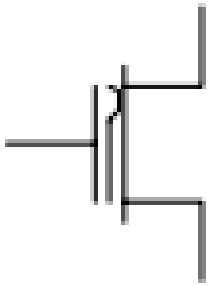


FIGURE 16.17
FLOTOX transistor and circuit symbol.

EPROM CELL

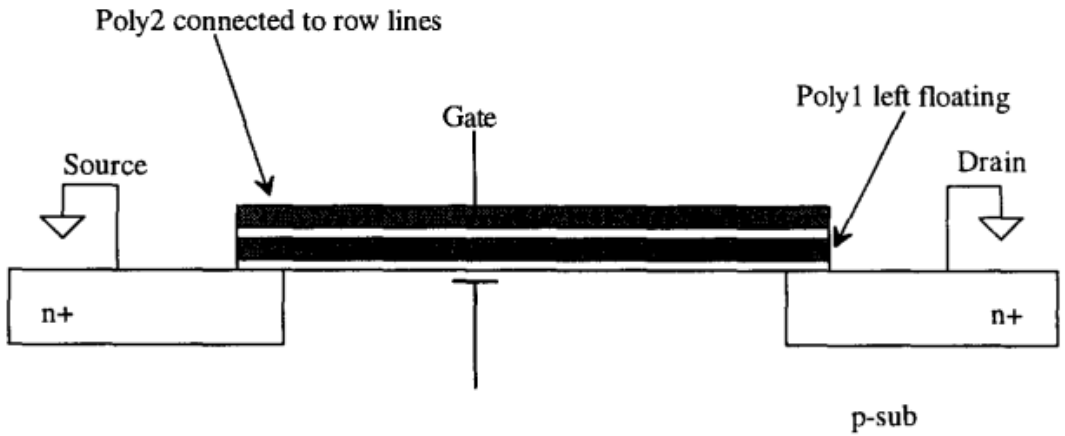


Figure 17.26 EPROM memory cell.

Erasable Programmable Read-Only Memory (EPROM)

EPROMs make programming the ROM significantly easier. Consider the cross-sectional view of an EPROM cell shown in Fig. 17.26. This modified n-channel MOSFET is used at the interesection of the column and row lines in the ROM memory array shown in Fig. 17.24. A second layer of polysilicon is added directly above the original polysilicon layer. The original poly layer is floating (i.e., not connected to anything). The second layer (poly2) is now used for connection to the row lines. Note that this is simply a poly2-poly1 capacitor where the bottom plate of the capacitor (poly1) is used in MOSFET formation.

To understand how to program the MOSFET, let's begin by assuming that both gates, poly1 (the bottom) and poly2, are at 0 V. A capacitance exists between poly2 and poly1 as well as between poly1 and the substrate. If the potential of poly2 is increased above 0 V, the voltage between these two capacitors ideally divides evenly since they should be approximately the same value. The result is an increase in the potential on poly1. If the potential on poly2 is raised to approximately $2 \cdot V_{THN}$, then the potential of the poly1 is raised to approximately V_{THN}. Increasing the row line voltage (the voltage

EEPROM

Electrically Erasable Programmable Read-Only Memory (EEPROM)

The inability to reprogram EPROM quickly has led to the use of EEPROM in many applications requiring nonvolatile memory. A voltage generator (see Ch. 18) is used on chip to generate the large voltage needed to program the EEPROM memory cell. The gate-oxide used in EEPROM is thinner than that used in EPROM. The result, when a large voltage, 10 V, is applied to poly2, is a conduction mechanism called Fowler-Nordheim tunneling, [11] between the substrate and poly1. This mechanism, unlike avalance breakdown, can conduct current in both directions between poly1 and substrate. A logic high in an EEPROM is programmed by raising the voltage on poly2 to 10 V, while a logic low is programmed by lowering the voltage to –10 V.

Flash memory is based on both EPROM and EEPROM technologies. Flash memories are programmed in a fashion similar to EPROM where hot electrons are used to accumulate charge on poly2 (see Fig. 17.26). Structurally, the Flash memory cell is the same as the EPROM cell except for the oxide thickness [9]. The oxide thickness is on the order of 100 Å, whereas the oxide thickness used in EPROM is 200-400 Å thick. Unlike EPROM, Flash memories can be erased in a fashion similar to EEPROMs. In other words, the Flash memory is programmed using hot electrons and erased using Fowler-Nordheim tunneling.

FLASH MEMORY

Flash memory is based on both EPROM and EEPROM technologies. Flash memories are programmed in a fashion similar to EPROM where hot electrons are used to accumulate charge on poly2 (see Fig. 17.26). Structurally, the Flash memory cell is the same as the EPROM cell except for the oxide thickness [9]. The oxide thickness is on the order of 100 Å, whereas the oxide thickness used in EPROM is 200-400 Å thick. Unlike EPROM, Flash memories can be erased in a fashion similar to EEPROMs. In other words, the Flash memory is programmed using hot electrons and erased using Fowler-Nordheim tunneling.

NAND And NOR ROM ARRAY

CELL CONNECTIONS IN ROM

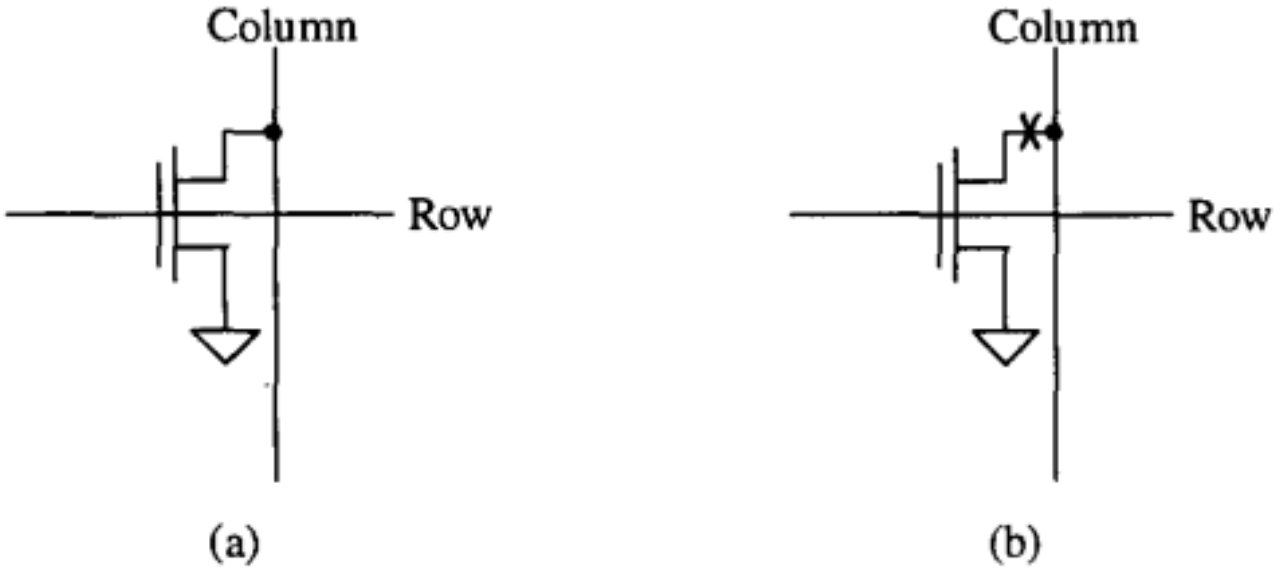
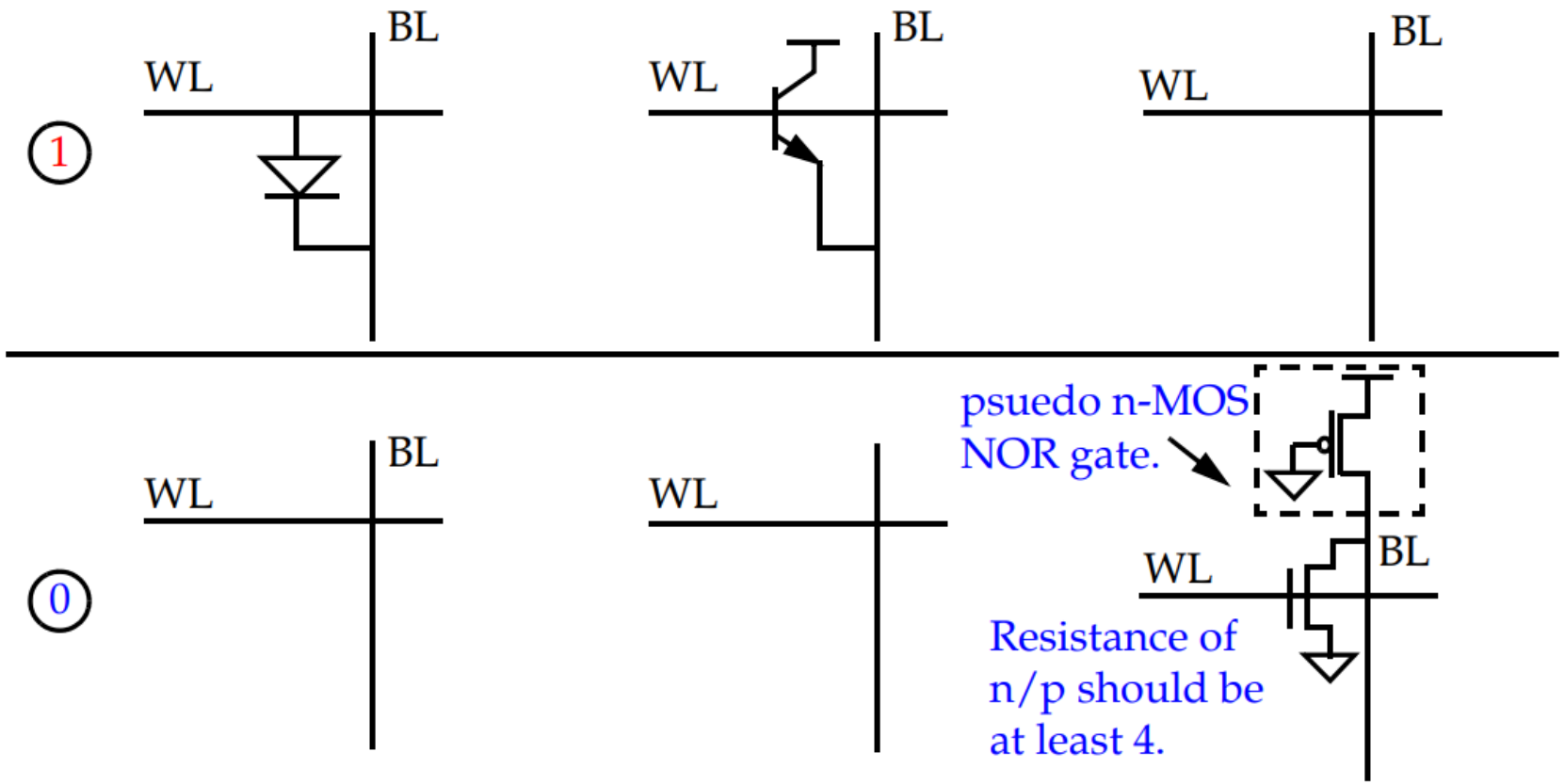


Figure 17.25 (a) n-channel MOSFET at the intersection of every column and row line and (b) eliminating the connection between the drain and column line to program the ROM.

ROM Circuit Design Possibility

ROM

ROM cells are permanently fixed: Several possibilities:

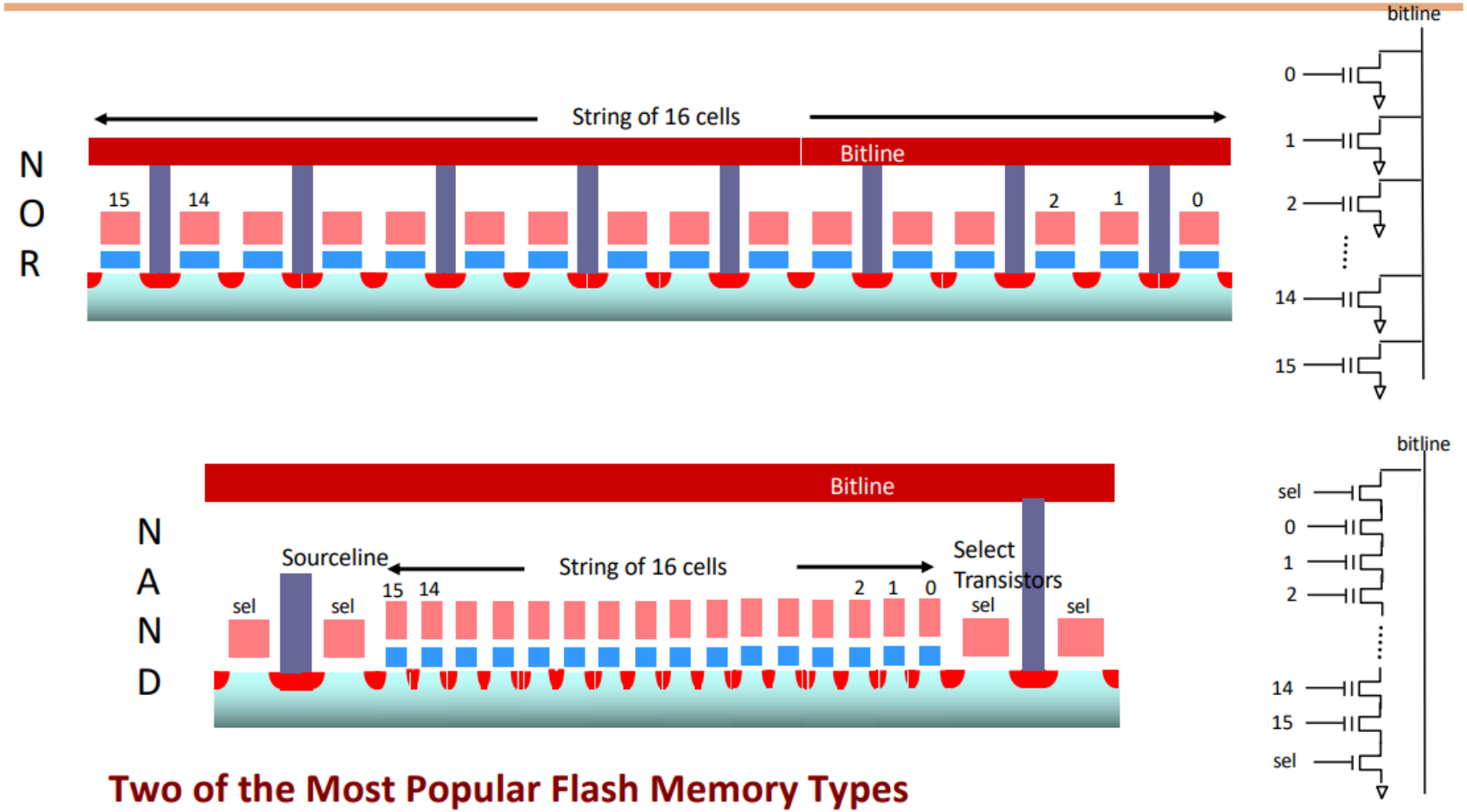


Diode supplies current to raise BL (bitline) for all cells on the row.

BJT supplies current to raise BL for each cell on the row. Requires V_{DD} to be routed.

p-MOS used to hold BL high. n-MOS provides pull-down path.

NAND vs. NOR Cross-sections



Two of the Most Popular Flash Memory Types

Both have Dual Gate NMOS with charge storage in Poly1 floating gate

Lack of contacts in NAND cell makes it inherently smaller in size

NAND vs. NOR

Merits of NAND

★ *Multiple cells at once*

① High speed programming (★)
② High speed erasing
③ High capacity
④ Low cost per bit

Merits of NOR

① High speed random access
② Bit programming

(Individual bit cell for programming/erasing process)

Demerits of NAND

① Slow random access
② Bit programming cannot be performed
③ Slow read operation

(Serial access => sequential data readout)

Demerits of NOR

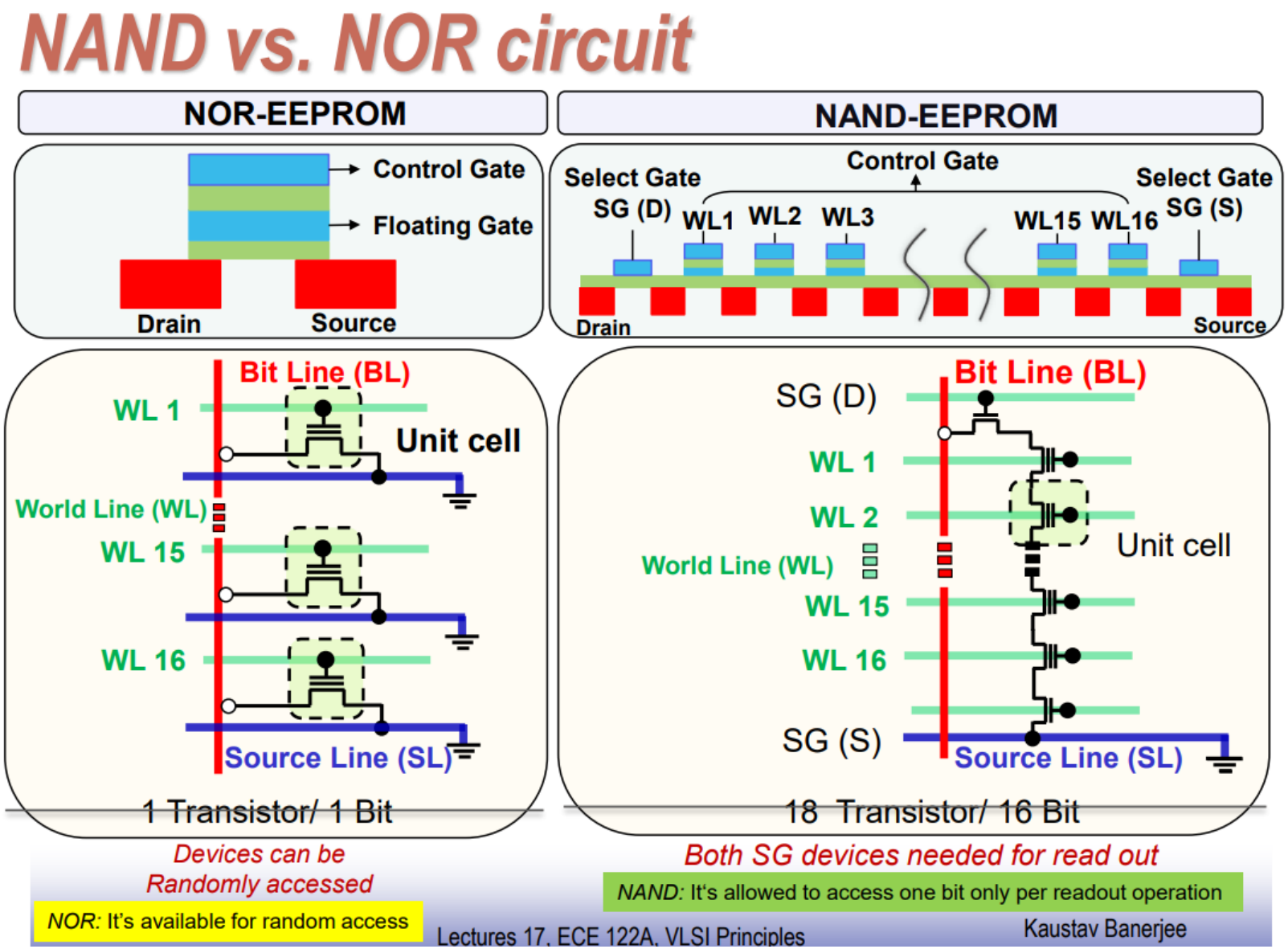
① Slow programming
② Slow speed erasing
③ Low capacity
④ High cost per bit

(NOR must be erased in large "chunks". Those chunks can often be subdivided and erased in smaller subunits resulting in a performance penalty)

- Applications -
• Suitable for Data Memory (Handy terminal, Voice recorder, DSC, Fax modem, etc)

- Applications -
• Suitable for replacement of EPROM
• Suitable for control memory (BIOS, Cellular, HDD, etc)

NAND VS NOR CIRCUIT



Diode ROM

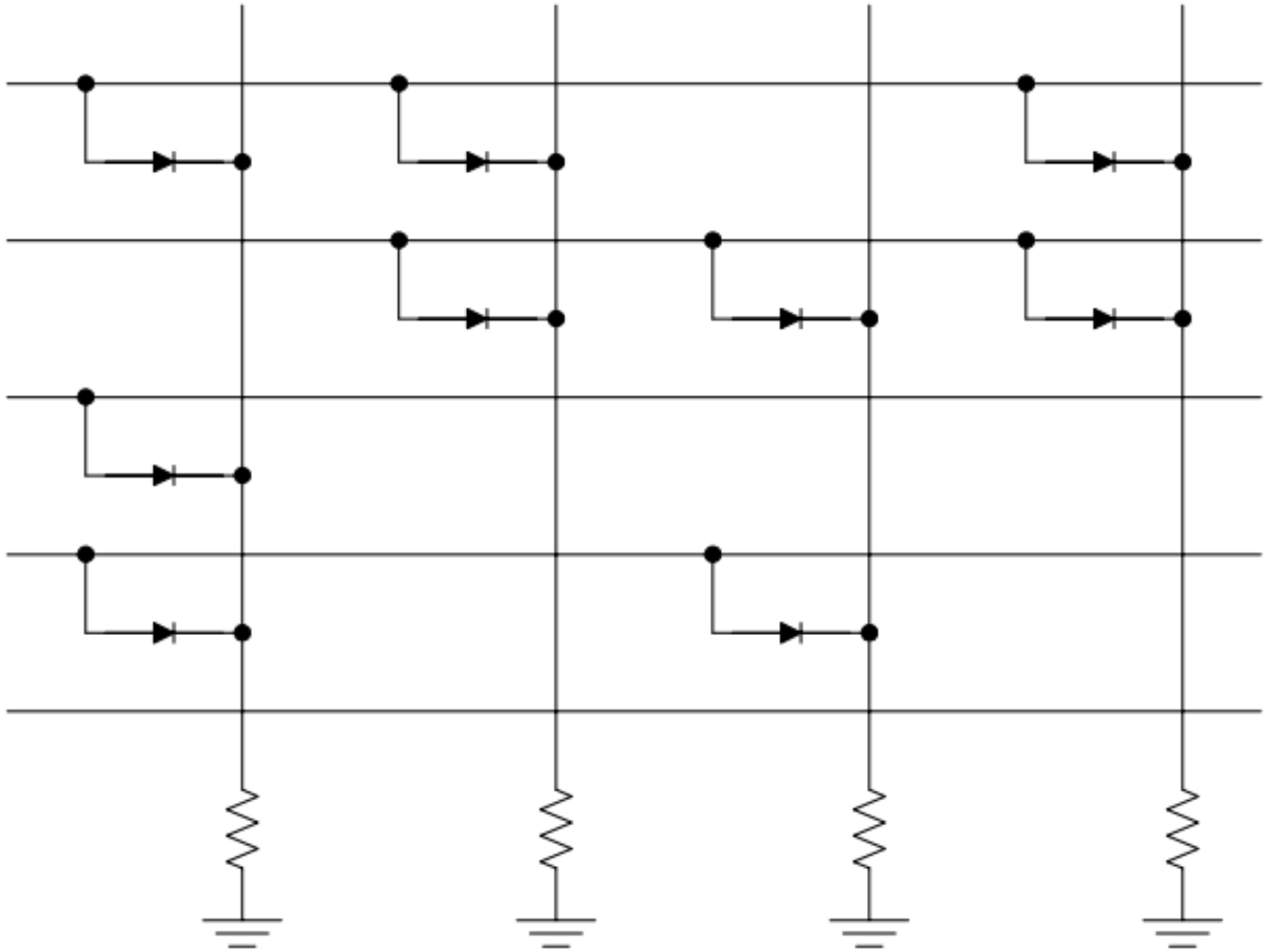
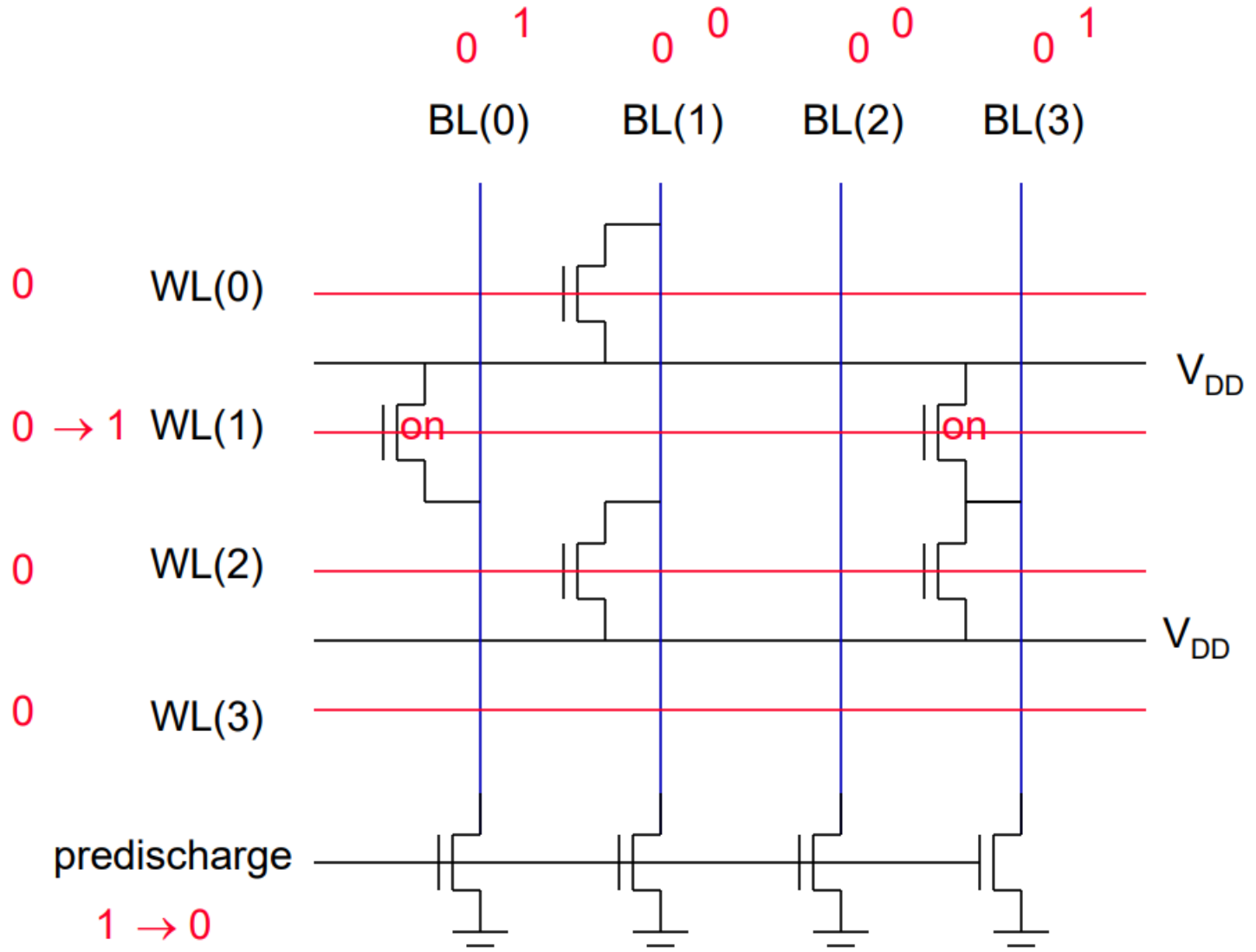


FIGURE 16.10
Diode ROM.

ROM CELL ARRAY

MOS OR ROM Cell Array



NMOS NOR ROM

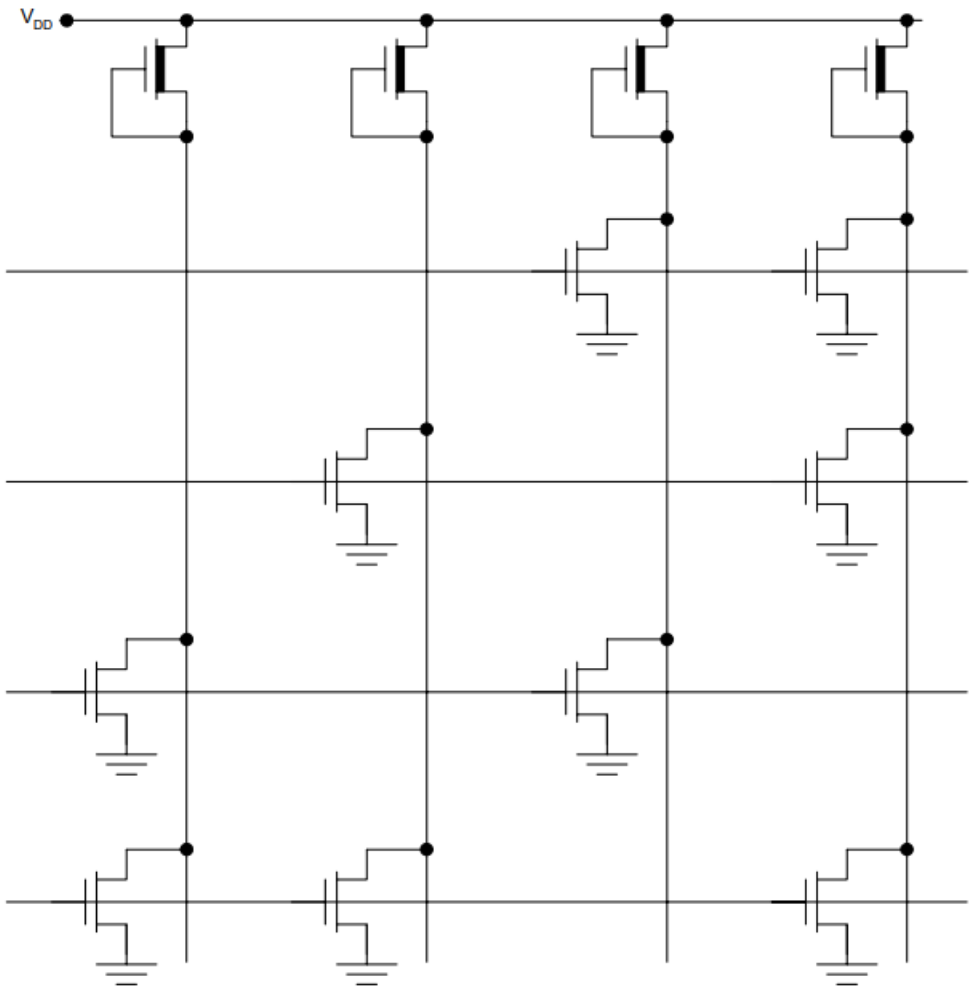


FIGURE 16.12
NMOS NOR read-only memory.

CMOS NOR ROM

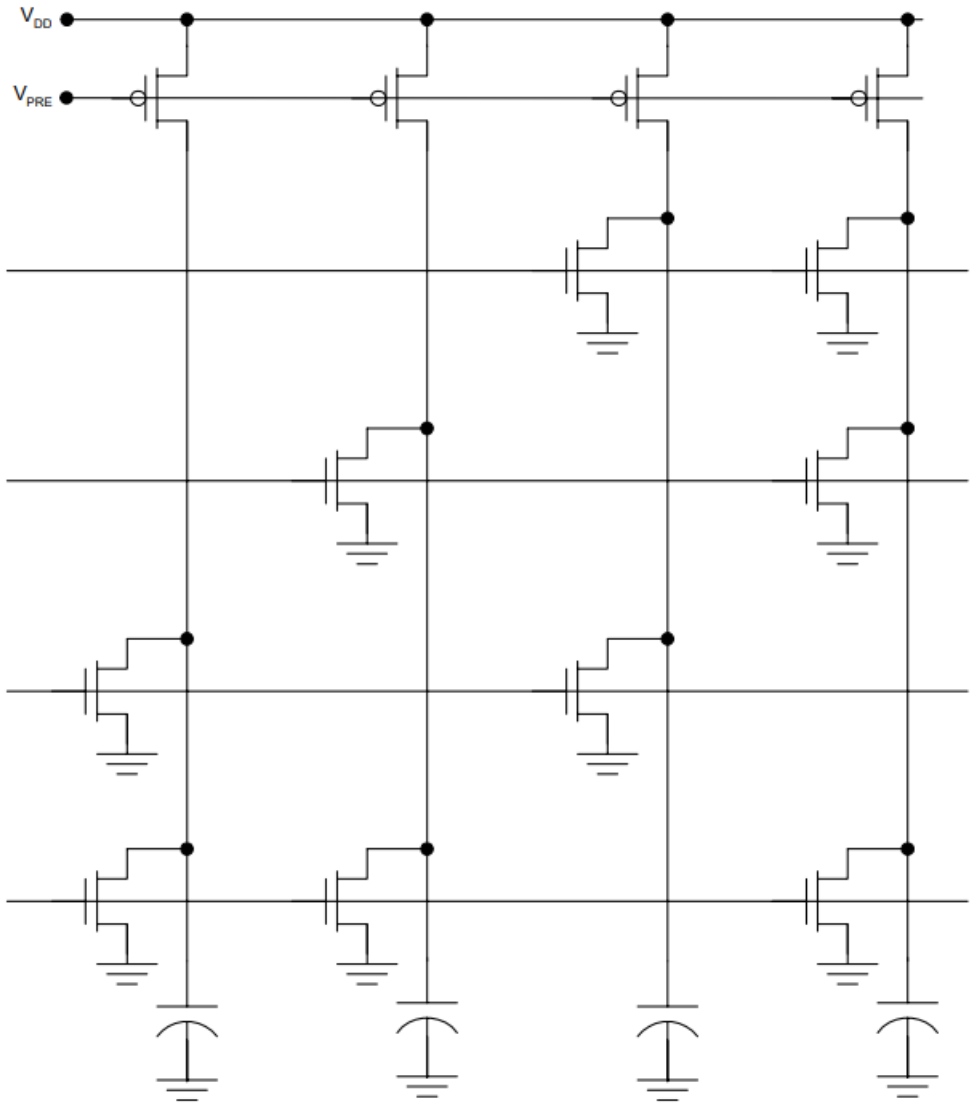
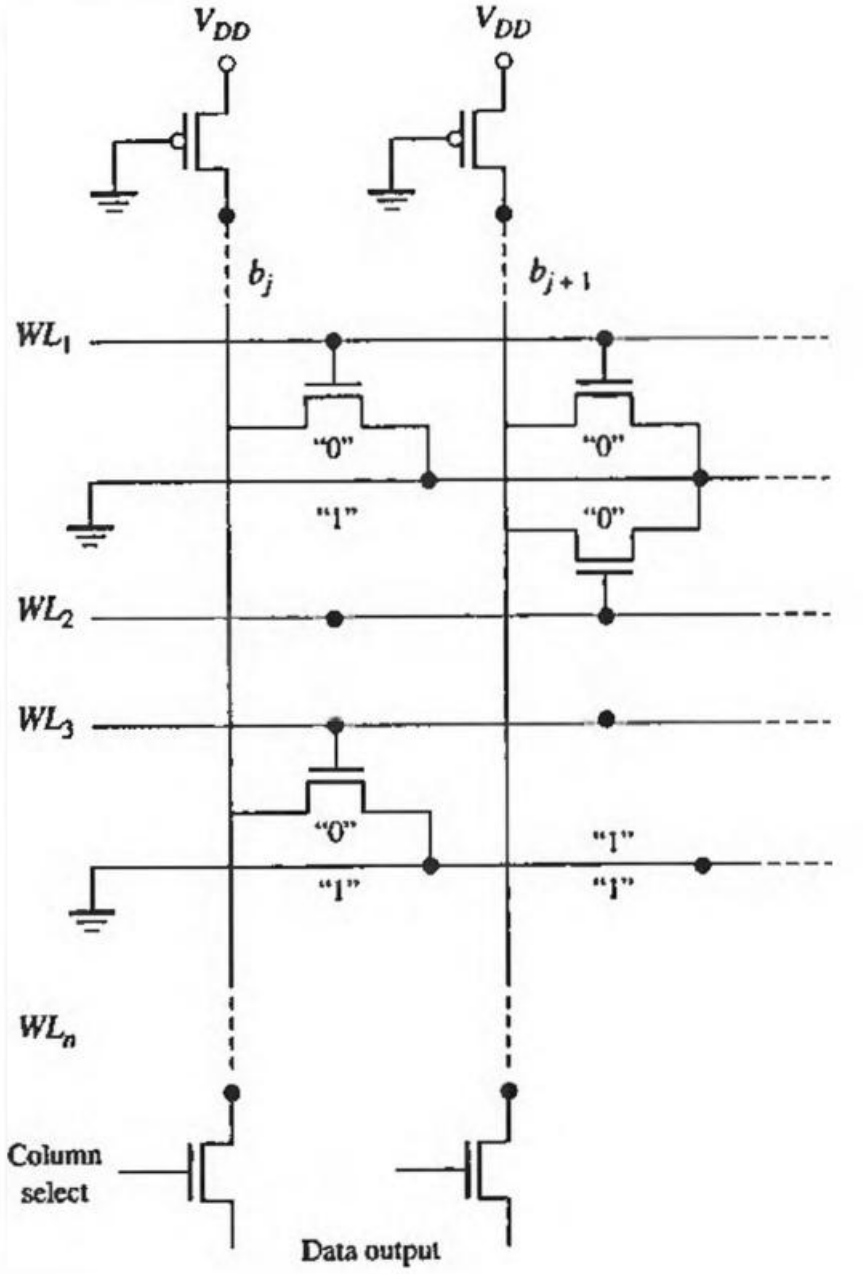


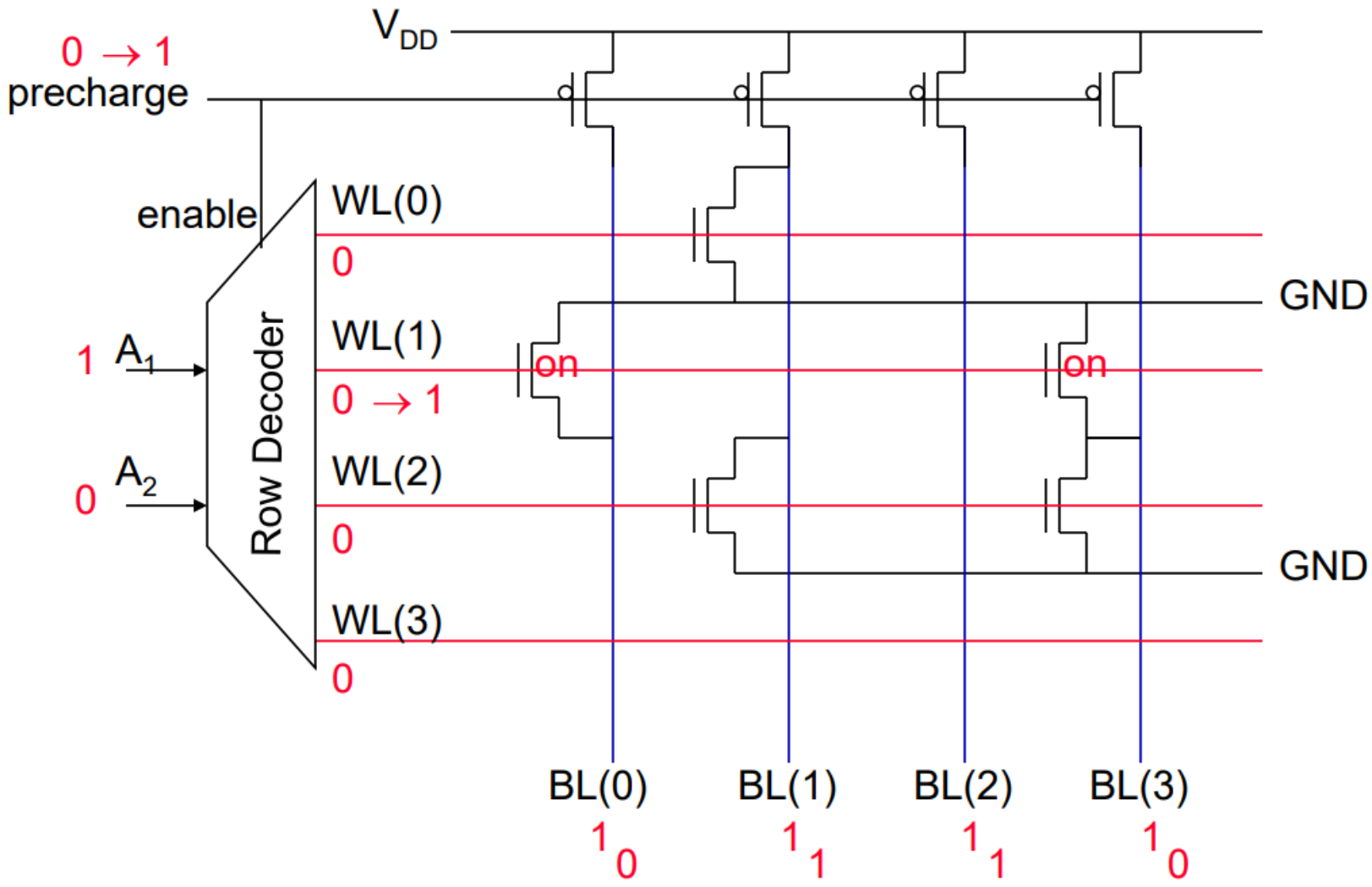
FIGURE 16.13
CMOS NOR read-only memory.

NOR ARRAY



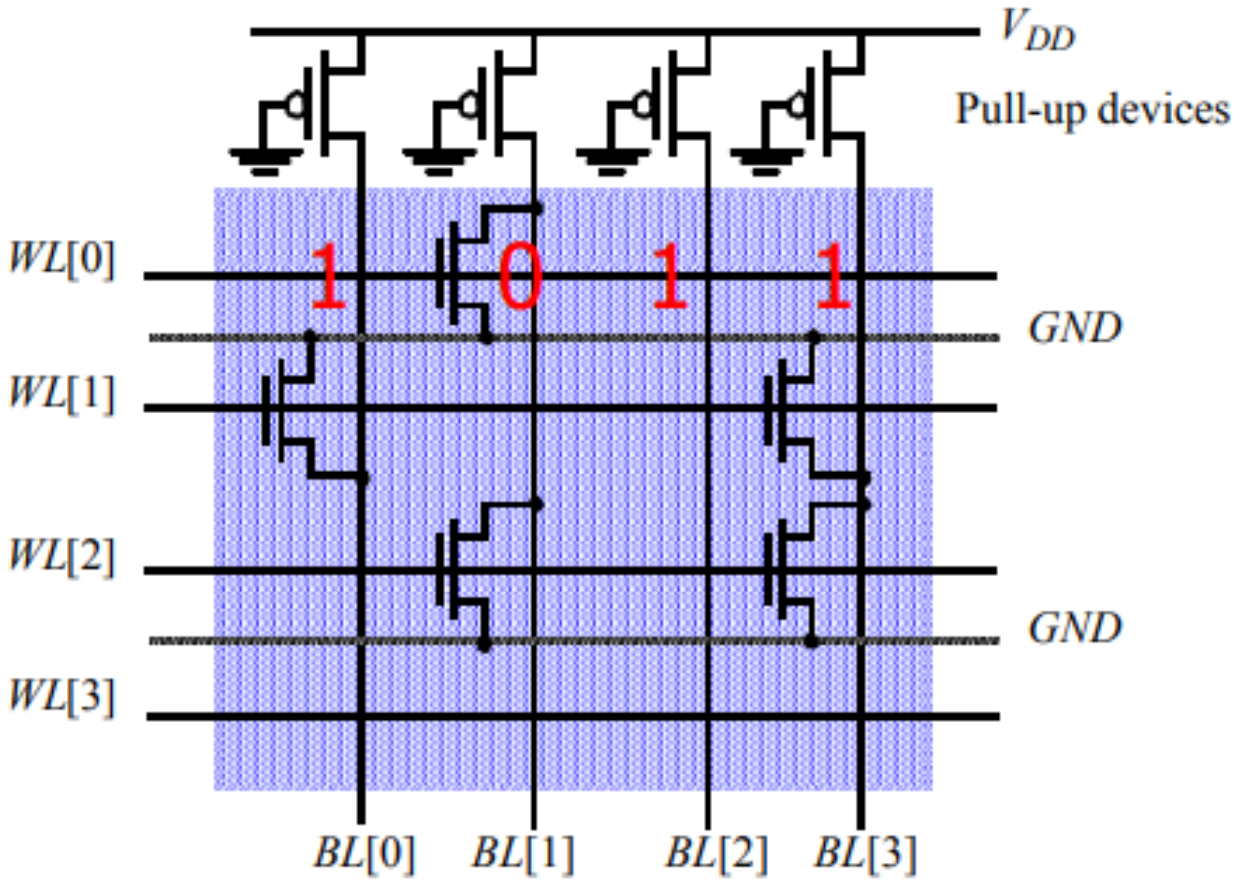
(a) NOR Array

Precharged MOS NOR ROM



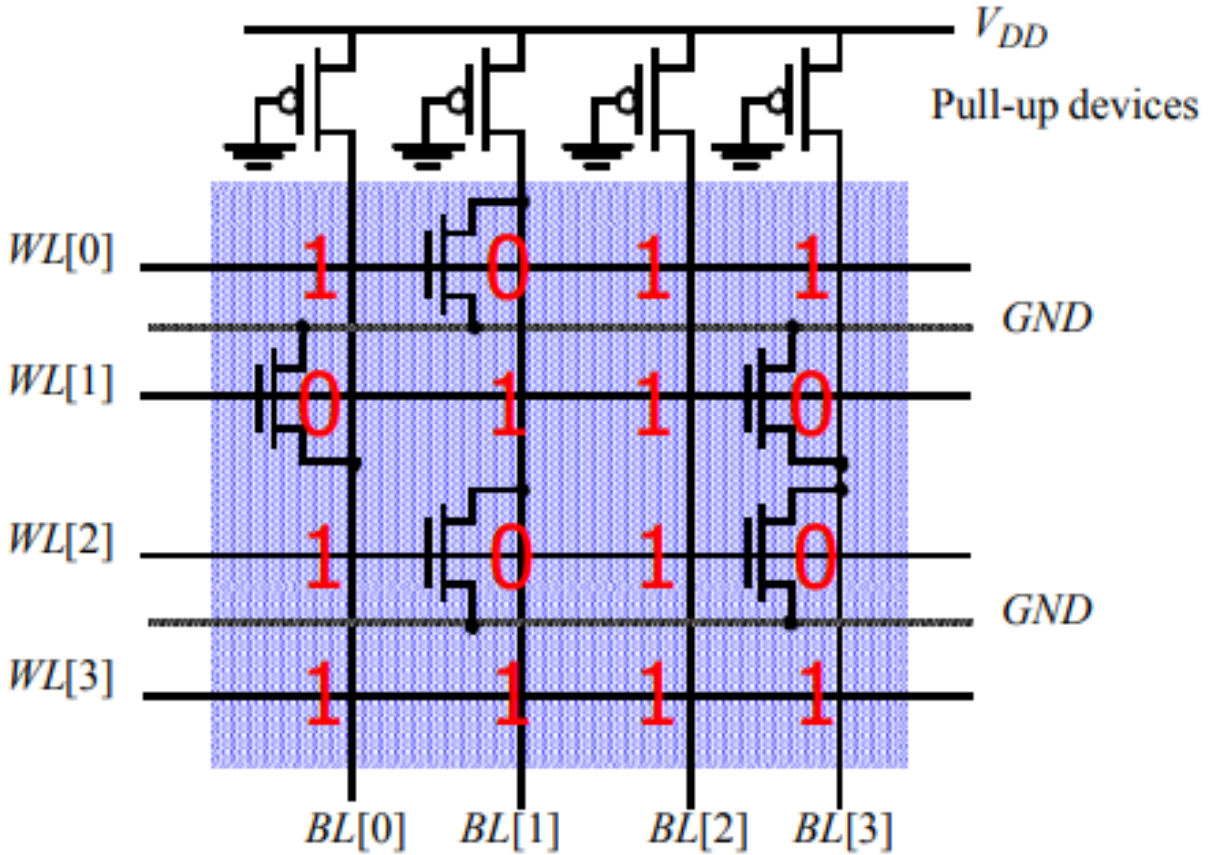
MOS NOR ROM

MOS NOR ROM

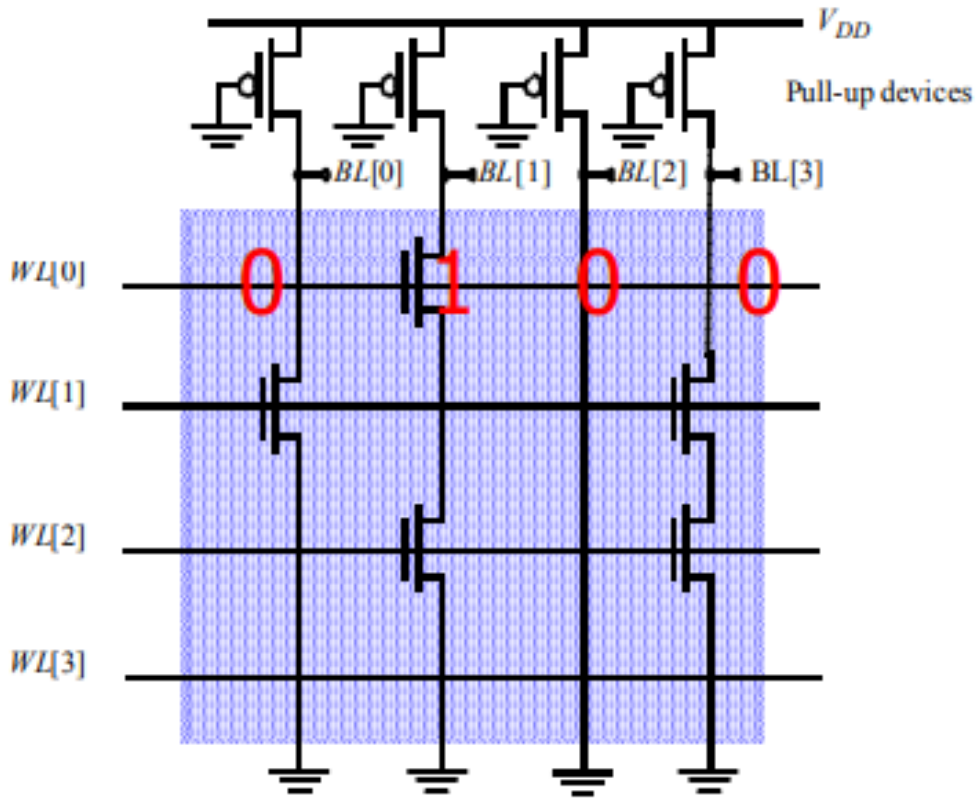


MOS NOR ROM

MOS NOR ROM

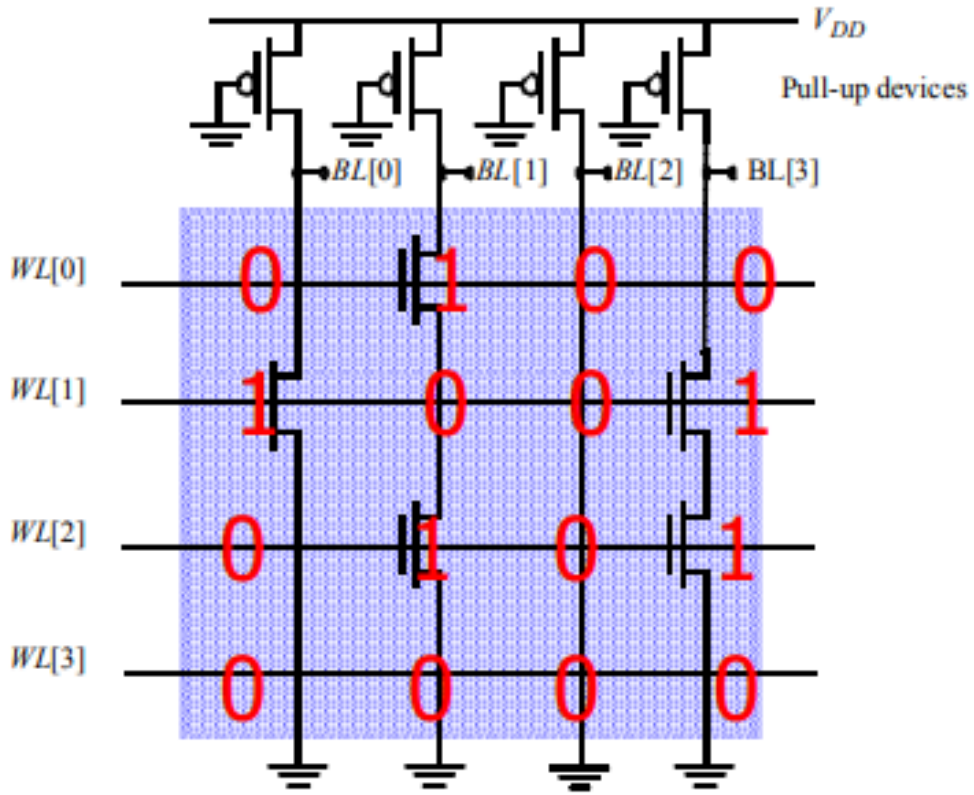


MOS NAND ROM



All word lines high by default with exception of selected row

MOS NAND ROM



All word lines high by default with exception of selected row

Pseudo NMOS Nand ROM Array

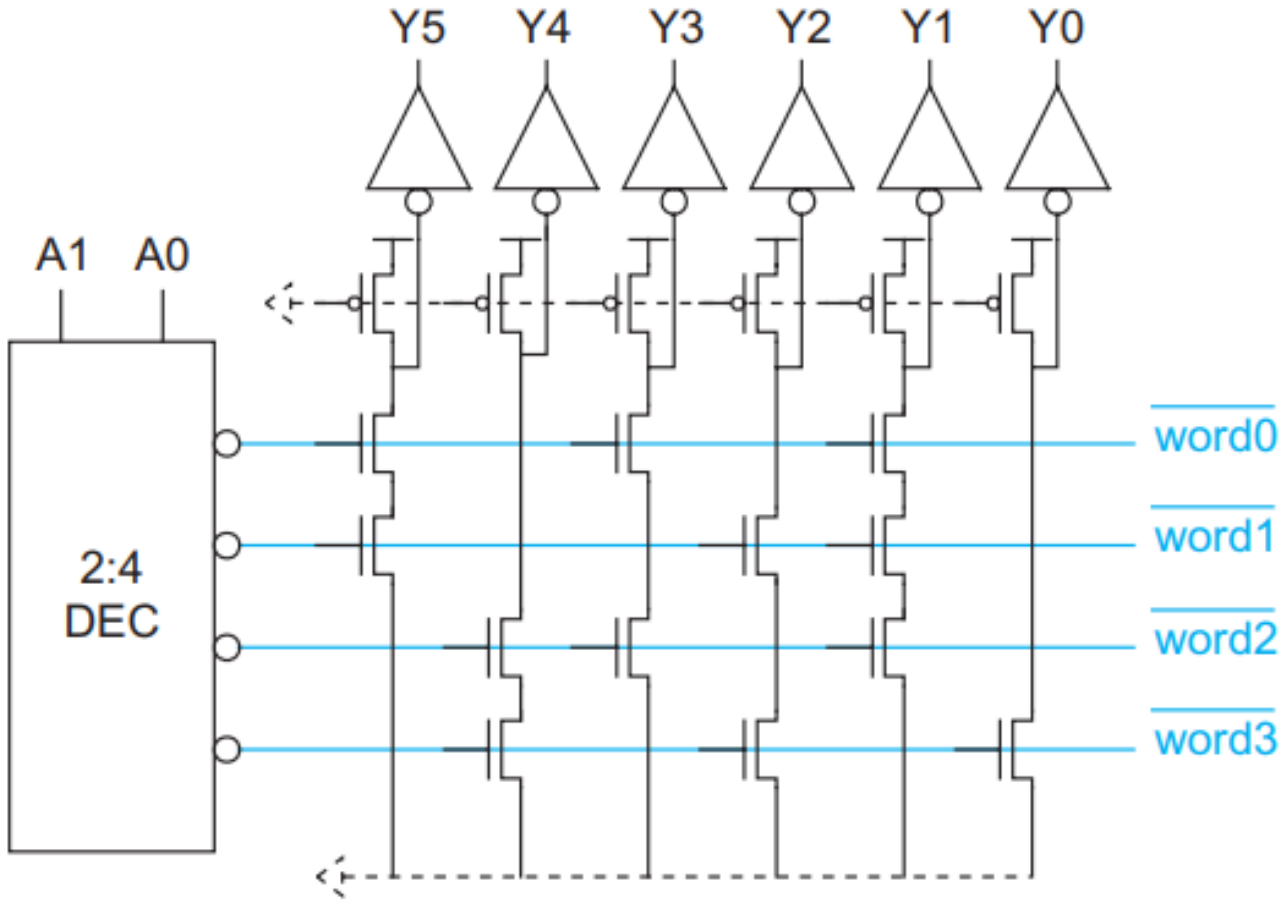
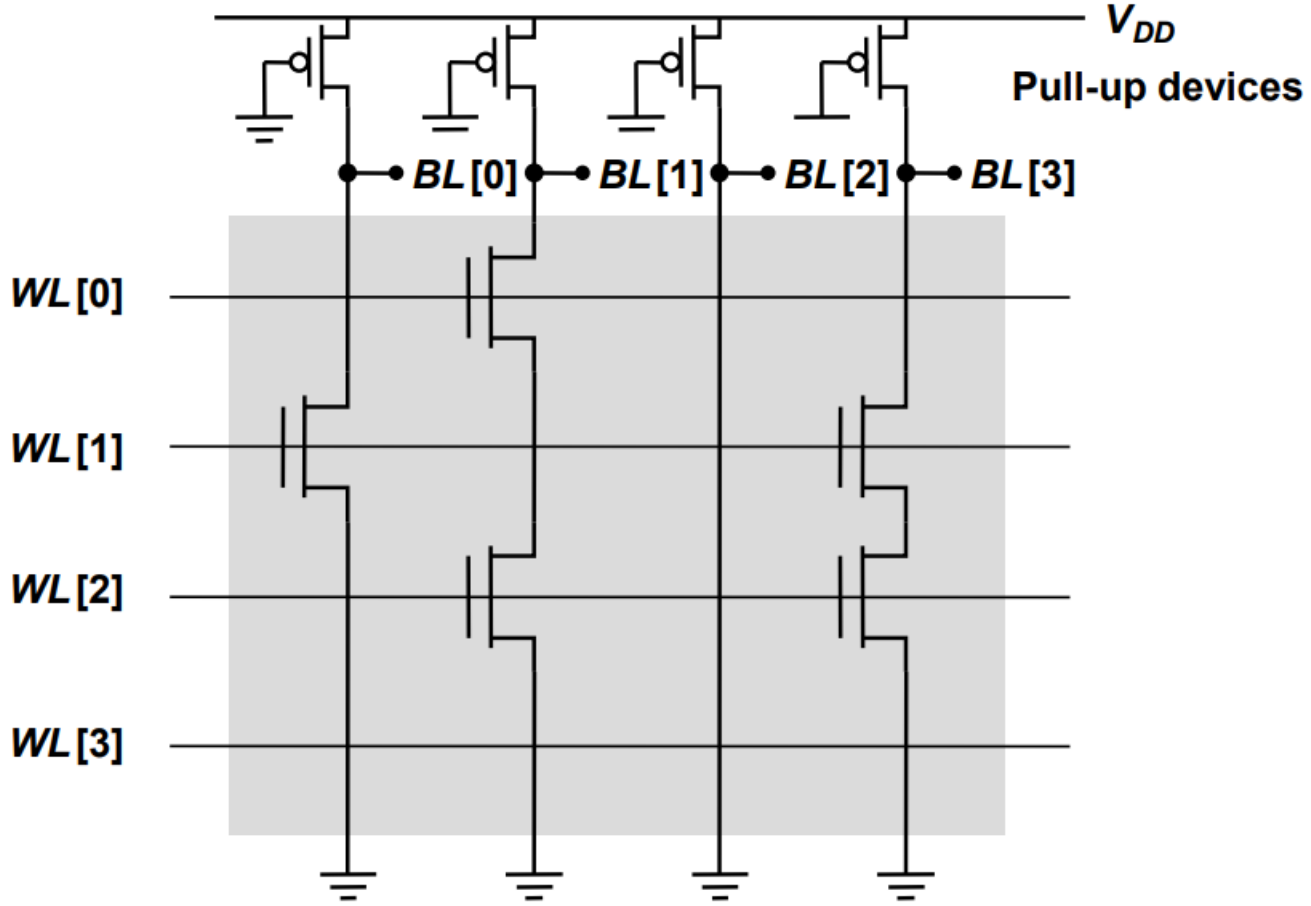


FIGURE 12.58 Pseudo-nMOS NAND ROM

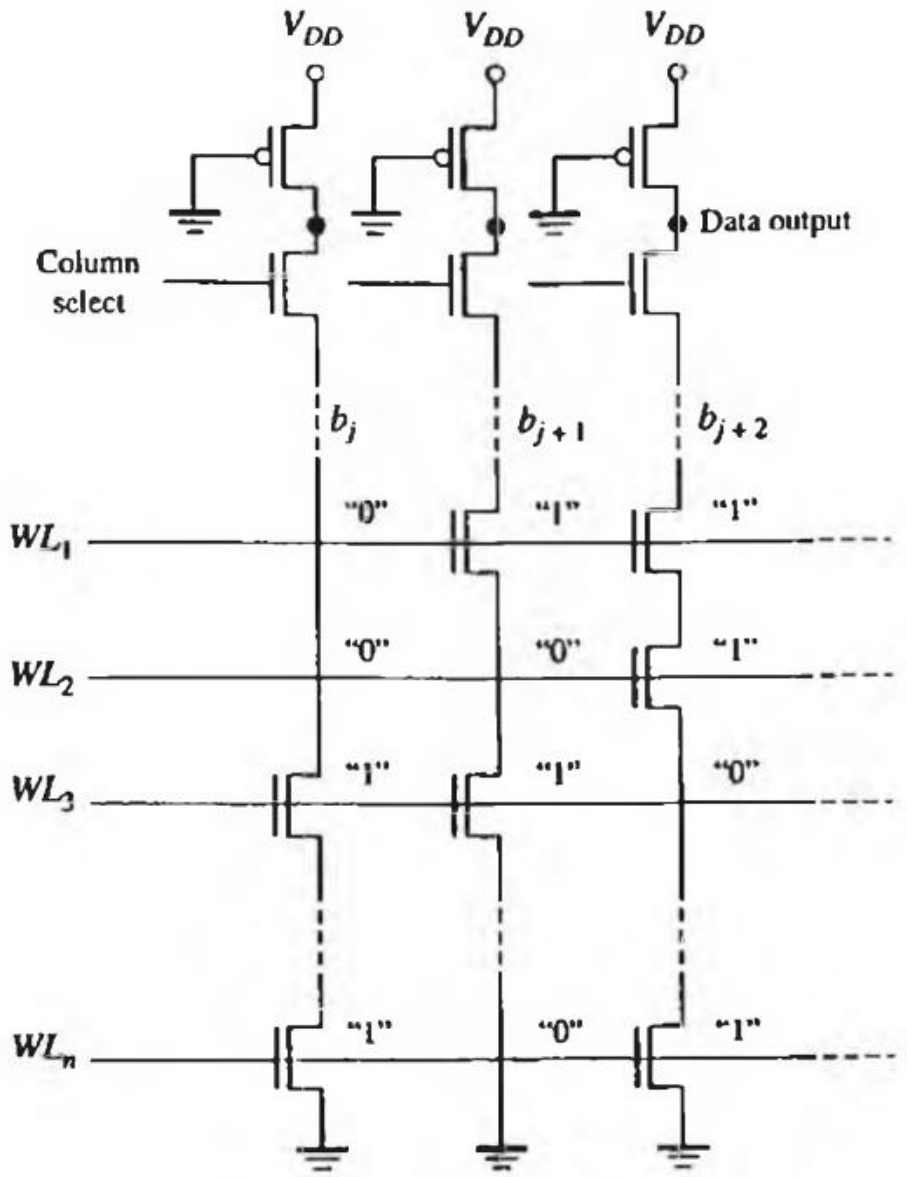
MOS NAND ROM

MOS NAND ROM



All word lines high by default with exception of selected row

NAND Array



(b) NAND Array

NMOS EPROM

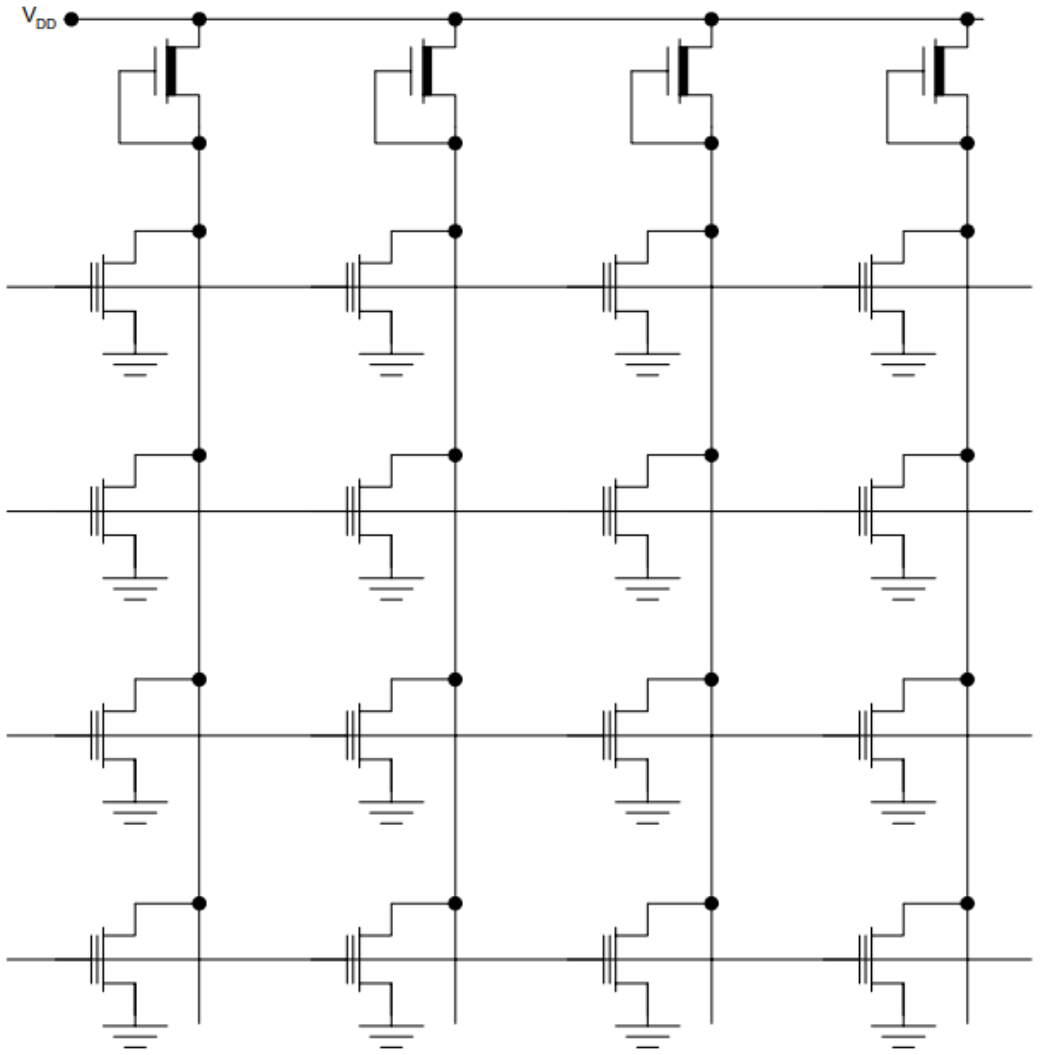


FIGURE 16.16
Erasable programmable read-only memory (EPROM).

EEPROM Circuit

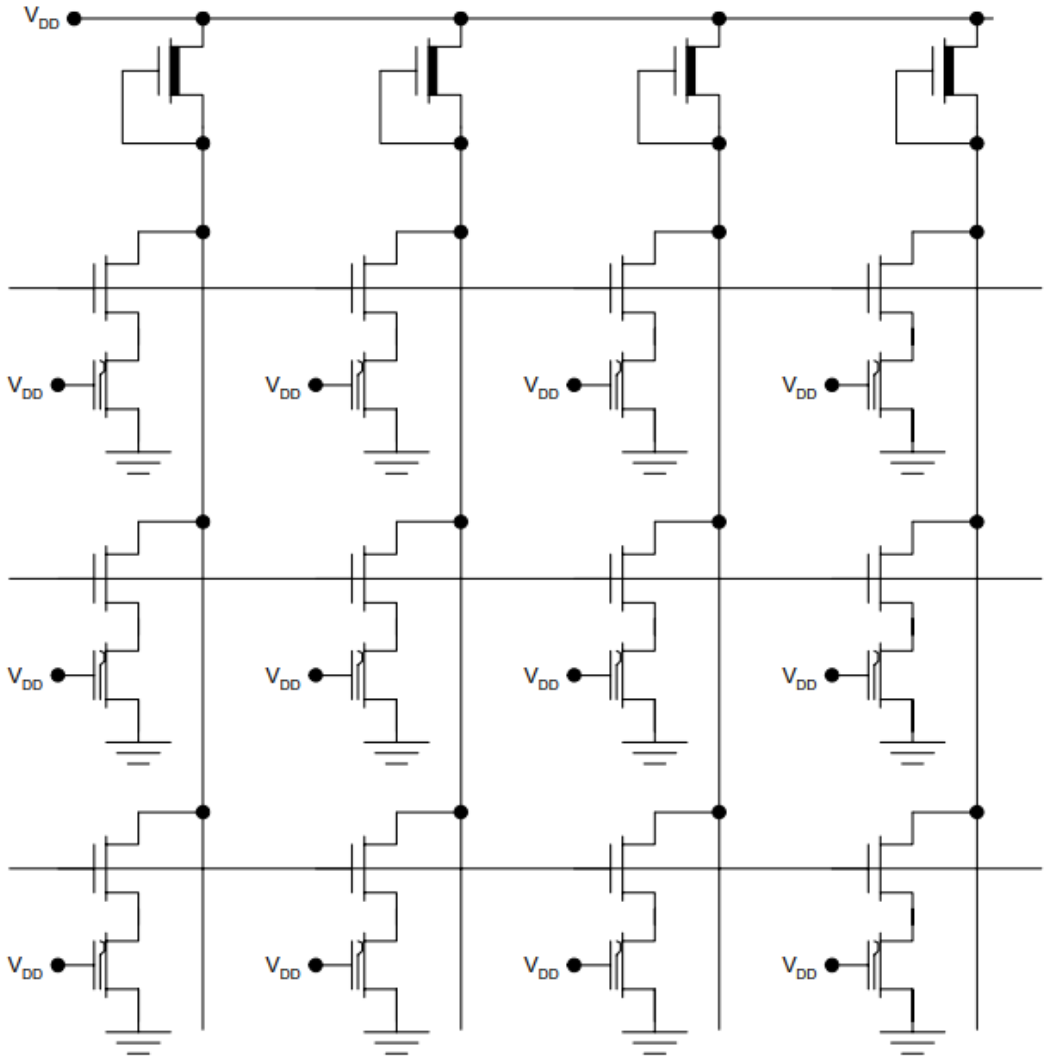


FIGURE 16.18
Electrically erasable programmable read-only memory (EEPROM).

FLASH Memory

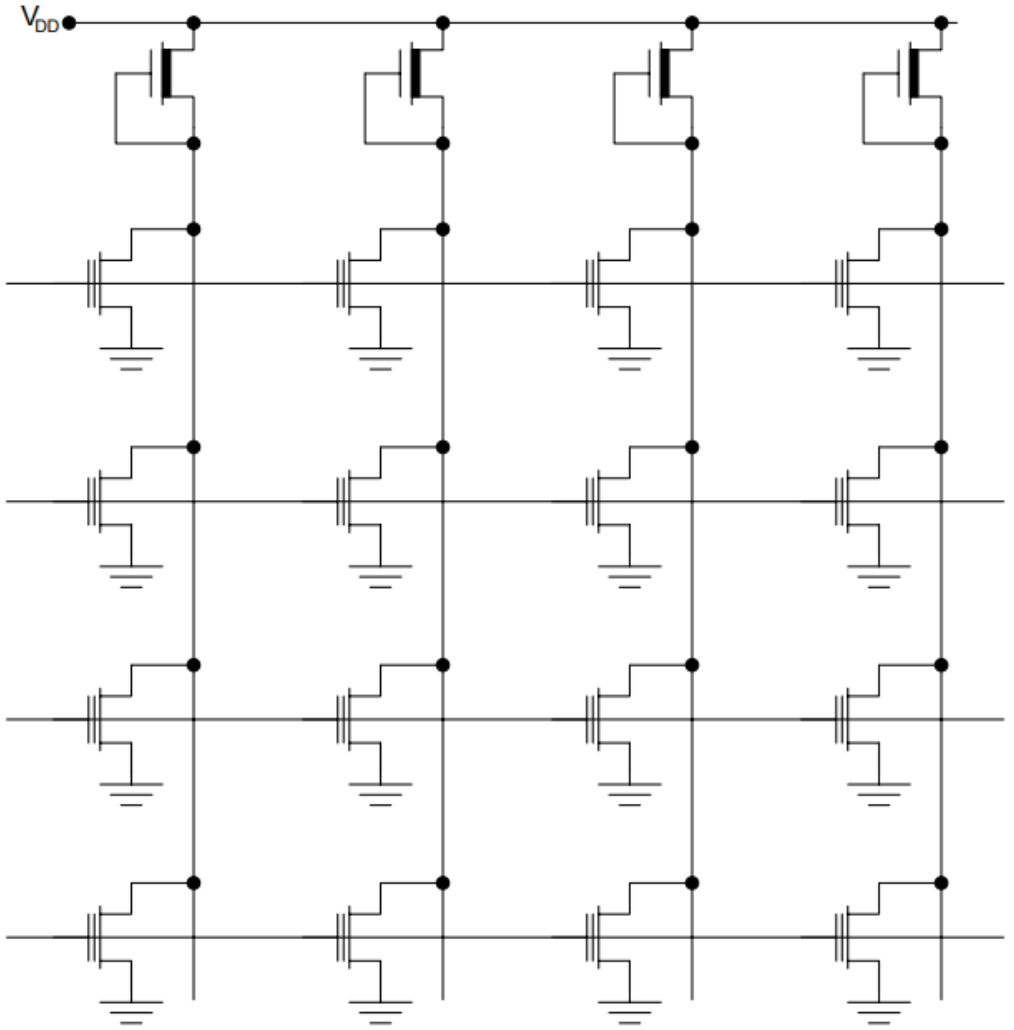
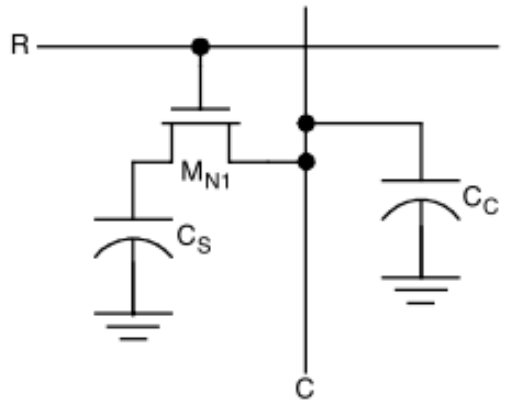


FIGURE 16.19
Flash memory.

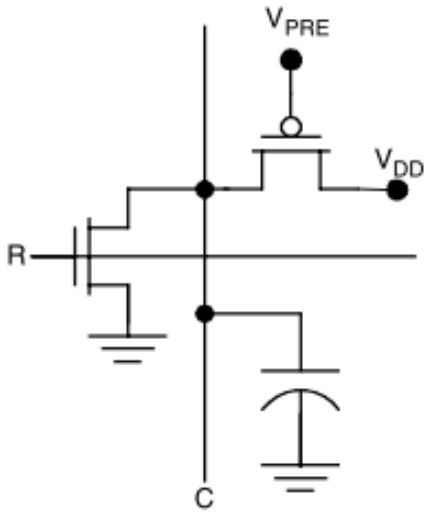
MEMORY CELLS in Different Technologies

DRAM



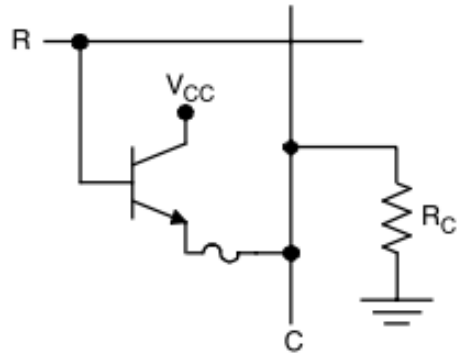
1T DRAM cell

ROM



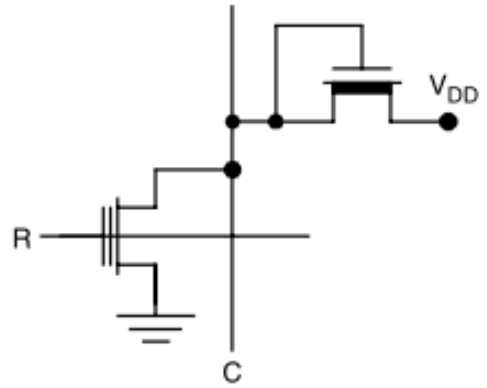
CMOS NOR ROM cell

PROM



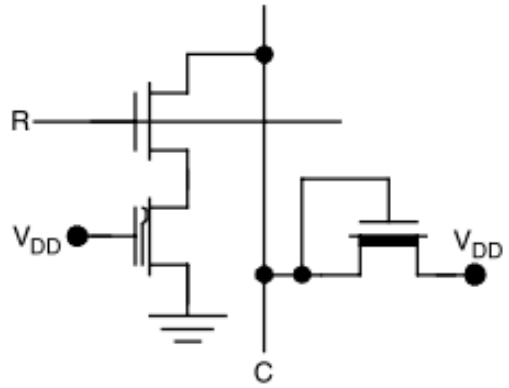
bipolar PROM cell

EPROM



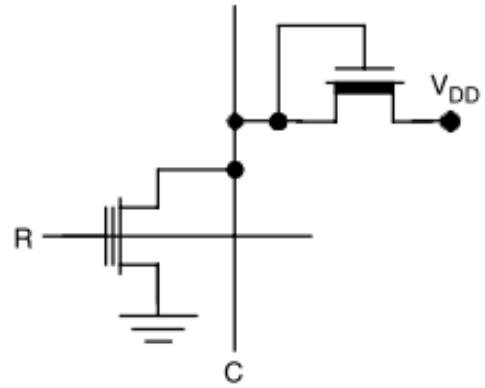
1T cell based on FAMOS transistor

EEPROM



2T cell using FLOTOX transistor

Flash Memory



1T cell based on ETOX transistor

ROM Memory Array

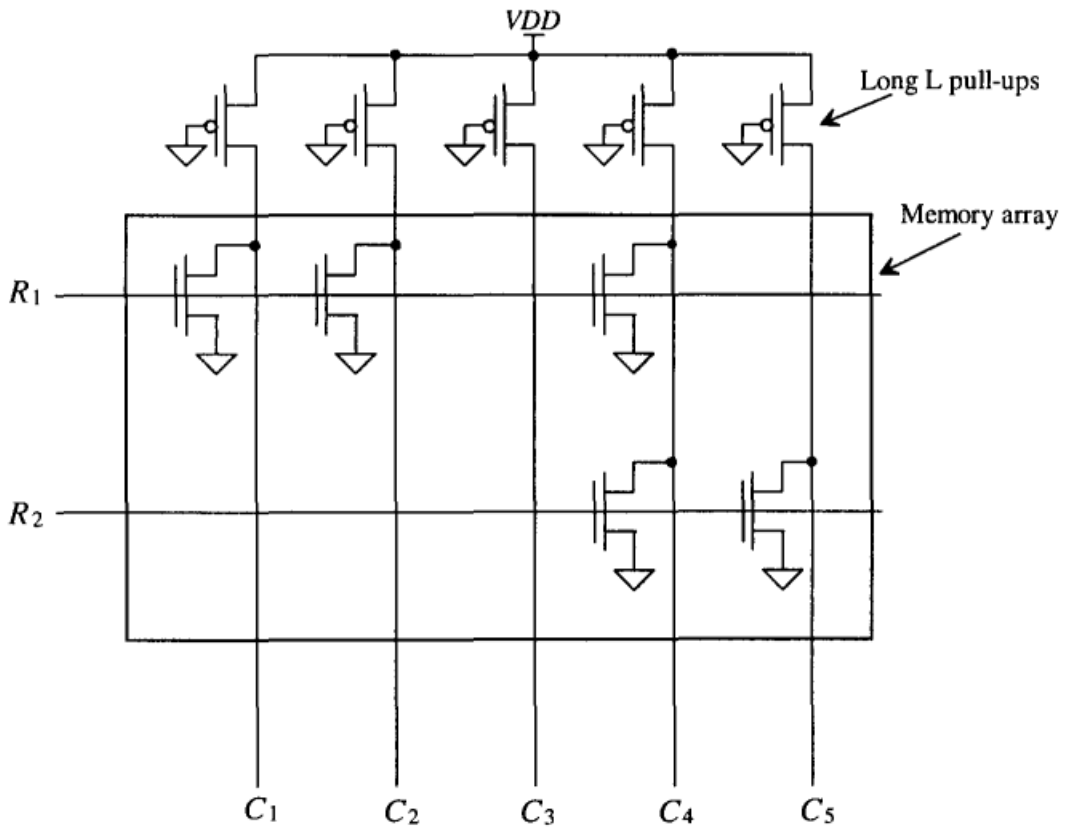


Figure 17.24 A ROM memory array.

Read-Only Memory (ROM)

ROM is the simplest semiconductor memory. It is used mainly to store instructions or constants for use in a digital system. The basic operation of a ROM can be explained with the ROM memory shown in Fig. 17.24. 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. 17.25a). The ROM is programmed (PROM) by cutting (or never making during fabrication) the connection between the drain of the MOSFET and the column line Fig. (17.25b). Because it is not easy to program ROM, it is limited to applications where it is mass produced.

ROM ARRAY ARCHITECTURE

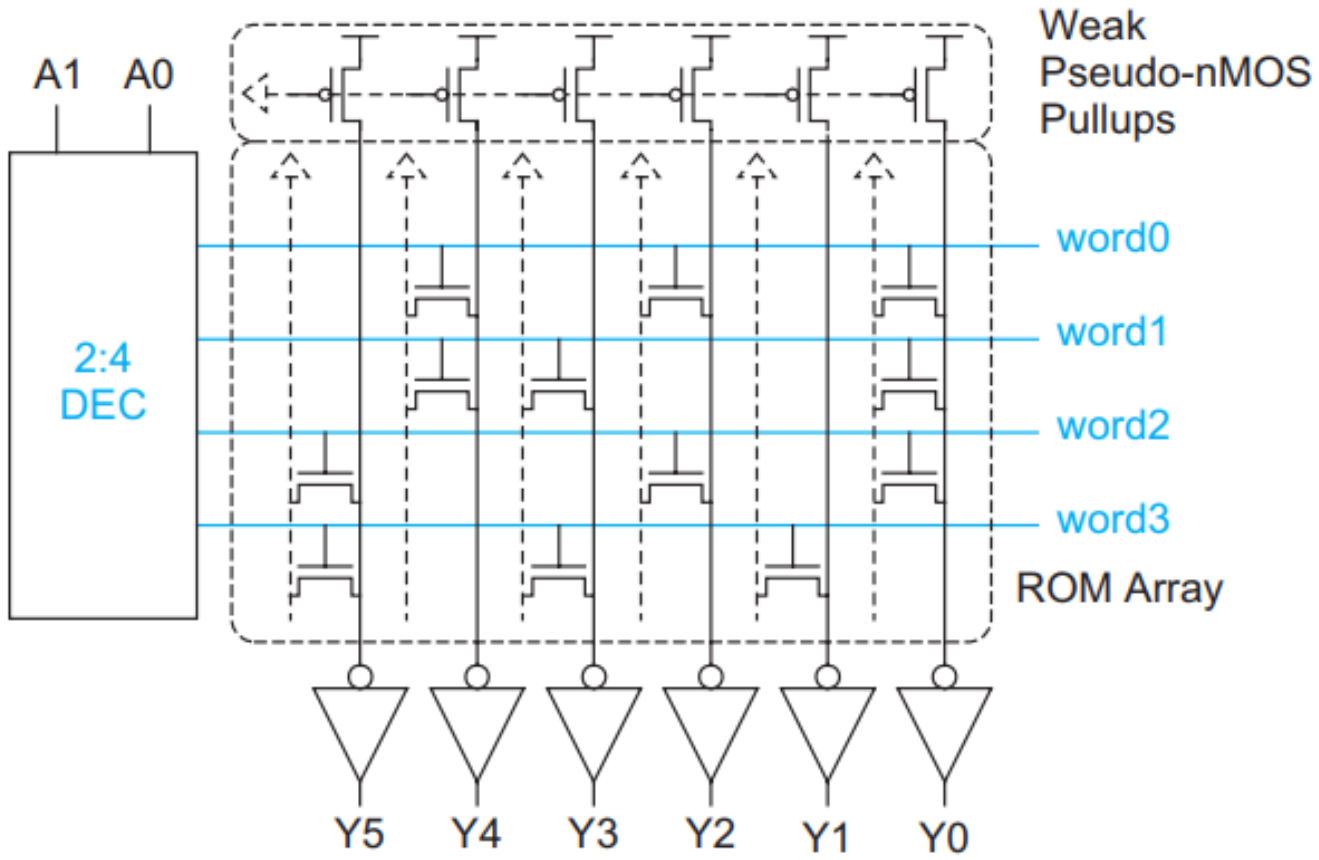


FIGURE 12.52 Pseudo-nMOS ROM

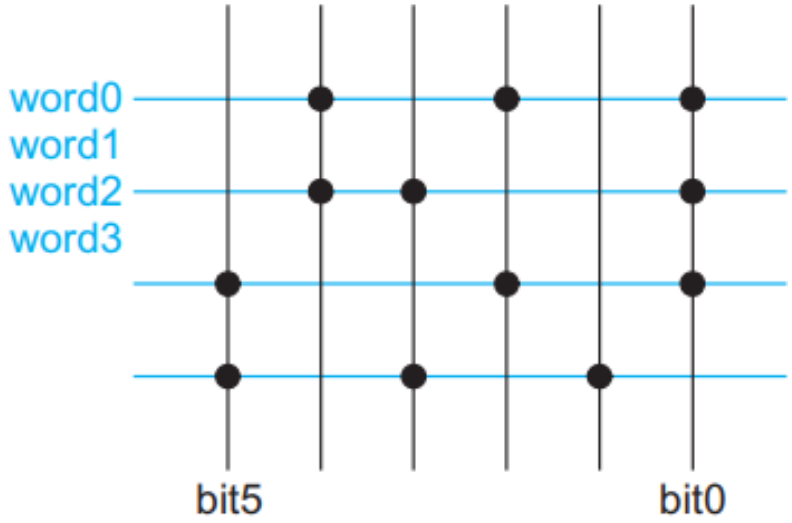


FIGURE 12.53 Dot diagram representation of ROM

NOR Flash Memory Architecture

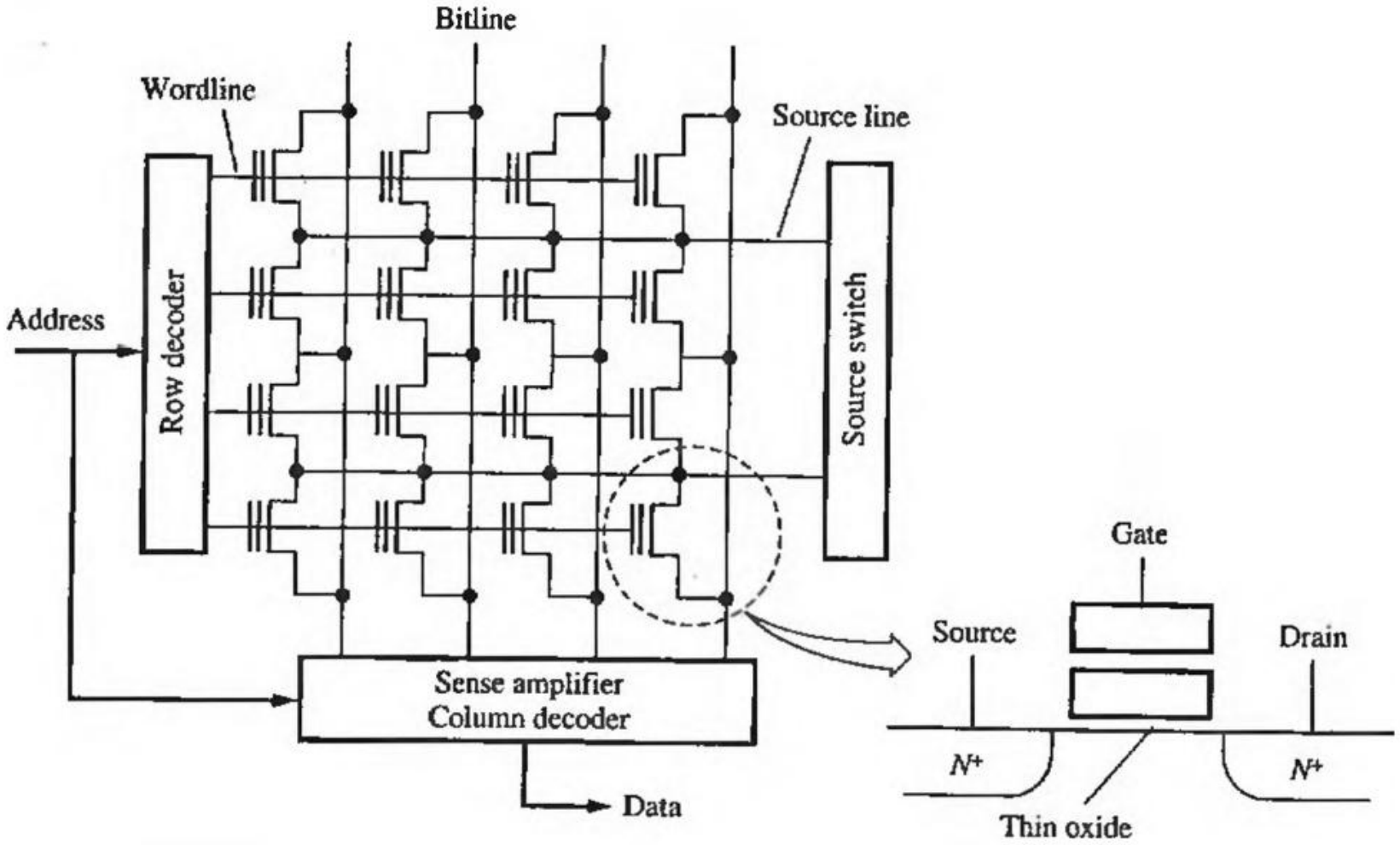


Figure 9.29
NOR Flash memory architecture.

Write NOR Flash

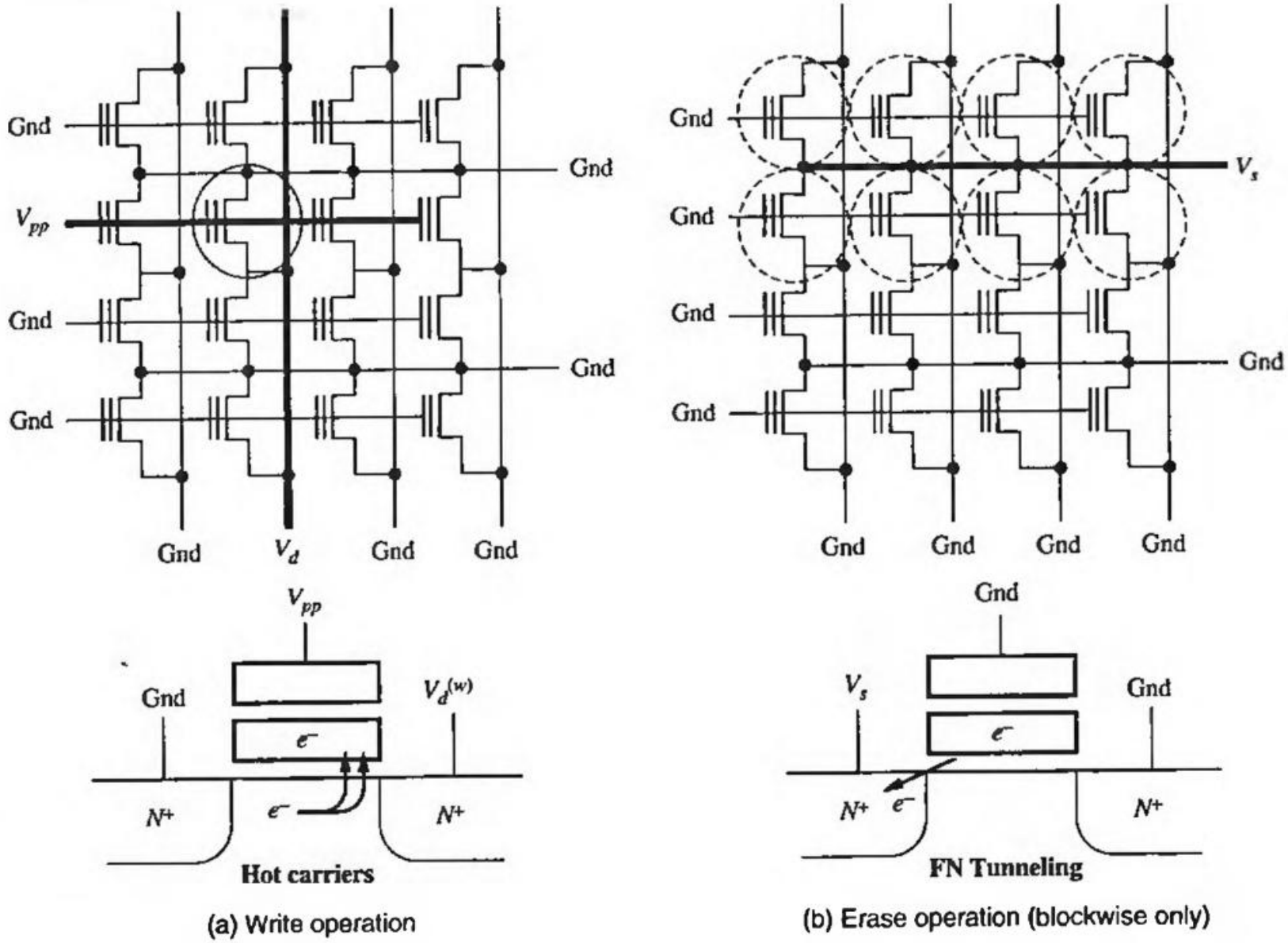


Figure 9.30
Write/erase operations of NOR Flash memory.

PROM READ AND WRITE OPERATION

READ Nor Flash

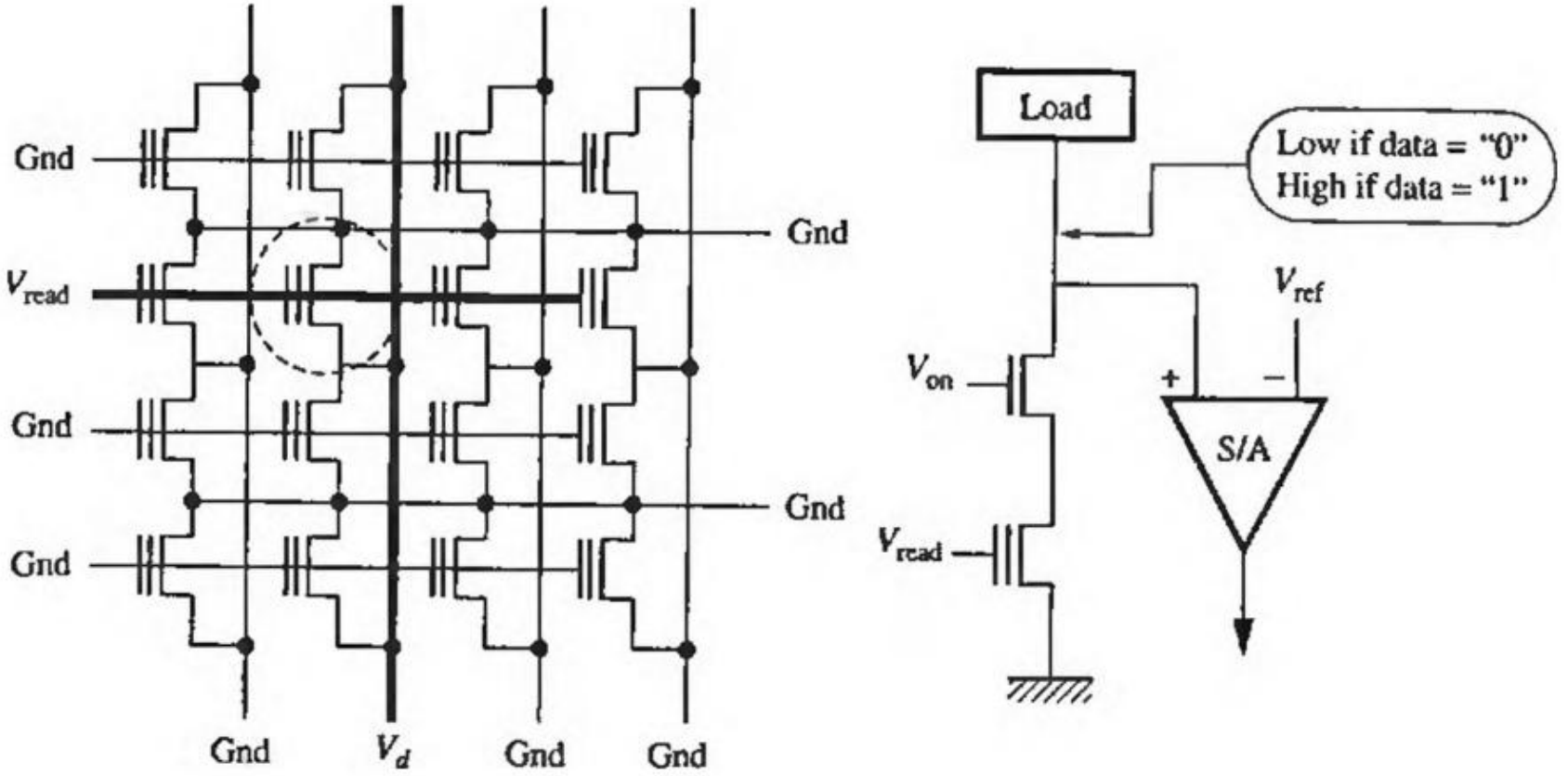


Figure 9.31
Read operation for NOR Flash memory.

Programming FLASH NAND

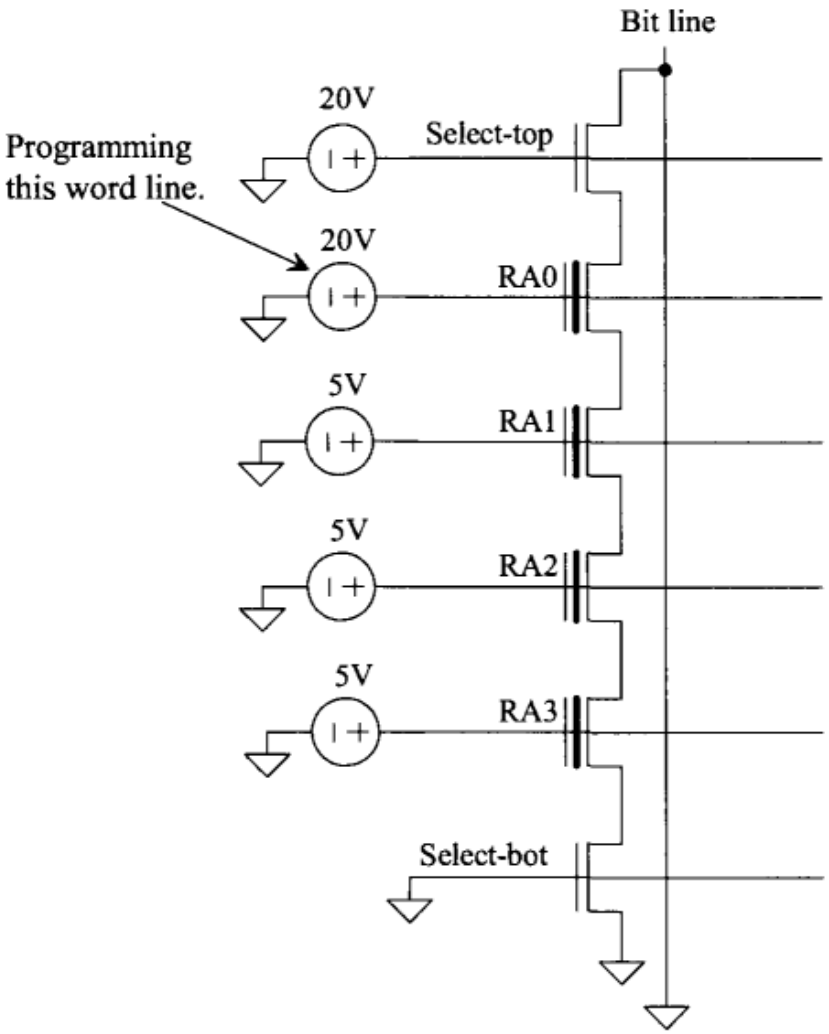


Figure 16.62 Programming in a Flash NAND cell.

Erase and Program Operations

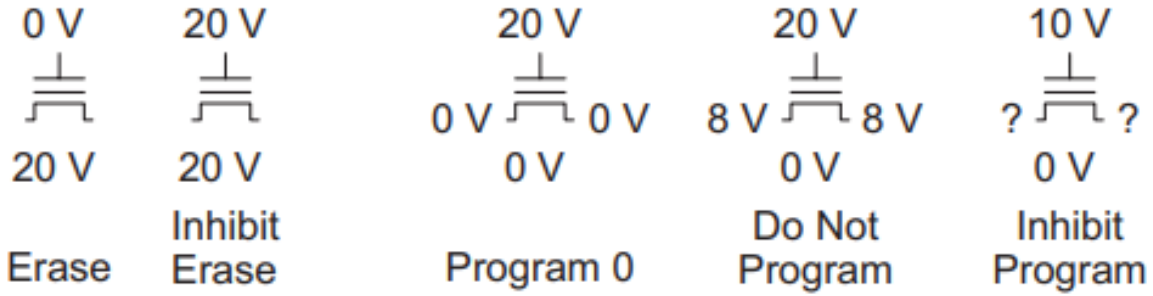


FIGURE 12.61 Erase and program operations

Figure 12.61 shows the operation of the Flash memory using voltages representative of a multimegabit design. The block is erased by setting all of the control gates to GND and raising the substrate to 20 V. The high voltage across the gate oxide induces FN tunneling, causing the electrons to flow from the floating gate to the substrate. At the end of the erase step, all the floating gate transistors have a negative V_t and thus represent 1. Tunneling is a slow process, so block erase takes on the order of a millisecond. The wordlines for other blocks on the chip are set to the same voltage as the substrate to inhibit erasing. An on-chip charge pump (see Section 13.3.7) is used to generate the high voltages.

A cell is programmed (written) to 0 by tunneling electrons onto the floating gate. The programming cannot restore 1 values, so the block must be erased before any cell is reprogrammed. An entire page is programmed at once. To program a page, the bitlines are driven with the data values: 0 V for a logic 0 and 8 V for a logic 1. The substrate is held at ground. The wordline is set to 20 V for the page being programmed and 10 V for the other pages in the block. The ground select line (*gsl*) is left OFF but the string select line (*ssl*) for the block is turned ON, passing the voltage on the bitline to the channels of all the transistors being programmed. Thus, cells being programmed to 0 see 20 V on the control gate and 0 V on the channel. This high voltage difference induces FN tunneling that drives electrons onto the floating gate, raising V_t to a positive voltage. The other cells see a smaller voltage that is insufficient to cause tunneling.

EPROM ERASE STRUCTURE

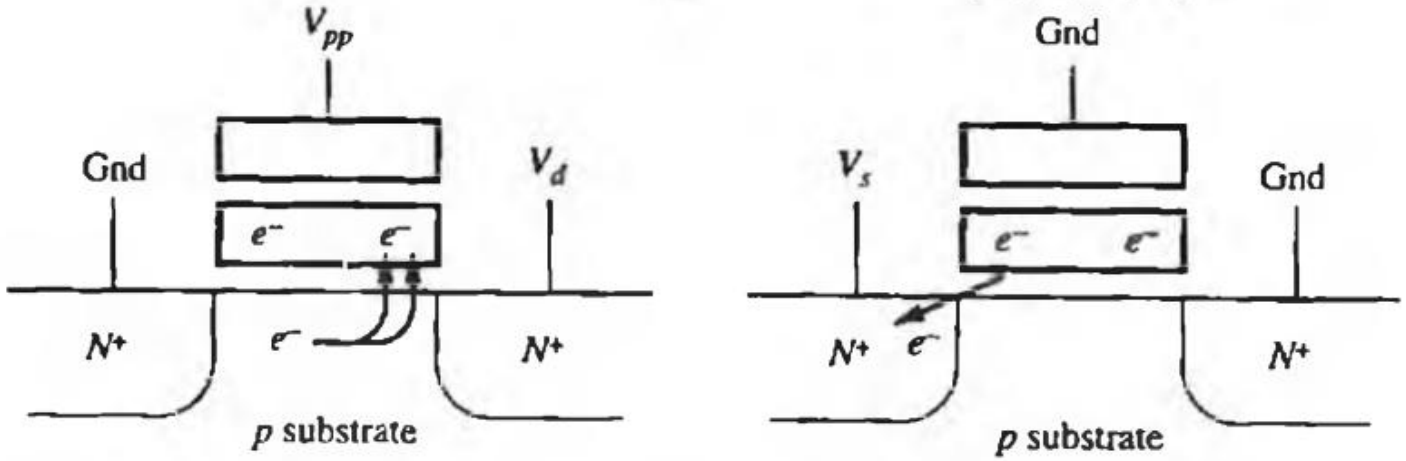


Figure 9.24
E^2PROM write/erase process.

Readings

- Memory arrays are discussed in Chapter 12 of Course textbook