# Chapter 2
# Nonvolatile Memories: NOR vs. NAND Architectures

L. Crippa, R. Micheloni, I. Motta and M. Sangalli

## 2.1 Introduction

Flash memories are nonvolatile memories, i.e., they are able to retain information even if the power supply is switched off. These memories are characterized by the fact that the erase operation (the writing of logic “1”) has to be performed at the same time on a group of cells called a sector or block; on the other hand, the program operation (the writing of logic “0”) is a selective operation during which a single cell is programmed. The fact that the erase can be executed only on an entire sector allows one to design the matrix in a compact shape and therefore in a very competitive size, from an economic point of view. Depending on how the cells are organized in the matrix, it is possible to distinguish between NAND Flash memories and NOR Flash memories. The main electric characteristics are reported below.

For the Table 2.1 the following definitions hold:

- Dword: 32 bits;
- Output parallelism: the number of bits that the memory is able to transfer to the output at the same time;
- Data read/programmed in parallel: the number of addressable bits at the same time during read/program operation;
- Read access time: time needed to execute a read operation, excluding the time to transfer the read data to output.

L. Crippa
Qimonda Design Center, Vimercate, Italy

R. Micheloni
Qimonda Design Center, Vimercate, Italy

I. Motta
Numonyx, Agrate Brianza, Italy

M. Sangalli
Qimonda Design Center, Vimercate, Italy

R. Micheloni et al. (eds.), *Memories in Wireless Systems*,

**Table 2.1** Comparison between NOR and NAND Flash memories

| | NOR | NAND |
|---|---|---|
| Memory size | <= 512 Mbit | 1–8 Gbit |
| Sector size | ~1 Mbit | ~1 Mbit |
| Output parallelism | Byte/Word/Dword | Byte/Word |
| Read parallelism | 8–16 Word | 2 Kbyte |
| Write parallelism | 8–16 Word | 2 Kbyte |
| Read access time | <80 ns | 20 µs |
| Program Time | 9 µs/Word | 400 µs/page |
| Erase time | 1 s/sector | 1 ms/sector |

The aim of this chapter is to explain the reasons why the performances in read program and erase are so different owing to the connection used to create the memory matrix.

## 2.2 The Read Operation in Flash Memories

One of the most important parameters for any kind of memory, not only for Flash memories, is the access time to the data stored in it.

The access time is the temporal interval that passes through any address commutation to the moment in which the addressed data are available to the output. This time is commonly defined as "asynchronous access time." The term "asynchronous" is used to distinguish this kind of read operation from another one, defined as "synchronous" and characterized by the fact that the read data should be synchronized to the output with an external clock.

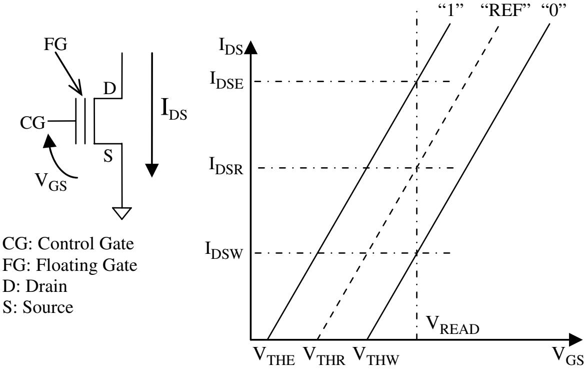
Nowadays, the most commonly used architectures for Flash memories are called NOR and NAND. The common element of both architectures is the nonvolatile (single) cell. The program operation acts on the threshold voltage of the Flash cell, modulating its value and the current/voltage characteristic as a consequence.

The read of a nonvolatile memory cell is done applying convenient voltages to its terminals and measuring the current that flows into the cell. NOR and NAND memories measure this current in different ways. In the following, the measuring method will be discussed separately to better highlight the fundamental differences.

### *2.2.1 NOR Architecture*

In NOR Flash memories, the read of a matrix cell is done in a differential way, i.e., making a comparison between the current of the read cell and the current of a reference cell which is physically identical to the matrix cell and biased with the same voltages $V_{GS}$ and $V_{DS}$.

**Fig. 2.1** Current/voltage characteristics as a function of the threshold voltage of a Flash cell



In the case of memories storing only one bit per cell, the electrical characteristics of the $I_{DS}$-$V_{GS}$ of the written cell (logic "0") and of the erased cell (logic "1") are separated as sketched in Fig. 2.1; this is due to the fact that the two cells have different threshold voltages $V_{THW}$ and $V_{THE}$.

Since the read voltage $V_{READ}$ applied to the control gate is the same, the "written" cell (logic "0") sinks a current $I_{DSW}$ lower than the current sunk by the "erased" cell (logic "1") $I_{DSE}$. To distinguish correctly the two characteristics it is necessary to act on the threshold voltage of the reference cell so that its characteristic is placed between the erased and the written cell characteristics. In this way, if the same $V_{GS}$ is applied to all cells ($V_{READ}$, in Fig. 2.1), the cell will be recognized as erased if its current is higher with respect to the current of the reference cell $I_{DSR}$; vice versa, it will be seen as written. The current of the cells is converted into a voltage by means of a current to voltage converter (I/V), a circuit able to supply to the output a voltage whose value depends on the value of the current at its input. The voltages are then compared through a voltage comparator which gives at its output the cell status: the logic levels "0" or "1" (Fig. 2.2).

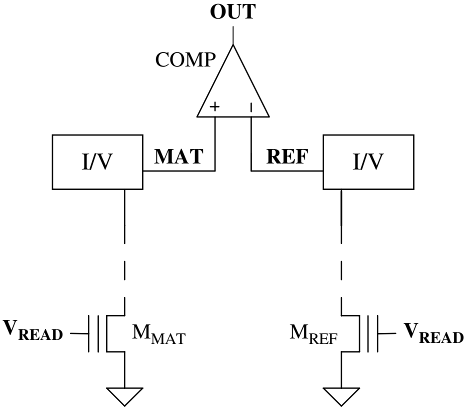


**Fig. 2.2** Block scheme of a circuit used to compare two currents

It is important to avoid the drain of the cells from reaching too high a voltage during the read operation because it may cause a spurious program operation, thus modifying the threshold voltage of the cell. For this reason the drain has to be biased with a voltage lower or equal to 1 V. Therefore, a circuit able to fix the drain voltage at 1 V is needed before the I/V converter.

In most cases the circuitry necessary to execute a read operation (commonly known as *sense amplifier*) is composed of the following fundamental blocks:

- I/V converter;
- drain voltage limiter (this voltage is usually about 1 V);
- output comparator.

One of the simplest circuit implementations of a sense amplifier is sketched in Fig. 2.3: on the left hand side there is the matrix cell ($M_{MAT}$), while on the other side there is the reference cell ($M_{REF}$). The same voltage ($V_{READ}$) is applied to the gates of both cells, while on their drains the voltage limiter ($V_{DS}$REG) is placed in order to limit the drain voltage. The voltage limiter is designed using a NMOS transistor and biasing its gate with a fixed voltage. The I/V converter is realized using a resistive load, and its output nodes (MAT and REF) are compared by the voltage comparator COMP.

Over the years, new and more complex circuits have been realized to improve the behaviour of the sense amplifier and to reduce the access time.

For example, the drain regulator is designed using closed loop structures, while the I/V converter is realized using current mirrors or active loads [1].

It is possible to have a sensible improvement using the so-called equalization technique: before the comparison takes place, the nodes MAT and REF are

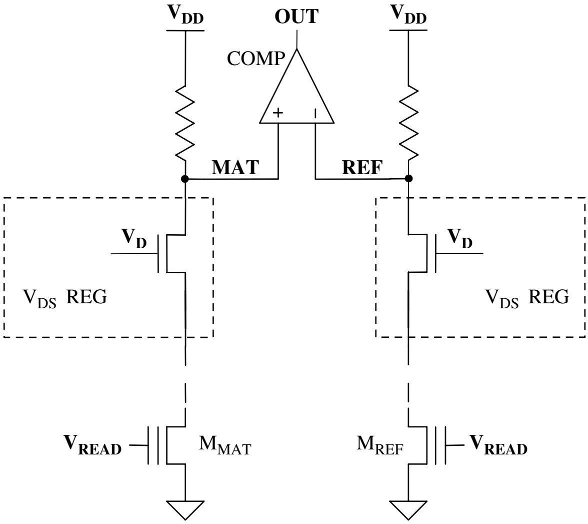


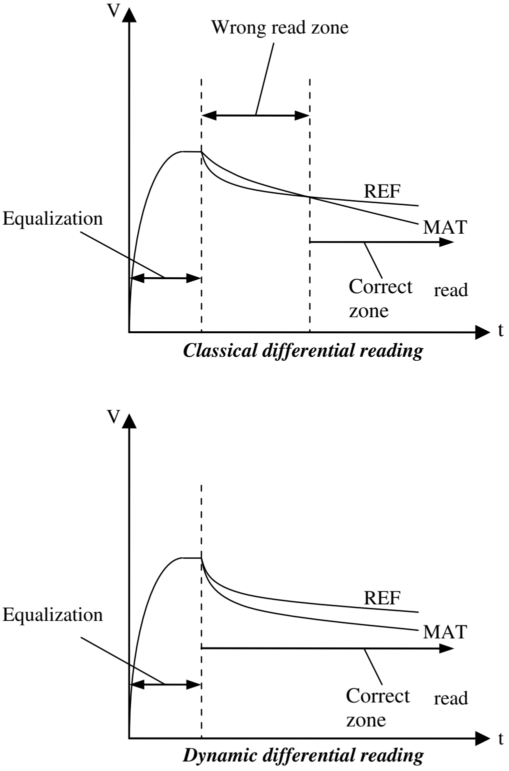
**Fig. 2.3** Basic scheme of a sense amplifier

"equalized" (i.e., they are forced) to the same value. In this way the node MAT is always near to the final voltage value.

A further equalization technique can be used to increase the read speed. This kind of equalization, that seems simple from a circuital point of view, consists in using a number of reference cells equal to the number of cells which are read at the same time [2]. To gain a real benefit from this equalization method, it is very important for the reference cell to have the same load condition as the matrix cell. For this reason, the capacitive load of the matrix cell is recreated on the drain of the reference cell; a real bitline is usually used to have a complete matching.

Using the above described architecture, it is possible to read in a dynamic differential way, obtaining a correct separation in voltage of nodes MAT and REF, as soon as the equalization phase is finished: being equal to the sensibility of comparator COMP, a faster read is performed than with the classic approach, where it is necessary to wait until the nodes MAT and REF are stable.

In Fig. 2.4 the evolution of nodes MAT and REF is shown in the case of a simple differential read operation and in the case of a dynamic differential read operation. Typically, during the read operation, the time dedicated to the comparison of the currents is about 10–20 ns.



*Classical differential reading*

*Dynamic differential reading*

**Fig. 2.4** MAT and REF node transients in classical and dynamical differential reading

### 2.2.2 NAND Architecture

In the NAND Flash architecture, the cells are connected in series, in groups of 16 or 32. Two selection transistors are placed at the edges of the stack, to ensure the connections to ground (through $M_{SSL}$) and to the bitline (through $M_{DSL}$).

This basic structure is shown in Fig. 2.5. When a cell is read, its gate is set to 0 V, while the other gates of the stack are biased with a high voltage (typically 4–5 V), so that they work as pass-transistor, regardless of their threshold voltage.

An erased NAND Flash cell has a negative threshold voltage; on the contrary, a programmed cell has a positive threshold voltage but, in any case, less than 4 V. In practice, driving the selected gate with 0 V, the series of all the cells will sink current if the addressed cell is erased, otherwise no current is sunk if the cell is programmed.

Figure 2.6 shows the threshold voltage distributions $V_{TH}$ for the erased and programmed memory cells; note that, for gate voltages above the right margin of the programmed distribution ($V_{THSD}$), the cells always sink current whatever their threshold voltage is. This feature is used particularly when the cell should operate as a pass-transistor.

Unlike NOR Flash memory, the current to be sensed in these serial structures is very low. This value of current is typically 200–300 nA (tens of μA in NOR architecture). It is unfeasible to detect such current with a differential structure as in the previous section.

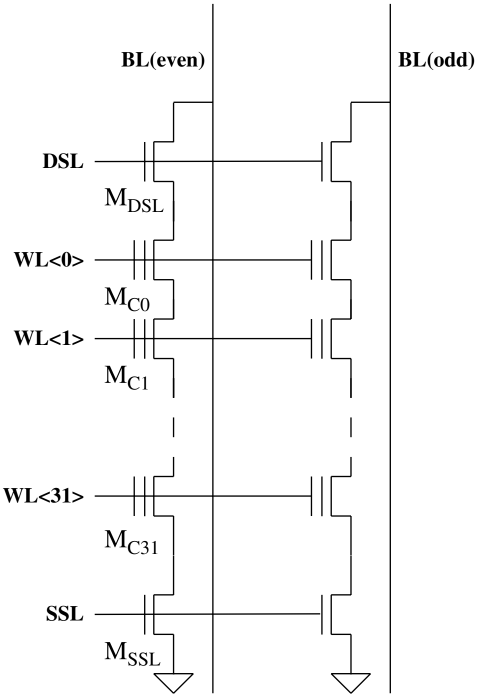
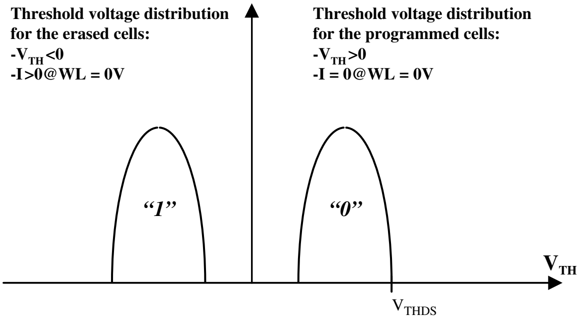


**Fig. 2.5** Matrix structure in NAND architecture

**Fig. 2.6** Threshold voltage distributions for erased and programmed cells



The reading method in NAND memories is the charge integration, which uses the parasitic capacity of the bitline. This capacitance is precharged to a fixed value (typically 1.2 V): if the cell is erased, it sinks current and discharges the bitline; otherwise, if it is programmed, it does not sink current and the bitline keeps its initial value. There are many circuits to detect the charge status of the bitline parasitic capacitance: they can be summarized with the structure shown in Fig. 2.7a. The bitline parasitic capacitance is indicated with $C_{BL}$; the electric characteristic of the NAND string is summarized with a current generator ($I_{CELL}$).

During the bitline precharge, the gate terminal of $M_P$ is kept at GND (0 V), while the gate of $M_N$ is at a fixed value $V_1$ (for example, 2 V). At the end of the charge transient, the voltage $V_{BL}$ on the bitline is:

$$V_{BL} = V_1 - V_{THN} \tag{2.1}$$

where $V_{THN}$ represents the threshold voltage of the n-channel transistor $M_N$.

The bitline precharge phase usually lasts 2–6 µs, in order to reduce the current consumption peak from the power supply VDD. The voltage $V_{OUT}$ is initially precharged at VDD. After the precharge phase, $M_N$ and $M_P$ are turned off, leaving OUT and BL nodes floating (high-Z status), i.e., charged to their precharge value.

After the precharge phase, the "evaluation phase" begins, where the cell current is checked. If the cell does not sink current, the bitline capacitance keeps its precharged value; if the cell sinks current, the bitline begins to discharge. At the end of $T_{VAL}$

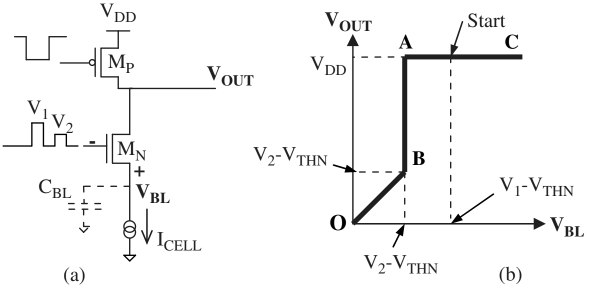


**Fig. 2.7** (**a**) Structure to detect the bitline discharge and (**b**) $V_{OUT}$ characteristic as a function of $V_{BL}$

(we will see later what its value is and what factors it depends on), the gate of the n-channel transistor $M_N$ is biased with a value $V_2 < V_1$, typically 1.4–1.6 V.

If the $T_{VAL}$ time lasts long enough to discharge the bitline under the value:

$$V_{BL} < V_2 - V_{THN} \tag{2.2}$$

then $M_N$ turns on, equalizing the voltage $V_{OUT}$ to the bitline level $V_{BL}$.

Figure 2.7b shows $V_{OUT}$ voltage as a function of $V_{BL}$ when $V_2$ is applied to $M_N$: the point referred to as “Start” corresponds to the end of the phase of bitline precharge and to the beginning of the evaluation phase. The segment referred to as AC corresponds to the cell that does not sink current or to a cell that did not have enough current to discharge the bitline below the value ($V_2$–$V_{THN}$) during $T_{VAL}$. If the cell has discharged the bitline to a value less than ($V_2$–$V_{THN}$), then OUT is shorted to the bitline and assumes the same value (segment OB in Fig. 2.7b). Obviously, the transition between points A and B is not so sharp: in actuality, it shows a slope that depends on the relationship between the bitline capacitance and the capacitance of OUT (this ratio is in the order of 30–100).

What is the minimum $T_{VAL}$ time for the bitline to be discharged? This time depends on the bitline capacitance value, the minimum cell current, and the difference between $V_1$ and $V_2$:

$$T_{VAL} = C_{BL}\,(V_1 - V_2)\,/I_{CELL} \tag{2.3}$$

Typical values for the bitline capacitance are 2–4 pF, while the cell, as we said, can sink currents of about 200 nA. The difference between $V_1$ and $V_2$ is about 500 mV, so it follows that the evaluation time may vary from 5 to 10 µs. Since the bitline capacitance and the cell current cannot be modified (they depend on the manufacturing technology), one could think of reducing the difference between $V_1$ and $V_2$ to reduce the evaluation time. This difference is designed to mask side effects during the evaluation, such as disturbances on the bitline voltage or spurious sinking superimposed to that of the cell (leakage currents); these effects may invalidate the reading result.

The output voltage is “digitized and frozen” in simple latch structures. An implementation of this circuit is shown in Fig. 2.8.

At the beginning of the read operation, the DATA_N node is forced to ground through the $M_{SET}$ transistor. The portion of the circuit outside the LATCH box corresponds to Fig. 2.7a, which has already been analyzed. At the end of the evaluation phase, the $V_{OUT}$ voltage value can be either at VDD (programmed cell) or at a value corresponding to the segment OB of Fig. 2.7b (erased cell); this voltage biases the gate of the $M_{SEN}$ transistor.

Let us see what happens if the READ signal is set to VDD level. If $V_{OUT}$ is equal to VDD, the series $M_{SEN}$-$M_{EN}$ sinks current, thus discharging the voltage on DATA to ground. Otherwise, if $V_{OUT}$ has been discharged, the series $M_{SEN}$-$M_{EN}$ can’t change the latch status, which remains at its initial state.

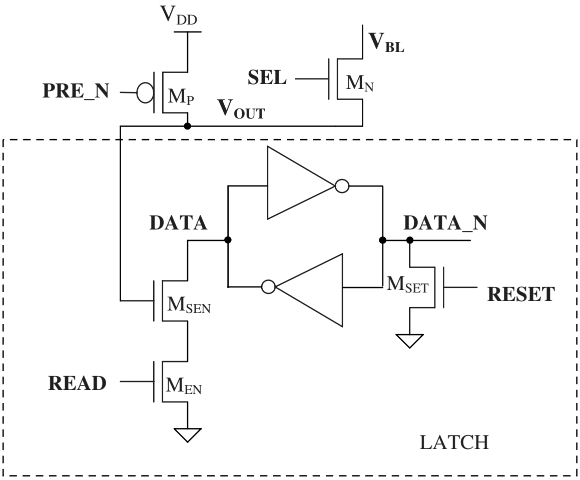


**Fig. 2.8** Read circuit for NAND-architecture Flash memories

In conclusion, the differences between the two architectures force the use of different reading techniques: differential reading in NOR architecture, charge integration reading in NAND architecture. These two modes affect the reading timing: a few tens of nanoseconds for NOR architecture, a few tens of microseconds for the NAND one.

## 2.3 Program Operation in Flash Memories

The "program operation" is the writing of the information in a memory cell and it is usually performed by transferring the electrons from the substrate of the cell into its floating gate; in this way the threshold voltage of the cell is increased.

The number of programmed bits per second is greater for a NAND memory array than for a NOR one, due to the fact that NAND and NOR memories use two different physical mechanisms to perform this operation.

The aim of this section is to describe how the program operation takes place in NOR and NAND memories, trying to explain why a different mechanism is chosen and its impact on performances.

### *2.3.1 Program in NOR Memories*

The writing of information in NOR memory cells takes place through the channel hot electron mechanism.

Applying a voltage difference between the source and the drain of a cell, a longitudinal intensive electric field is created, which causes the electrons to knock against each other. In this way, they acquire energy greater than the energy they would have at the thermal equilibrium in the silicon lattice. The greater part of the generation of hot electrons takes place in the region where the electric field is more intensive, i.e., the depletion region near the drain.

In this condition some electrons acquire enough energy (greater than 3.1eV) to overcome the potential barrier at the channel-oxide interface.

Applying a voltage to the control gate, a transversal electric field is then created in order to support the injection of electrons from the channel to the oxide and their gathering into the floating gate.

In any case, the injection of electrons comes naturally to an end, because the increasing of the negative charge in the floating gate causes a continuous decrease in the gate potential, and therefore the electrons are less and less attracted.

The program operation through hot electrons is a fast operation, but the current necessary for the program to take place is very high. Obviously, the greater the number of cells to be programmed at the same time, the greater is the current consumption. For example, to program 64 cells, it is necessary to produce on chip 3.2 mA, supposing that the current consumption of a single cell is about 50 µA. Therefore the programming of a huge number of cells using this kind of mechanism becomes an expensive operation, especially concerning the area required to design the peripheral circuits needed to create such high currents.

The voltages applied to the cell during the program operation are:

- 4.5 V on drain;
- 9 V on gate;
- 0 V on body and source.

Actually, the voltages applied to the cell terminals depend on the technological node. The cell should have a good speed during the program operation and a good reliability as regards parasitic effects such as the drain turn on, the snap back, the soft programming during the read operation, and the soft erasing during the erase operation. Therefore, in the choice of voltages to be applied, it is necessary to consider aspects as the channel length, the program efficiency of the cell, the drain current, and the process variation. Moreover it is very important to apply the voltages with high precision; time conditions being equal, a variation in the value of the voltages applied during the program operation may cause a variation in the drain current and therefore a variation in the cell's capacity to accumulate the electrons.

Last but not least, it is very important to follow a precise sequence to bias the cell terminals; once the voltage has been applied to the gate of the cell, the drain will be biased. This choice depends on the cell matrix structure; particularly sensible cells with 4.5 V on drain and 0 V on gate might present snap back effects or undesired program/erase effects.

A distribution of threshold voltage is obtained as result of a modify operation (program or erase operation) executed on a number of cells. This is due to different factors such as process variation, dissymmetrical geometry, power supply variation, and source and drain modulation. Despite all these effects, it is important that the distributions have a precise and well controlled width so that they can be correctly allocated in the working window of the memory cells. This requirement has been highlighted with the introduction of multilevel memories, i.e., memories in which there is more than one bit per cell. As the number of bits to be stored in the memory cell increases, the number of distributions to be allocated in the working window increases as well. In fact, due to technological and reliability constraints, usually the designer cannot enlarge the working window. For example, if there are three bits to store, than eight distributions must be allocated.

The width of the distributions may be controlled by choosing an appropriate program algorithm. For example a *program and verify* approach can be chosen, where the program pulse is followed by a verify operation. The verify operation consists in controlling the cells under program if they have reached the target threshold voltage. In the NOR memories, the threshold voltage of the memory cell is compared to the threshold voltage of a reference cell, as in the read operation. But there are two factors that make the verify operation different from the read operation: the reference cell used, and the time necessary to execute the comparison. Usually all the cells are overprogrammed with respect to the read reference voltage, so that there is a margin during the read operation. Obviously it is not possible to take great margins, especially when many distributions must be allocated in the same working window; therefore, in order to guarantee a greater precision to the verify operation, the timing is relaxed compared to the one used during the read operation. The cells that result as programmed after the verify operations (i.e., that reach the desired threshold voltage) are left in that position, while another program pulse is applied to the other cells. Either when all the cells are programmed, or when all the attempts to program the cells have been made, the algorithm finishes. While in the first case the operation finishes with success, in the other case the “fail” information is communicated to the user.

If a cell should not be programmed, it is necessary to avoid the applying of high voltages on gate and drain at the same time. Due to the memory architecture, there will be some cells with a high voltage on the gate but 0 V on drain, and some other cells with a high voltage on drain but 0 V on gate. These cells may suffer from different disturbances. In Fig. 2.9, a little matrix is sketched where the potentials for the cell under program and for the cells sharing the same bitline and the same wordline are stressed. The cells sharing the same bitline suffer the so called “drain disturb”—owing to the potential applied on the drain, the already programmed cells (therefore with a negative charge collected in the floating gate) may show a loss of charge. On the contrary, the erased cells sharing the same wordline may suffer a program operation by Fowler–Nordheim tunnelling. Therefore, they can show, at the end of the program operation, a greater threshold voltage. For the programmed cells sharing the wordline with the cell under program, an injection of electrons from the floating gate to the control gate may take place, and these cells will show, at the end, a lower threshold voltage.

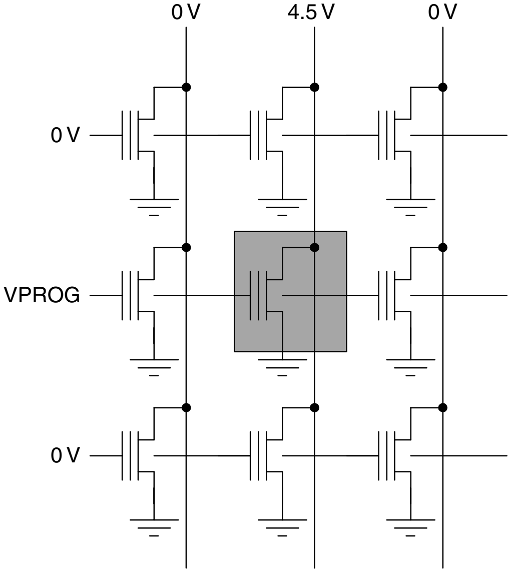


**Fig. 2.9** A NOR memory architecture showing the biasing of wordlines and bitlines during the program operation; the cell under program is the highlighted one

### *2.3.2 Program in NAND Memories*

As already mentioned above, the program operation in NAND memories takes place thanks to a different physical principle: it is exploiting the quantum effects of tunnelling of electrons in the presence of a high electric field. In particular, the operation depends on the polarity of the electric field: if it is directed from substrate to gate, than a program of the cell is obtained; an erase operation occurs if the polarity of the electric field is the opposite one.

In reality the tunnelling effects may be two:

- the *channel tunnelling*, where the electric field is applied between gate and substrate, while the drain and source terminals of the cell are floating;
- the *junction tunnelling*, where the electric field is applied between the gate and one of the two terminals (drain or source).

The effect used in NAND memories is the channel tunnelling. In Fig. 2.10 it is possible to see the modifications of the potential barrier: the potential barrier of the oxide insulating the floating gate from the substrate during the read operation, and how it is modified during the program operation due to the intensive electric field applied.

During programming the number of electrons that pass through the tunnel oxide depends on the electric field: the greater the electric field, the greater is the probability of the injection of electrons. In order to improve the program performance it is necessary to have high electric fields and high voltages. This requirement is one

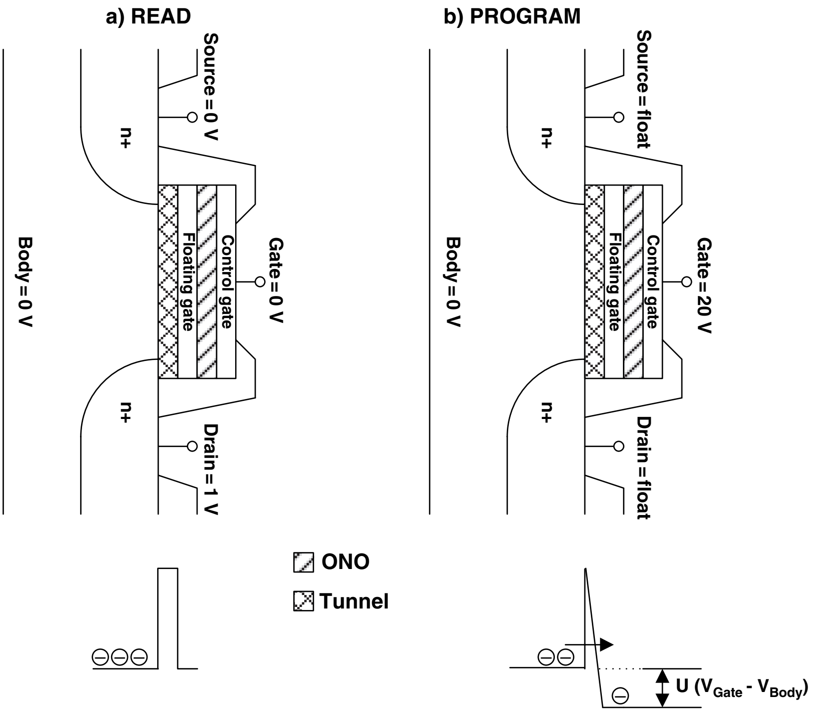


**Fig. 2.10** Biasing voltages for a NAND cell memory in case of (**a**) a reading operation, and (**b**) a programming operation and relative band diagram

of the main disadvantages of this method of programming, because damages of the tunnel oxide are due to these high voltages.

A reduction of the thickness of the dielectric seems to be the best solution to this problem. In fact, in this way, the injection efficiency improves, while the voltages necessary to obtain the program and the erase of the cells (and therefore the total energy of electrons crossing the oxide) decrease. Unfortunately, if the dielectric is too thin other negative effects may take place, such as the *stress induced leakage current* (SILC).

Another disadvantage of the tunnelling method for programming is the time required, which is typically longer than in the channel hot electrons case.

On the other hand, the main advantage is the current required for this operation, which is rather contained (on the order of a nanoAmpere per cell). This characteristic makes the Fowler-Nordheim method the right one to program in parallel a great number of cells.

As with the cells of a NOR memory, the algorithm used to program the cells in a NAND memory is a *program and verify* algorithm (after a program pulse, a verification on the threshold voltage of the cell is done). In the NAND memories, as in NOR ones, the verify threshold voltage used is higher than the one used during the read operation, in order to gain a margin and to guarantee a correct allocation of the distributions.

In a NAND memory, a cell is a part of a string of cells, and this string can be selected by drain and source selectors. Let's consider the string in which we want to program a specific memory cell. The drain selector is biased to VDD, the cells of the strings which should not be programmed are placed at 8–10 V, the gate of the source selector is at 0 V, and the bitline is biased at 0 V. The gate of the cell under programming ranges from 15 to 20 V (it has 0 V on drain, and its source is floating, while there is a high voltage applied to its gate).

At this point it may be interesting to understand how it is possible to prevent cells sharing the same wordline with the cell to be written from suffering an undesired program. For example, it is possible to bias the drain of these cells with a high voltage so that the channel voltage of the cells increases. Therefore the probability for the electrons to pass from the substrate to the floating gate is reduced.

Indeed, it is very difficult to implement this method if the target is to realize memories of a certain size. Let's consider a 32 Mbit memory with a matrix organized in 8 k (8128) rows and 4 k (4096) columns. When we program the selected 2048 cells at the same time, the drains of the other 2048 cells have to be biased with a high voltage in order to avoid spurious program pulses on them.

The disadvantages of this method are evident:

- area occupation due to the big high voltage circuitry (the parassitic capacitance of the unselected bitlines must be charged);
- sensing circuit able to manage high voltages;
- time needed to precharge all the bitlines.

As an alternative, the so-called *self-boosting* mechanism [3] can be used (Fig. 2.11). To increase the channel voltage, the self-boosting exploits the high voltages involved in the operation, the capacitances of the oxide between control gate and floating gate ($C_{ONO}$), the capacitance of the tunnel oxide ($C_{tun}$), and the capacitance of the channel ($C_{ch}$).

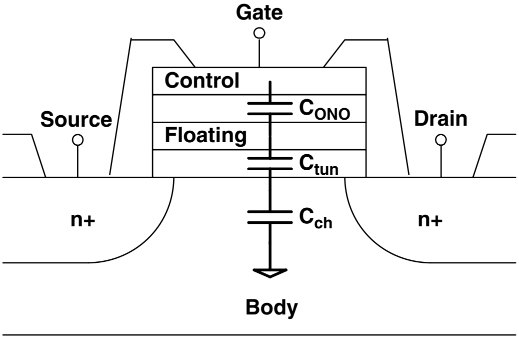


**Fig. 2.11** Capacitors involved in the channel boosting

In particular, the unselected bitlines are precharged to a $V_{pre}$ voltage, and then left floating. When the unselected wordlines are biased to the $V_{gate}$ potential, the channel of the cells of the unselected bitlines is boosted to the voltage:

$$V_{chunsbl} = V_{pre} + V_{gate} \cdot C_{ins} / (C_{ins} + C_{ch}) \tag{2.4}$$

where $C_{ins}$ is the parallel of the tunnel oxide capacitor and the ONO capacitor:

$$C_{ins} = (C_{tun} \cdot C_{ONO}) / (C_{tun} + C_{ONO}) \tag{2.5}$$

Now, knowing this mechanism, it is useful to reconsider the voltages applied to both the selected and unselected wordlines (Fig. 2.12).

The bitlines of the selected cell are biased at 0 V, as mentioned above, even if the terminals of source and drain can be floating according to the channel tunnelling mechanism. Indeed, the 0 V is necessary to prevent adjacent bitlines from leading the selected one to a high potential, exploiting the coupling between bitlines. If this should happen, the effectiveness of the program operation will decrease, making other program pulses necessary. The drain selector is biased to the supply voltage. This means that the unselected bitlines will be precharged to VDD-$V_{TH}$, where $V_{TH}$ is the threshold voltage of the drain selector. The gate of the drain selector should

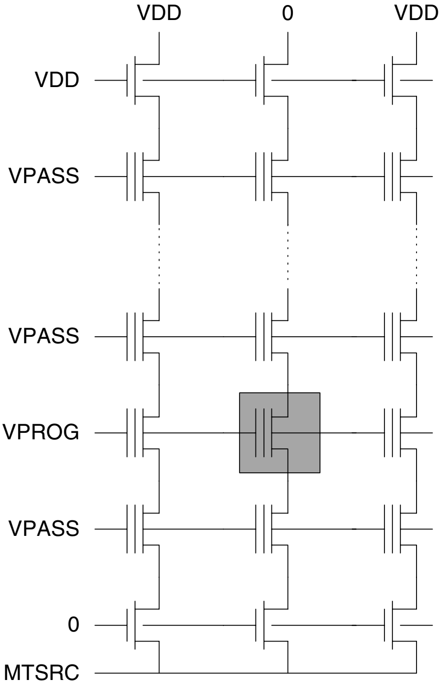


**Fig. 2.12** Biasing of a string of cells (NAND matrix) during the program operation; the cell under program is the highlighted one

be biased to a high voltage, so that the channel will be precharged exactly to VDD. Really, a compromise between the maximum voltage obtainable on the channel, time, and consumption must be reached. The gate of the unselected cells is biased at a voltage between 8 V and 10 V. The boost on the channel is directly proportional to this value: the greater it is, the greater and the longer the boost can be kept. However, the choice of this voltage is a critical point: too high a value increases the probability that cells sharing the same string of the selected cell will suffer a program operation (Vpass Stress); too low a value could not guarantee that the boost will last for the entire program operation, programming the cells placed on the same row as the selected one (Vprogram stress).

The source selector is off because its gate is biased at 0 V, while the matrix source is biased at VDD. In this way the source-channel junction is backward biased to prevent the leakage current from discharging the overboosted channel of unselected wordlines.

A further method that can be used to improve the boost of the channel consists in keeping at 0 V the wordlines adjacent to the selected one. For example, if the WL<7> is biased to the program voltage, than the WL<6> and WL<8> are kept to 0 V while all the other wordlines are biased at 8–10 V. In this way the channel of the unselected cells can reach higher voltages because its isolation has been improved. This technique is known as *local self boosting*.

However, as in a NOR memory, a specific sequence must be followed for a NAND memory also.

First of all, the bitlines are precharged and the gate of the drain selector in the string is biased at VDD. When this phase is finished, all the wordlines are biased at 8–10 V and the selected wordline is biased from 8–10 V to the final voltage. Two different ways may be adopted to do this last transition. In one case, the gate of the selected cell is directly biased at the program voltage, and the time needed for the charging of the wordline is proportional to its *RC*. In the other case, the final program voltage is reached by controlling the ramp (for example, stating that in 1 μs the voltage on the selected wordline can increase 1 V). During this phase, if the first method to bias the wordlines is chosen, the gate of the drain selector is biased to a lower voltage. Otherwise, the gate of the drain selector is left to the initial voltage. The scaling of technology, in fact, raises some problems, such as the reduction of the distance between wordlines and between the wordline and the source and drain selector's gate, which in turns implies a grater capacitance. When the wordline to be programmed is the one nearest the drain selector and it is biased directly to the final voltage, the gate of the drain selector may grow owing to the parasitic capacitance. Since the unselected bitlines are biased at VDD, if the gate of the drain selector increases over the VDD, the boost may be lost and cells on that wordline may suffer an unwanted program. This unwanted behaviour does not happen if the gate of the drain selector is lowered to a safe voltage (e.g., 1.8 V) before the raising up of the wordline. Otherwise, the slope of the wordline may be controlled by trying to choose a slope that avoids the boost of the capacitance between wordline and drain selector.

The same problem arises as soon as the cell near the source selector has to be programmed. However, in this case, the gate of the selector is biased to 0 V, while the source (which is the matrix source) is biased at VDD; so that to switch on the source selector, its gate should reach the voltage $\mathrm{VDD} + \mathrm{V}_{\mathrm{TH}}$ (its threshold voltage).

It is now important to underline that the program operation in NAND memories should follow a precise and well defined "hierarchy"—it is necessary to start from the cell nearest to the source selector and proceed along the string up to the cell nearest to the drain selector. This procedure is important, because the threshold voltage of a cell depends on the state of the cells placed between the considered cell and the source contact (the *background pattern dependency* phenomenon); the series resistance of the cells is different if they are programmed or erased. If this procedure is not followed, the threshold voltage of the cell may be different in the read phase, with respect to the verify phase.

## 2.4 Erase Operation in Flash Memories

Fowler-Nordheim tunneling [4, 5] is the physical phenomenon used to perform electrical erase in both NOR and NAND Flash memories.

In order to trigger the Fowler-Nordheim tunneling, a high voltage across the tunnel oxide must be applied. In first-generation NOR memories this is accomplished by biasing the source terminal with a high voltage (18 V is a typical value), the control gate with 0 V, and the drain terminal floating; the body terminal is biased to ground voltage because it is shorted with the core bulk. With such biasing, the Fowler-Nordheim tunneling arises between gate and source and extracts electrons from the floating gate; in this way, the memory cell voltage threshold becomes more negative. In any case, this biasing condition pushes the working point to the source/body junction voltage breakdown (practically speaking, when the erase starts the memory cell is in breakdown condition).

To avoid the junction breakdown, in second-generation memories the high voltage required is divided between gate and source terminals; if it is possible to bias the gate terminal with a negative voltage, the tunneling phenomenon may start as well, even if the source terminal is biased with a lower positive voltage level. This, in turn, poses the issue of generating negative voltages on-chip.

Regardless of both the previous biasing schemes, the band-to-band tunneling phenomenon increases the leakage current between the source/bulk junction; the leakage current represents a nonnegligible steady-state current consumption associated with a physical phenomenon—i.e., the tunneling—that is theoretically not current consuming. Last but not least, the hole's injection into the floating gate that is associated with the band-to-band tunneling reduces the Flash cell reliability.

In last-generation memories, this problem is solved by placing the memory matrix in a triple-well (Fig. 2.13); in this way the Flash cells bulk, i.e., the insulated p-well (iP-well), may be biased with a high voltage. This allows electron extraction

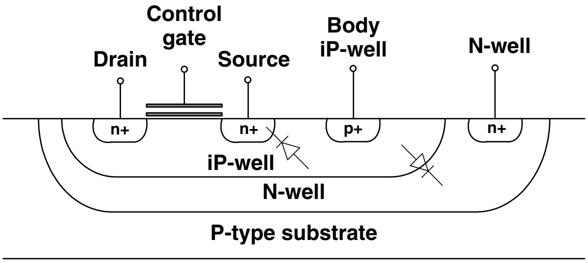


**Fig. 2.13** Triple-well matrix

all along the channel without the parasitic contribution of the source junction; the current consumption is reduced about three orders of magnitude [6].

As shown in Table 2.1, the sector erase time in the NOR architecture is about three orders of magnitude greater than the block erase time in the NAND architecture. Even if the physical phenomenon is the same in both cases, the architectural differences due to the different technical specifications to be met have a great impact on the complexity of the erase algorithm itself. The reason for these differences resides in the accuracy of the erased distribution positioning, as it will be shown in the following.

Figure 2.14 resumes the bias necessary for Fowler-Nordheim tunneling in the case of a standard matrix (a and b cases) and a triple-well one (c and d cases).

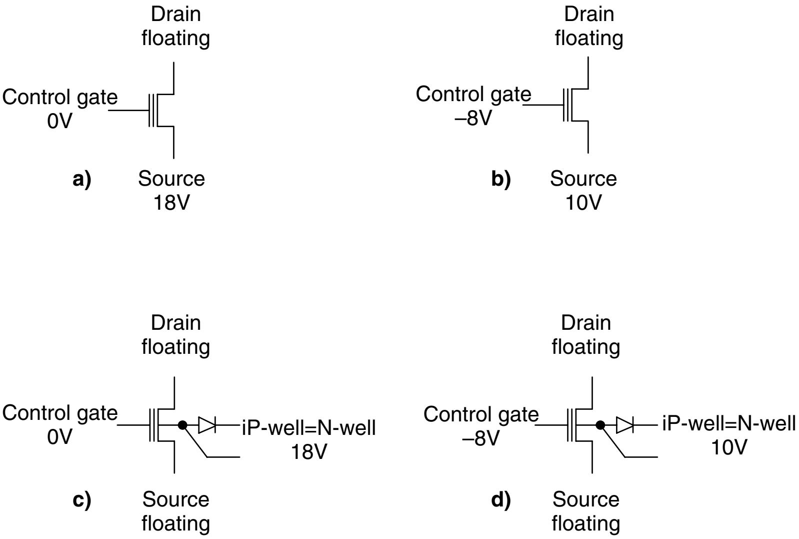


**Fig. 2.14** Biasing examples to trigger the Fowler-Nordheim tunneling in standard (**a** and **b**) and triple-well Flash memory cells (**c** and **d**)

### 2.4.1 Erase in NOR Architecture

In present generation NOR Flash memories, each memory sector has its own triple well with dedicated high-voltage switches for selective biasing of the addressed "sector's terminals" (source, body, and N-well). The electrical erase algorithm is much more complex than the simple biasing, with a voltage high enough to turn the tunneling on; it must manage the voltage values applied both to the addressed cells and to the unselected cells, as well as their transients.

Of course, the leading edge of the erased distribution (the least erased cells) must be low enough to assure an adequate distance between the programmed and the erased levels; in any case, the leading edge cannot be placed too low; the reason will be clear in the following.

At the end of the electrical erase algorithm, the trailing edge of the threshold voltage distribution of the erased cells (the most erased cells) may extend into the negative half-plane. In NOR architecture it is not possible to accept this kind of erased distribution.

The trailing edge must be well controlled as well. Referring to the NOR architecture, many cells are directly connected to the same bitline. To avoid any spurious current consumption from the unaddressed cells, their wordlines are kept to ground. In any case, this biasing keeps the memory cells off only if their voltage threshold is greater than zero; if this condition doesn't hold, the unaddressed cells are not fully off and their sub-threshold current flows through the bitline. The sub-threshold current of a single cell is small, but if we consider that many cells are connected to the same bitline (from 128 up to 512), the total sub-threshold current reduces the read margins (Fig. 2.15). To keep those cells off, a negative gate voltage should be applied during read; this solution is not "economically" feasible, because it implies greater circuit and algorithm complexity, as well as access time increase. To keep the read access time that is typical of NOR architecture, the leakage current problem must be removed at the point where it is generated, i.e., the electrical erase phase, by means of a specific erase algorithm.

To avoid an excessively high erase pulse shifting all of the erased distribution (or even only a portion) in the negative half-plane, i.e., to avoid the erased distribution becoming depleted, an electrical erase is performed by an increasing staircase voltage applied to the body terminal with a predetermined voltage step; between two steps, an erase verify is performed. This mechanism is effective because the threshold voltage step $\varDelta V_{TH}$ of a cell after each erase step applied to its body terminal is equal to the erase step itself, $\varDelta V_{ES}$:

$$|\Delta V_{TH}| = \Delta V_{ES} \tag{2.6}$$

The erase verify consists in checking that the cell current is high enough to distinguish between erased and programmed cells. A reference cell (called erase verify, EV) is used for the current comparison; the erase pulse and verify sequence goes on until all the cells of the sector to be erased show a current greater than EV (or at least equal to it). Since the erase pulse is applied globally to the whole sector,

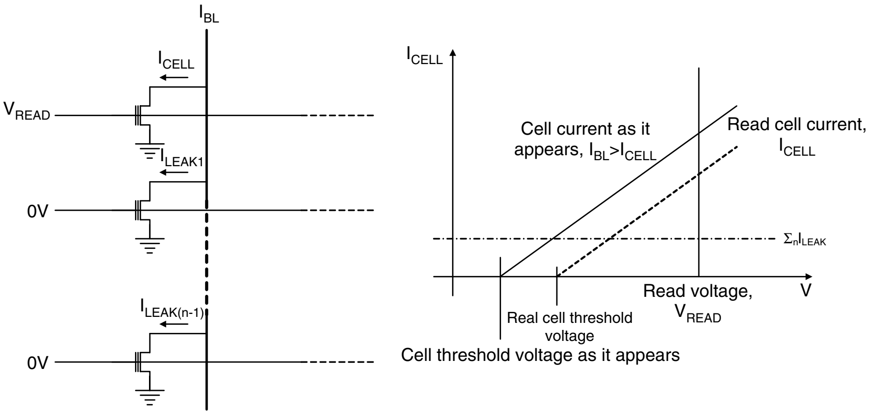

$I_{BL}$
$I_{CELL}$
$V_{READ}$
$I_{LEAK1}$
0V
$I_{LEAK(n-1)}$
0V
$I_{CELL}$
Cell current as it appears, $I_{BL}>I_{CELL}$
Read cell current, $I_{CELL}$
$\Sigma_n I_{LEAK}$
Read voltage, $V_{READ}$
V
Real cell threshold voltage
Cell threshold voltage as it appears

**Fig. 2.15** Effect of the sub-threshold bitline leakage current

and since the erase speed is different for each cell, when the slowest cells reach EV level, the fastest cells will be highly depleted. To recover the cells in such condition, during the second phase of the algorithm, named "soft-program and depletion verify," the depleted cells are programmed so as to have only enough threshold voltage to reduce to zero their leakage current. The soft-program applies program pulses (by CHE) until the threshold voltages of the cells of the sector are higher than that of the Depletion Verify (DV) reference cell. If it is possible to generate negative voltages on-chip, the unselected wordlines may be biased with negative voltage, so as to reduce to zero the leakage current contribution during the verify operation.

Let us reconsider the threshold level of the erase verify reference cell, EV. One could think that having this level as low as possible would result in a benefit in terms of read margin, but the lower the EV, the more the sector must be erased, and the more depleted cells have to be replaced over DV. This situation worsens both the soft-programming time (we will see that this time is the major part of the erase algorithm) and the current consumption in the first soft-programming steps, since the program overdrive during the first steps could be very high.

When the user sets the sector erase command, the cell status inside the sector is not defined; few programmed cells and many erased ones is a possible configuration. When such a sector is cycled (i.e., it is subject to many program and subsequent erase operations), the already erased cells are stressed; the erased distribution gets larger and with more leakage current. To avoid this degradation, a nonselective program pulse is applied to the whole sector, before the erase pulse. This phase is named "preconditioning" or "program all 0"; it allows the whole sector to be more or less in a uniform threshold state before the erase pulse, so as to avoid the erased distribution widening. Figure 2.16 shows the NOR architecture erase algorithm phases, as well as their effect on the threshold distribution.

As we have seen, the erase algorithm in NOR architecture Flash memories is very complex [7, 8, 9]; it is composed of many phases that must be connected correctly, especially because it deals with large high-voltage biased capacitive loads.

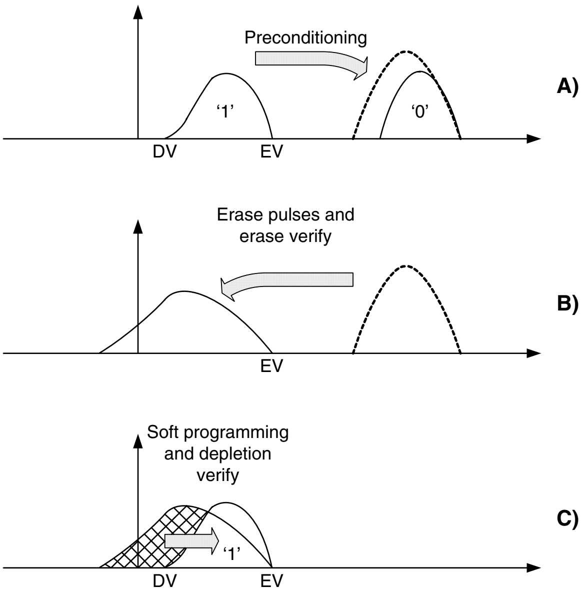

Preconditioning
'1'
'0'
A)
DV
EV
Erase pulses and erase verify
B)
EV
Soft programming and depletion verify
'1'
C)
DV
EV

**Fig. 2.16** Erase algorithm phases for NOR architecture Flash memory and their effect on the distribution

For example, to perform the erase verify after the erase pulse, the IP-well and the source of the sector under erase must be discharged from the high-voltage erase pulse to ground level, while the corresponding N-Well must be discharged from the high voltage to VDD. Great care must be taken in discharging these nodes. During each erase pulse the gates of the cells are biased at about −8 V, and the IP-well is biased at about 10 V; since source and drain terminals are floating, they are charged to a voltage that depends on the IP-well voltage and on their capacitive loads.

If the gate is abruptly discharged to GND, the gate-drain parasitic capacitance boosts the (floating) drain node to voltage (whose value depends on the gate and drain nodes capacitance), pushing the transistors towards breakdown. To avoid this problem, when switching from erase pulse to erase verify, the IP-well is discharged to GND first; then the drains are discharged to GND through the column decoder; then the source and N-well are biased respectively to GND and VDD. Finally, the gate is discharged to GND and subsequently biased with the erase verify voltage. All of these operations involve nodes with high parasitic capacitance and must not be too fast, so as to avoid snap back in discharge transistors and ground bounce that could cause spurious circuit switching. This sequence must be applied each time we need to switch from erase pulse to erase verify and vice versa; the algorithm and circuit complexity is evident.

Let us take into account an erase algorithm in a NOR architecture Flash memory with 1 Mbit sectors and 128-bit programming parallelism, with 1 s typical sector erase time. The preconditioning occupies a negligible time, in the order of some hundred μs (only a few programming pulses of programming, without verification); 250 ms are dedicated to erase pulse and verify sequence, and approximately 750 ms to the soft-programming and depletion verify. None of these phases may be skipped;

the voltages and their timings must be chosen with care, because no erase failure is admitted either at the beginning or during the device lifetime (100 k program/erase cycles allowed, 10 years lifetime).

### 2.4.2 Erase in NAND Architecture

In NAND architecture Flash memory the erase is performed by biasing the IP-well with a high voltage and keeping to GND the wordlines of the sector to be erased. Why? To generate negative voltages, negative boosters and high-voltage triple-well transistors are needed [10]; these transistors would be needed also to bias the row decoder with a negative voltage, and the row decoder itself should have at least one of these transistors to transfer the negative voltage to the wordlines. On the other hand, to avoid the use of negative voltages means saving from the point of view of the masks and lithographic complexity of technology. Also, in NOR architecture it is not trivial to place a row decoder able to drive negative voltages in the wordline Y-pitch, because this circuitry is composed at least of three transistors [11]; since in NAND architecture the Y-pitch is further reduced, it is more convenient to design a single N-channel transistor row decoding to pass positive (or zero) voltage values. Furthermore, as we will explain later, it is also possible to get the information we need on the position of the erased distribution with sufficient precision using positive voltages only: even from this point of view, it is not necessary to push towards the use of negative voltages.

In NAND architecture, the erase pulse is applied to the bulk terminal, which must be brought to a voltage higher than in the NOR case (the common source node is left floating). This terminal is shared between all the blocks to get a more compact matrix and to reduce the number of the structures to bias the iP-well (in NOR architecture there is one of these structures for each sector or group of sectors). By contrast, the parasitic capacitance to load is much higher, and the sectors not to be erased (i.e., all except one) should be managed properly to avoid spurious erase.

The advantage of NAND architecture from the erase point of view is that it is possible to locate the erased distribution into the negative threshold half-plane, i.e., the erased distribution may be depleted. Unlike NOR architecture, the cell threshold may be brought to the negative because the cells must act as pass-transistors, i.e., biased in conduction, for the reading mechanism to work. The subthreshold current does not represent a leakage contribution, because the selection transistors of the unselected bitlines prevent them from injecting any spurious current; for this reason, it is not necessary to bias the unselected wordlines with negative voltages to switch them off. The erased distribution width is huge, but without great effect on the series resistance of the stack when it has to be read, because during read algorithm all the cells of the stack except the addressed one are biased with a sufficiently high gate voltage ($V_{PASS}$, about 4.5 V).

In NAND architecture a great precision in placing the erased distribution is not necessary; all that is needed is an adequate margin with respect to the read condition.

For this reason, the erase is performed with a single impulse, calibrated to bring all the erased distribution to the negative, followed by a verification phase. The NAND specification helps in this case, allowing the management of bad blocks; if the block after an erase pulse does not meet with the erase verify, the user must consider it failed and store this information so as to prevent the use of this block. (If the malfunction occurs during the factory test, the sector can be directly marked as bad. The user is required to check all the blocks to avoid using the bad ones.)

The way the erase verify is performed in NAND architecture is substantially different from that used in NOR architecture. It is not possible to know exactly where the edges of the erased distribution are, since negative voltages are not provided on-chip. On the other hand, what is important is that all the cells of the string after erase are read as erased, i.e., they must be read as “1” when their gate is biased to GND and the other cells of the string are biased as pass-transistors. This condition will apply to all the cells of the string. The erase verify in NAND architecture is therefore a “stack operation,” because it requires that the whole stack is read as “1” when it is biased with GND (i.e., when biasing all the wordlines of the stack with such voltage). As already said, the exact placement of the erased distribution is unknown, but much time is saved for the erase verify, since it is possible to have the required information with only one read operation, instead of as many reads as the cells of the stack.

A further requirement is that the erased distribution have enough margin to contain the degradation due to subsequent erase and program cycles. Figure 2.17 is a qualitative representation of the “cycling window” of a NAND cell as a function of the number of program/erase cycles (program and erase are executed in a “blind” way, i.e., applying the program and erase pulses without verification). The thresholds of both the erased and the programmed cells increase with the number of cycles. This phenomenon is due to gain degradation and charge trapped into the oxide. The macroscopic effect on the erase operation is such that the erase pulse that allows having pass result at erase verify at the beginning of the operating life could no longer be enough as the cycle number increases.

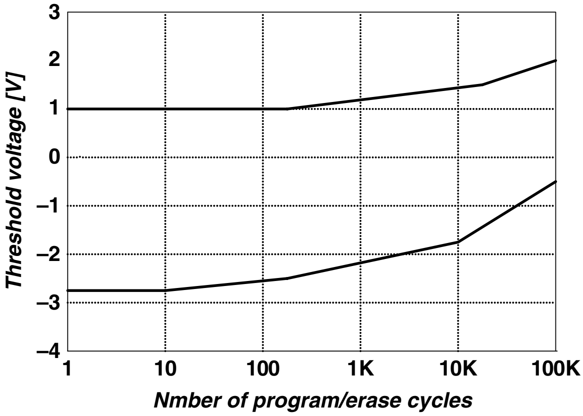


**Fig. 2.17** Cycling window in NAND Flash memories; since the erased level shifts towards positive voltages, a proper threshold margin at the beginning of the operating life is required

The cycling degradation also affects the NOR architecture, but in this case it is possible to apply further erase pulses if they are set at the beginning of operating life. In other words, in this case the cycling degradation results in an erase time increase (both for the erase time itself and for the soft-programming phase), but the NOR specifications allow this. In NAND architecture instead, the specifics leave no room for a further erase pulse if the first is not effective; for this reason, the erase pulse level and width must be carefully calibrated after accurate analyses at the process level.

Also in NAND architecture, a preconditioning before the erase pulse may be useful to get more uniform threshold voltages so as to reduce the distribution spread. Thanks to the program mechanism, which uses the tunneling phenomenon, a lot of time is saved if the preconditioning is executed at a time with a unique pulse on the wordlines of the block to be erased.

The erase sequence in NAND architecture is shown in Fig. 2.18, together with the effect that the various phases have on the distribution. The preconditioning is not displayed because it is often not used (in any case, its effect is similar to that in NOR architecture).

The nodes involved in the erase operation must be discharged carefully in NAND architecture also. Since the bitlines are floating, they are charged to a voltage potential, depending on the bulk voltage and the capacitive loads. The bulk discharge must be well controlled, as in NOR architecture. The common source node may be initially left floating (it starts discharging anyway, due to its capacitive coupling to the bulk), and then connected to ground when the bulk discharge is over. The same concept applies also to the bitlines: after the initial discharge due to their coupling with the bulk, they are discharged to ground, thanks to proper transistors.

The selected wordlines are already kept to ground, but the unselected are floating, so they are discharged to ground by activating all the row decoders. Great attention must be paid at the technological and manufacturing levels to the electrical characteristics of the row decoder. The leakage should be as low as possible. In fact, during the erase pulse the wordlines of the unselected blocks are left floating, so they are free to load to a certain voltage by their capacitive coupling with the bulk, and tunneling is not allowed. However, if there is a leakage at the row decoder level, the unselected wordlines may discharge; if this happens, the voltage drop across the tunnel oxide may be enough to trigger a spurious erase of the unselected blocks.

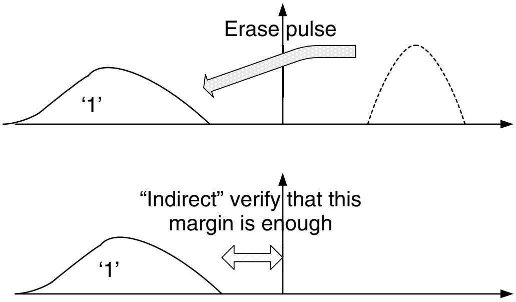


**Fig. 2.18** Erase algorithm phases for NAND architecture Flash memory and their effect on the distribution

In summary, in a NAND Flash the typical block erase lasts approximately 1 ms; 800 µs for the erase pulse and about 100 µs for the erase verify. Usually, the preconditioning is not carried out, but if it is carried out it lasts no more than 100 µs. The reader should note that in the NAND architecture the erase time of a block is really independent of the size of the sector, because none of the operations that compose the algorithm is made at the page level, but everything is done at the block level. Furthermore, the erase time is "rigid," as all the phases last exactly the same time for all the blocks, since "repetitions" are not possible to recover the failed blocks.

## References

1. Campardo G, Micheloni R, Novosel D (2005) VLSI-design of nonvolatile memories. Springer series in advanced microelectronics
2. Elmhurst D et al. (2003) A 1.8V 128Mb 125MHz multi-level cell flash memory with flexible read while write. ISSCC Dig Tech Pap 286–287
3. Jung TS (1996) A 3.3-V 128-Mb multilevel NAND flash memory for mass storage applications. ISSCC Dig Tech Pap 32–33
4. Lenzlinger M, Show EH (1969) Fowler-Nordheim tunnelling into thermally grown $SiO_2$. IEDM Tech Dig 40:273–283
5. Hu C (1993) Future CMOS scaling and reliability. P IEEE 81:682–689
6. Kenney S et al. (1992) Complete transient simulation of flash EEPROM devices. IEEE T Electron Dev 39:2750–2757
7. Bez R et al. (2003) Introduction to flash memory. P IEEE 91:554–568
8. Cappelletti P et al. (eds) (1999) Flash memories. Kluwer, Norwell, MA
9. Pavan P, Bez R, Olivo P, Zanoni E (1997) Flash memory cells—an overview. P IEEE 85: 1248–1271
10. Umezawa A et al. (1992) A 5 V-only operation 0.6-µm flash EEPROM with row decoder scheme in triple-well structure. IEEE J Solid-St Circ 27:1540–1546
11. Motta I, Ragone G, Khouri O, Torelli G, Micheloni R (2003) High-voltage management in single-supply CHE NOR-type flash memories. P IEEE 91:554–568