Fundamentals of solid-state memory technologies in consumer electronics - Part 1: Flash memory

Thomas Coughlin - August 21, 2012


4.1 Development and History of Flash Memory
Flash is an extension of the floating gate method of manufacturing nonvolatile memory. The first sort of floating gate memory was the Erasable Programmable Read Only Memory (EPROM), invented in the 1960s but not developed until the 1970s. In EPROM, like in Dynamic Random Access Memory (DRAM) and Read Only Memory (ROM), each memory bit was represented by a transistor.

Appendix C gives a list of some of the most significant flash memory manufacturers.

It helps to understand how transistors work if you want to understand these circuits. In a Field-Effect Transistor (FET), the FET current flows from the source to the drain. The gate controls how much current flows through the channel (the area between the source and drain). If the gate is unbiased, then the current flows through the channel relatively freely. If a bias is applied to the gate, then the channel depletes, that is, the carriers are moved out of part of the channel, making it seem narrower, and limiting the current flow.

This principle is key to all FET-based technologies, Metal-Oxide Semiconductor (MOS), Complementary Metal-Oxide Semiconductor (CMOS), Bipolar Complementary Metal-Oxide Semiconductor (BiCMOS), and so on. The ability to turn a circuit’s current flow on and off allows
individual bits to be routed to a data bus.

In a DRAM each memory bit’s transistor uses a capacitor to store the bit. The case is different with a ROM, where each bit’s transistor uses a short or open circuit to represent a bit (either programmed at manufacture by a mask, or afterward by a fuse that could be blown).

In EPROM, the transistor itself looks as if it contains something kind of like a capacitor, but it is actually a second gate complementing the control gate of each bit’s transistor. The bit’s transistor actually has two gates, one that is connected to the bit line, and one that was connected to nothing - it "floats." The technology is known as "floating gate." This concept is used by four technologies: EPROM, Electrically Erasable Programmable Read Only Memory (EEPROM), NOR, and NAND.

Figure 4.1 illustrates the cross section of a floating gate. When a bit is programmed, electrons are stored upon the floating gate. This has the effect of offsetting the charge on the control gate of the transistor. If there is no charge upon the floating gate, then the control gate’s charge determines whether or not a current flows through the channel. A strong charge on the control gate ensures that no current flows. A weak charge will allow a strong current to flow through.

The floating gate is capable of storing a charge of its own. This adds to the bias of the control gate. If there is a charge on the floating gate, then no current will flow through the channel, whether or not there is a charge on the control gate.

**Erasing, Writing, and Reading Flash Memory**

4.2 Erasing, Writing, and Reading Flash Memory

The key to using floating gate technology is to put a charge onto the floating gate when it is needed, and to take it off when it is no longer needed. There are two methods of getting charges onto and off of the floating gate: Channel Hot Electron injection (CHE) and Fowler-Nordheim tunneling (FN). Both concepts involve quantum methods which are beyond the scope of this text, so they will be dealt with here in a very simplistic way.

Channel hot electron injection takes advantage of the fact that electrons are more mobile with high current levels. When CHE is used to add electrons to a floating gate, a high source to drain current is run through the channel. With this high current, a number of electrons are looking for ways to
"boil" out of the channel. A positive bias on the control gate attracts electrons from the channel into the floating gate, where they become trapped. This adds a bias onto the floating gate (see Figure 4.2).

**Figure 4.2: Channel Hot Electron Injection.**

Fowler-Nordheim tunneling requires a high voltage (compared to the operating voltage of the chip - usually between 7-12 V) to be placed between the source and the control gate of the transistor. If the voltage is sufficient, the electrons "tunnel" through the gate oxide layer and come to rest upon the floating gate. This is illustrated in Figure 4.3.

**Figure 4.3: Fowler-Nordheim Tunneling.**

In EPROM, the original floating gate technology, the memory was erased through exposure to ultraviolet light. Light rays would allow the electrons trapped in the floating gate to migrate back to the channel. In EEPROM, Fowler-Nordheim tunneling was reversed by a second transistor on the memory bit to take electrons back out of the floating gate and put them back into the source where they originally came from. This erasure approach is illustrated in Figure 4.4.
Both NAND and NOR flash technology use the same electronic erasure concept as EEPROM, but rather than add an extra erase transition to each individual bit’s transistor, a single very large transistor erases all the transistors in a sub-array called a sector. This effectively halves the size of the memory array, providing significant cost savings to designers who do not need to be able to individually erase each bit or byte of the memory.

**Difficulties that Cause Wear in Flash Memory**

**4.3 Difficulties that Cause Wear in Flash Memory**

Tunneling electrons migrating through the tunnel oxide sometimes cause difficulties. It is inevitable that electrons from time to time will get trapped in the tunnel oxide. These electrons, once trapped, cannot be removed. This will impact the operation of the cell to a certain degree, depending upon the number of electrons trapped in the oxide: if a lot of electrons are trapped then there will be a big impact, but a low number of electrons are not likely to cause much impact at all.

As the number of trapped electrons in the oxide gets too large, it becomes harder to maintain electrons in the floating gate. Thus the memory cell leaks the stored charge.

The number of electrons trapped in any one cell’s tunnel oxide is a function of how the chip is made, and more importantly, of how many times that particular transistor has been erased and rewritten. A specification has thus been devised to recommend a maximum number of erase/write cycles that a memory cell on a certain chip can withstand before a failure is likely to occur, and this specification is called the chip’s *endurance*.

Typical endurances have been managed up to very high levels over the course of time. For many chips today, endurance is specified as 10^5 erase/write cycles. As we will later see, some novel approaches are used to further remove the possibility that endurance failures will impact the operation of a system.

**4.4 Common Flash Memory Storage Technologies: NOR and NAND**

There are two kinds of flash memory, *NOR* and *NAND*. The two terms are names of types of logic gates, the negated OR function and the negated AND function. The big difference between the two types of architectures is real estate. NAND has a significantly smaller die size than does NOR. This translates to significant cost savings.
These cost savings come with a trade-off. NAND does not behave like other memories. While NOR, SRAM, and DRAM are random access devices (the "RAM" part of DRAM and SRAM stands for random access memory) NAND is part random and part serial. Once an address is given to the device, there is a long pause, then that address and several adjacent addresses’ data come out in a burst like a machine gun. We will explain why in the next two sections.

Another way that NAND makers squeeze out costs is by taking some shortcuts in ensuring data integrity. NAND was designed to replace hard disk drives back in the 1980s. At that time some thought that there was a brick wall limiting the capacity of magnetic storage. Once hard disk drives hit that brick wall then semiconductor memory could race past it in density and cost.

NAND’s inventors borrowed a leaf from the hard disk drive designers’ book by allowing their design to have occasional errors that would be corrected by external logic. Both hard disk drive designers and NAND designers were able to use this more relaxed approach to data integrity in order to squeeze more bits onto a given piece of real estate (whether on a magnetic disc or on a silicon chip) than would be possible if absolute integrity were to be maintained.

Two techniques, serial access and lower data integrity, allow NAND die sizes to be less than half the size of their NOR counterparts. Let’s look at the two of these in depth.

**How Does NOR Memory Work?**

4.4.1 How Does NOR Memory Work?
We will use a classroom analogy. Picture a class where there is space around all the desks. Each of the desks will represent a bit transistor in the memory, with the students representing data. The teacher represents the system interface that handles the data transaction between the rest of the system and the memory bits.

We have labeled the students in Figure 4.5 as A1, B1, and so on, to indicate the row and column where that student sits. In this example there are 14 columns (A–N) and five rows (1–5), giving 70 students all together.
Most memories are designed similarly to this classroom. There are rows and columns, and word
lines to tell the memory bits which row is being requested and bit lines to access a column within
that row.

When a memory bit is requested, it is similar to the teacher calling a student to the front of the
room. Say the teacher in this example called for student B3 to come to the front of the room (Figure
4.6). That student would simply walk from his or her desk up the aisle to go to the teacher’s desk. All
students could get to the teacher’s desk in about the same amount of time, and it would not take
very long to get there.
This is very similar to the way that a NOR memory works. All data can be accessed rapidly, and all accesses take about the same amount of time. Data can be requested from individual locations in a completely random sequence.

**How Does NAND Memory Work?**

4.4.2 How Does NAND Memory Work?
Now let’s say that some cost-cutting measures were implemented by this school. The administration found that they could sell one of their buildings if they used their existing buildings more efficiently. Their plan is to shrink the size of each classroom, which they can do if they push all the desks up against each other, with the desks on one side of the classroom pushed right against the wall. This is illustrated in Figure 4.7. It is clear that there is significant space savings by taking this approach.
This approach does not come for free, though. Now, when the teacher calls student B3 to the front of the room, all the students in B3’s row must get up and walk to the front of the room in a line (Figure 4.8). There is simply not enough space for them to do it any other way. Once the row is in front, then B3, or C3, or N3, or any of these students can get to the teacher rapidly. On the other hand, none of these students can reach the front of the room until all the students blocking the way have gone to the front.
Figure 4.8: In the NAND Classroom All the Students in a Row Must Move for One of Their Row-Mates to Get to the Teacher.

This is similar to the way that NAND is laid out. Just as the students share a single aisle at the far side of the classroom, many bits in NAND share a bit line, dramatically reducing the amount of space used on the NAND die to move data back and forth to the bit transistors. Like the classroom analogy, NAND takes a longer time to get a memory bit (a student) that is randomly called to the rest of the system (the front of the class). Once that bit line’s data is ready (the student’s row is in the front) then data from that word line can be presented to the system at a rapid-fire rate (just as every student in that row can get to the teacher’s desk very quickly).

From this it should be clear that NAND is less expensive to manufacture than NOR, since the manufacturing cost of a silicon chip is a function of its size. Keep in mind, though, that the cost savings might be offset by the two technologies’ significant difference in functionality. Although NOR usually seems easier to use, surprisingly enough, this difference in functionality actually makes NAND a better fit than NOR in applications where serial accesses are preferred to random accesses. Video and audio streaming are two very good examples of applications where serial access is preferable to random.

©2012 Elsevier, Inc. All rights reserved. Printed with permission from Newnes Press, a division of Elsevier. Copyright 2012. For more information on this title and other similar books, please visit http://store.elsevier.com/Newnes/IMP_73/.

For more articles like this and others related to designing for consumer electronics, visit EDN’s Consumer Electronics Design Center and/or subscribe to the biweekly Consumer Electronics newsletter (free registration).