Flipping Bits and Spinning Disks

The Physics of Computer Memory

Matthew Adams

May 20, 2013

Abstract

In this paper, I explore the physics that makes some of the most important types of computer memory possible. First a basic computer science-level understanding of computer memory is established. Then I explain the physics background necessary to appreciate the design of various digital storage technologies. Finally, I examine four different memory technologies- SRAM, DRAM, magnetic disk drives, and flash memory- and describe how each is specifically suited to its role in modern computer systems.

1 Introduction

Pictures, music, applications- everything on your computer or phone is encoded as millions or billions of individual binary digits, called bits. Each bit is in one of two states: an "on"/"true" state, represented by a `1`, or an "off"/"false" state, which is represented by a `0`. This system makes storing digital information fairly straightforward; computers can simply use any two-state system to represent each bit.[1] Such a two-state system could be anything from a punch card with states of "hole" and "no hole" to a magnet with states of "pointing south-north" and "pointing north-south." The fundamental question that this paper tries to answer is: in modern computers, what physical systems serve as memory bits, and how do they work?

First, we need to understand just what "computer memory" means. Memory is not one single, specific part of a computer. Instead, data gets stored in several different places, depending on how it is being used. From a usage perspective, computer memory can essentially be grouped into five categories:

[1] In fact, there is no intrinsic reason why computers couldn't store data using a number system with a different base. However, binary numbers have proven particularly easy to work with since each bit is essentially an on or off switch. Using a different base would dramatically increase the physical complexity of computers without actually providing any computational benefit. However, if each bit could represent an infinite amount of states (i.e. a superposition of the states `1` and `0`), as is the case for qbits of quantum computers, there would be a dramatic performance gain.

Main Memory: Also called "primary memory," main memory is where computers store the programs they are currently running and the data that they are currently using. In this way, it acts sort of like a desk top; you may keep several books and papers out on your desk while you're working with them. Since main memory almost always consists of DRAM ("Dynamic Random Access Memory" - see section 3.2), main memory is often simply called RAM.

Secondary Memory: This is where all files, programs, the operating system, and other data gets stored permanently.[2] Secondary memory can be thought of as a filing cabinet: you keep your papers in the cabinet, and, when you want to use them, you take them out and put them on your desk, i.e. main memory. (Of course this analogy is not quite perfect, because when you take a file out of a filing cabinet, it is - obviously - no longer in the cabinet, but when you use a file from secondary memory in primary memory, the file actually remains in secondary memory as well.) Until the past few years, the most common secondary memory technology has been hard disk drives, so the term "secondary memory" is often conflated with "hard drive."

Processor Caches/Registers: Processors have memory incorporated into them that they use to hold data they are using and instructions that they are about to run. Information is stored here just briefly- only right as the processor is using it or about to use it. For example, if you are using a web browser, the whole web browser application is in main memory, while only specific snippets of programs and data are ever in the the CPU's registers and caches. (When you are not using the browser, the application is stored in secondary memory.) Implementations of this memory are typically very fast but relatively expensive per bit.

Removable Storage: As the name suggests, this category of digital storage encompasses memory that actually gets physically removed from the computer, such as CD's and USB memory sticks. While I will discuss some of the underlying technologies used in removable storage, I will focus on the first three types of memory.

Read-Only Memory (ROM): This kind of memory is only written once and typically contains very low-level software used to run the computer's hardware. It shares similar characteristics with some of the technologies used for secondary memory, and this paper will not delve deeply into the physics of ROM.

From a computer science perspective, it doesn't matter whether main memory is implemented by writing 1's and 0's on a piece of paper or by using the latest DRAM technology. It's up to physicists and engineers to create memory technologies that are fast, inexpensive, reliable, and specialized for each class of memory.

[2] In practice, sometimes there isn't enough space in main memory to do everything the computer is trying to do, and the operating system might use some of the secondary storage as if it were primary storage. This is called *virtual memory*. Virtual memory is a nice feature in that the computer can function even when it has run out of main memory space, but programs that are being run out of virtual memory are significantly slowed down, as you might expect.

In this paper, I will explore the physical systems of digital memory storage that are most prevalent in modern computers, including magnetic hard drives, flash memory, and two types of temporary memory based on electric circuit states.

2 Physics Background

There is one tiny, hugely important device that makes computers as we know them today possible: the transistor. The simple purpose of transistors belies both their importance and the fascinating physics that makes them work. As they are used in computers, transistors primarily function as voltage-controlled switches; they allow one circuit to turn another circuit on or off- without using moving parts or bulky, unreliable vacuum tubes. As we will see, this switching ability of transistors makes them the essential building block for circuit components that perform logic operations, which, in turn, are combined in various ways to make circuits that can calculate and circuits that can store data.

2.1 Semiconductors

In effect, transistors are just clever arrangements of semiconductors, so a good description of the physics behind transistors begins with a discussion about semiconductors. As the name implies, semiconductors are materials that have a conductivity between conductors and insulators.

The electrical conductivity and basic electric properties of a material can be understood through examining that material's energy *band structure*. Band structure refers to the specific energy levels that the electrons in a material are allowed (by quantum mechanics) to occupy. Just as the electrons of a single, isolated atom are restricted to specific, discrete energy levels, the electrons in a solid material must be in one of that material's energy bands. However, unlike an atomic energy level, which corresponds to a single energy value, an energy band consists of an effectively continuous range of allowed energies. For the purposes of analyzing conductivity, the most important energy bands are the highest occupied band, called the valence band, and the lowest unoccupied band, called the conduction band. All of the lower energy bands are filled with electrons that are bound too tightly to their atomic nuclei to move, under normal conditions.

The valence band is defined as the highest occupied energy band when the material's electrons have not been excited (ie. there is no external electromagnetic field and the temperature is at absolute zero). Electrons in the valence band are not free to move unless they are excited into the conduction band. As shown in Figure 1, for conductors, the valence and conduction bands actually overlap, which allows the valence electrons to move freely and generate a current in the presence of an applied electric field. By contrast, for insulators, the separation between the valence and conduction bands, called the *band gap*, is

large; therefore it takes a significant amount of energy to promote electrons from the valence band into the conduction band, where they can actually move. Under normal conditions, without that large energy input, insulators cannot conduct. As you can see in Figure 1, semiconductors are just insulators with a small band gap, so only a small energy input is needed to excite a semiconductor's electrons into the conduction band, thereby allowing it to conduct electricity. In fact, using statistical mechanics, one can show that, at room temperature, a sufficient amount of a typical semiconductor's electrons are excited into the conduction band to allow it to conduct, albeit poorly.

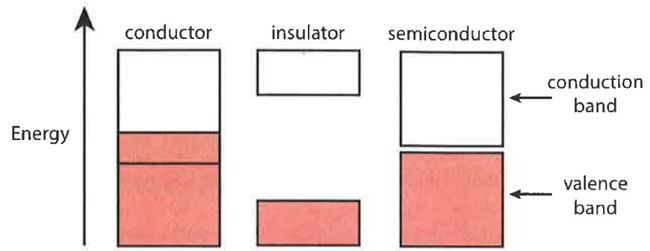


Figure 1: This simplified energy band diagram illustrates the difference between conductors, semiconductors, and insulators. Shaded areas represent energy levels occupied by electrons; the vertical axis indicates increasing energy levels, and the horizontal axis has no physical meaning. Conductors have no band gap, as their valence and conduction bands overlap, insulators have a large band gap of ~5 eV, and semiconductors have a small band gap of ~1 eV.

So far, the discussion of semiconductors has been limited to describing a pure semiconductor, such as a pure silicon or germanium crystal. However, the semiconductors that are most commonly used contain certain intentionally-added impurities, called *dopants*. Naturally, the choice of dopant depends on the desired electrical properties of the resulting doped semiconductor. If the atoms of the dopant have *more* valence electrons than then the atoms of the pure semiconductor they are being added to, then the resulting doped semiconductor material will actually have electrons in its conduction band at absolute zero. Although the overall doped material is still electrically neutral (because for each "extra" electron, there is a corresponding "extra" proton), the doped semiconductor has an excess of negative charge carriers and is therefore called an *n-type*, meaning "negative type," semiconductor. By contrast, as shown in Figure 2, *p-type*- or "positive type"- semiconductors have dopants with *fewer* valence electrons and thus have an unfilled valence band. There are states that electrons could occupy but don't. Just like bubbles in a glass of water, holes are an absence of a substance that can be effectively treated as a particle. When discussing transistors, it will be useful to conceptualize p-type semiconductors as having extra holes and n-type semiconductors having extra electrons.

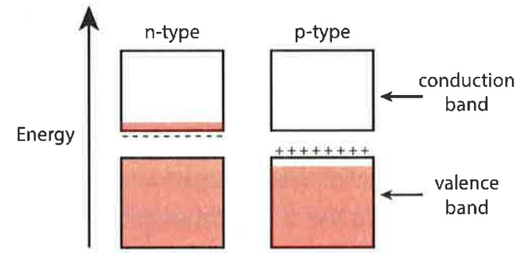


Figure 2: This picture shows the difference between n-type and p-type doped semiconductors.

2.2 Transistors

Now that we know some of the basics of semiconductors, we can finally begin to understand how transistors operate. Recall that transistors, as they are typically used in computers, act as voltage-controlled on/off switches. Although there are many different varieties of transistor, in this paper, "transistor" refers to a MOSFET (Metal Oxide Semiconductor Field Effect Transistor- the meaning of these terms will hopefully become clear), since MOSFETs are the type of transistor typically used in the circuits I'll discuss. As shown in Figure 3, a typical MOSFET consists of a metal *gate* plate connected to an insulating oxide layer (usually SiO_2), which is on a carefully doped semiconductor. In *npn* MOSFETs, like in Figure 3, the semiconductor has been doped so that it is p-type except for two n-type regions, one called the *source* and the other the *drain*. Critically, the boundaries between the n and p regions are aligned or slightly overlapping with the edges of the gate and oxide layers.

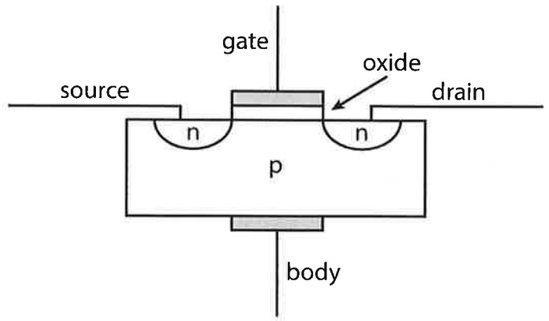


Figure 3: This diagram shows the components that make up an npn MOSFET: a metal gate, insulting oxide layer, and a doped semiconductor.

This begs the question: how does the control voltage turn the transistor's ability to conduct on and off? The answer lies in the electrical properties of doped semiconductors described earlier. Through examining charge carrier diffusion, one can show that, at thermal equilibrium, any non-uniformly doped semiconductor has an electric field given by

$$E = -V_t \frac{1}{n(x)} \frac{dn(x)}{dx} , \tag{1}$$

where x is position in the semiconductor, $n(x)$ is free electron concentration, and V_t is the thermal voltage, which, according to the Einstein relation $V_t = kT/q$, is constant for a fixed temperature [1]. Taking the

negative integral of E to get an expression for potential difference yields

$$\psi_{x_1} - \psi_{x_2} = V_t \ln \frac{n(x_1)}{n(x_2)} . \tag{2}$$

Choosing x_1 and x_2 to be on either side of a pn junction in a transistor, we can derive an expression for the potential barrier of that junction and begin to understand MOSFETs' electrical properties. [1]

On the n-type side, the electron concentration n_n at thermal equilibrium is effectively equal to N_D, the concentration of *donor* dopant atoms- atoms added to the pure semiconductor that donate extra electrons [1]. Similarly, the concentration of holes p_p on the p-type side is equal to N_A, the concentration of *acceptor* dopant atoms. For doped semiconductors in general, it is known that the concentrations of free electrons n and holes p are inversely proportional according to

$$np = n_i^2 , \tag{3}$$

where n_i, called the intrinsic carrier concentration, is constant for a given temperature [1]. Therefore, the concentration of electrons on the p side n_p equals n_i^2/N_A, which we can plug into equation 2 to obtain the voltage barrier of the pn junction:

$$V_b = V_t \ln \frac{N_A N_D}{n_i^2} . \tag{4}$$

The fact that the barrier potential V_b is nonzero means that there is is a potential difference across the depletion region absent an external applied voltage. For silicon, for a dopant concentration $N_A = N_D = 10^{16}$ cm^{-3} and thermal voltage $V_t = 0.0259$ (at room temperature), $n_i = 1.6 \times 10^{10}$ cm^{-3} and $V_b = 0.69$ V [1]. This $\sim .7$ V across a pn junction makes transistors function, because having that potential barrier means that it takes work to move electrons from n doped regions to p doped regions than vice versa.

This potential difference can be further understood by considering the qualitative behavior of the charge carriers at the semiconductor junction. When a p-type and an n-type semiconductor are put in contact with one another, some of the "extra" electrons from the n-type semiconductor combine with some of the extra holes in the p-type semiconductor, which leaves a small ($\sim 10\mu$m) area around the junction, called the *depletion zone*, where there are no charge carriers [2]. Note that the depletion region is small because only a small fraction of the charge carriers can ever actually meet. Before the electrons and holes joined in the depletion region, the volume that would become the depletion region was electrically neutral. This means that after the holes and electrons cancel each other out, we are left with a net *negative* charge on the p semiconductor side of the depletion region and a net *positive* charge on the n semiconductor side, as shown in Figure 4. The fact that the two sides of the depletion region have opposite charges produces

the voltage barrier described by equation 4.

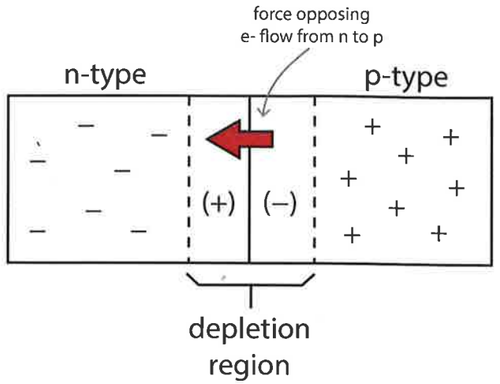
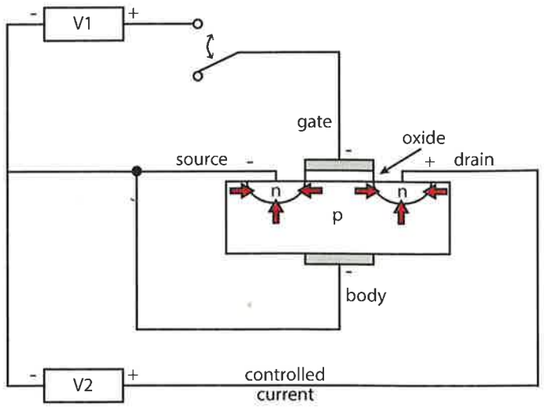


Figure 4: When a p-type and an n-type semiconductor are put in contact, the charge carriers diffuse and combine in the depletion region around the junction. This leads to the internal force shown.

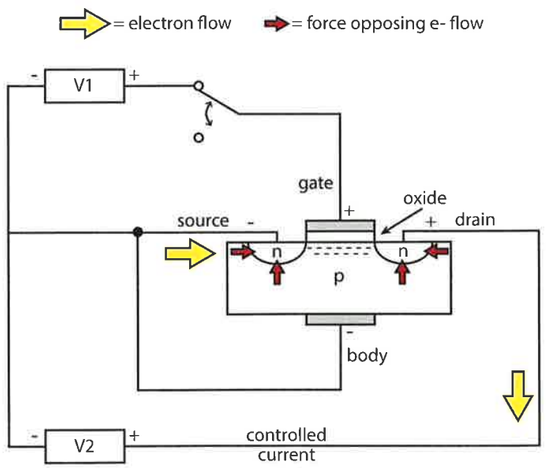
Keeping in mind the barrier voltage across the pn junctions, let's take another look the MOSFET structure shown in Figure 3. When there is no voltage applied to the transistor gate, the controlled current must pass through an np junction and a pn junction. To get current to flow through the np junction, you'll need to apply a voltage to overcome the np junction's internal electric field. However, that same voltage is across the pn junction as well, and, instead of opposing the internal electric field, the applied voltage actually enhances it. No matter how you apply an external voltage across an np and a pn junction in series, one of the junctions will be effectively very resistive, thereby making the whole transistor effectively non-conductive. Therefore, to make the transistor conduct, one of the materials in the $n \to p \to n$ path must change. This is where the other pieces of the transistor come into play: the metal gate, insulating oxide layer, and the body plate opposite them.

When the control voltage is applied across the gate and body terminals, an electric field is produced such that some electrons in the p-type semiconductor are attracted to the positive gate plate. The electrons in the p semiconductor get pulled towards the gate by the applied field, but they can't reach it because of the insulator blocking the way, so they basically pile up along the junction between the p semiconductor and the insulator. The material in that region of "piled-up" electrons is now effectively transformed into an n-type semiconductor, as it has an excess of electrons relative to the pure semiconductor. The region of the p-type semiconductor along the insulator that has been effectively transformed into an n-type semiconductor is called an *n-channel*, as it provides an $n \to n \to n$ channel through which current passes from the source to the drain. (For this reason, npn transistors are also called n-channel transistors.) This

changing semiconductor type in response to the field applied across the semiconductor is the "field effect" part of the MOSFET acronym. Figure 5 shows in detail how an n-channel semiconductor operates in the context of a simple circuit.



(a) When the control circuit is off, the forces (indicated with small red arrows) at the n and p junctions oppose the flow of electrons through the transistor. Even though there is a potential difference of $V2$ between the source and the drain, the controlled current does not flow.



(b) Turning the control voltage $V1$ on causes electrons in the p-type semiconductor to be attracted to the gate plate and pile up against the insulating oxide layer. This causes the small region of the p-type semiconductor near the oxide layer to effectively become an n-type semiconductor, which allows the controlled current driven by $V2$ to flow.

Figure 5: This is the archetypal MOSFET cross-section structure attached to a simple control circuit. Note that the transistor depicted is an npn *or* n-channel *transistor, but the structure is very similar for a* pnp/p-channel *MOSFET. Graphic adapted (with corrections) from [2].*

Now that we know how npn MOSFETs work, it's natural to wonder whether pnp-type MOSFETs work in the same manner. In practice, n-channel MOSFETs are "high on"/"low off" switches, whereas p-channel MOSFETs are "high off"/"low on," where "high off" means that the controlled current is off when the gate voltage is relatively high. To put it another way: given the same control input, when n-channel transistors are on, p-channel transistors are off, and vice versa. As we will see in the next section, this complementary nature of the two FET types lies at the heart of how digital electronics works.

2.3 Logic Gates

Every bit of "thinking" a computer does is really just algorithms run using many clever arrangements of logic gates. On some fundamental level, computers are strikingly simple. A logic gate is a circuit component that performs a Boolean logic operation, such as `AND`, `OR`, or `NOT`. Each gate has a corresponding symbol, shown in Figure 6, that is used to represent it in circuit diagrams. Boolean logic operations behave exactly as their names imply; `AND` gates output a `1` only if both of their inputs are `1`, `OR` gates output a `1` if at least one of their inputs is `1`, and `NOT` gates return a `0` given a `1` input and vice versa. The most common other gates are act like combinations of these three gates. For example `NAND` gates act like an `AND` gate followed by a `NOT` gate and `NOR` gates act like an `OR` gate followed by a `NOT` gate.

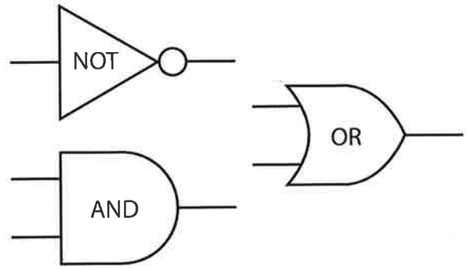
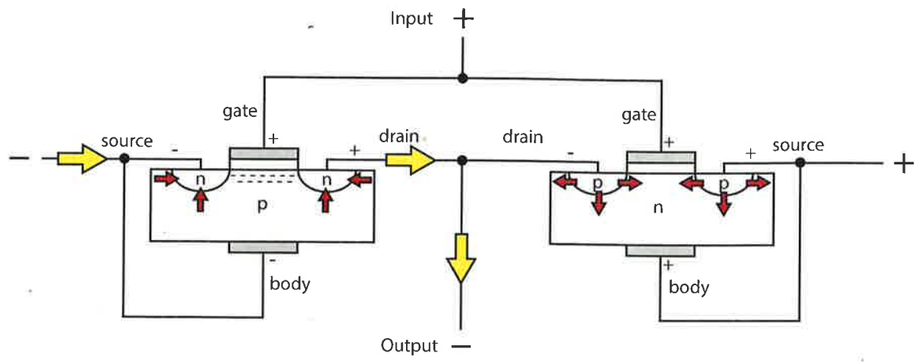


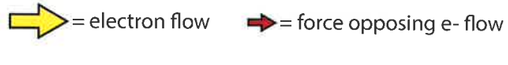
Figure 6: These are the symbols used to represent the various logic gates in circuit diagrams. (The labels in the middle of the pictures are not usually included, but I will include them to hopefully make my circuit diagrams easier to follow.)

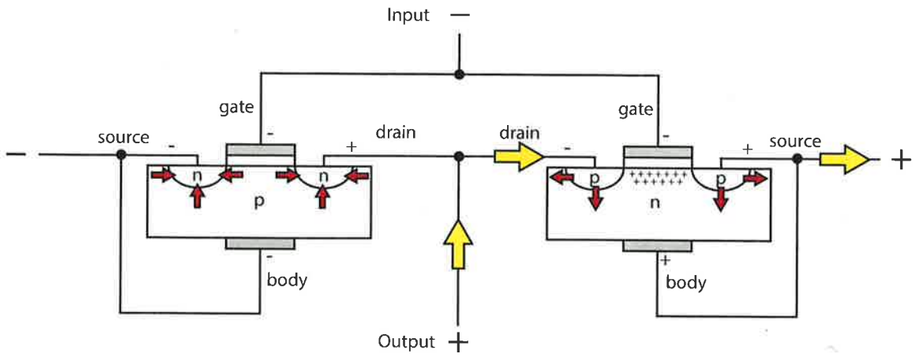
In digital circuits, inputs at a relative high voltage correspond to Boolean `1`'s/`True`'s, while relatively low voltage inputs correspond to logical `0`'s/`False`'s. Under the hood, logic gates are simply circuits made up of pnp and npn-type MOSFET transistors, but when designing complex circuits, it is easier for each to work with abstract logic gates than the transistors that comprise them. Figure 7 shows how we can take advantage of the complementary behavior of pnp and npn-type MOSFETs to construct a `NOT` gate. Although I do not diagram them all here, every logic gate can be implemented in this manner, so when assembling a circuit, one can simply replace the logic gate with its transistor circuit (called a

CMOS circuit, which stands for "Complementary Metal Oxide Semiconductor" circuit). However, for manufacturing efficiency and cost reasons, this is not what happens in practice. It turns out that any logic gate can be implemented using `NOR` gates[3], so, since it is relatively easy to manufacture something using one part arranged in a variety of patterns instead of multiple parts, computer circuits are implemented using only `NOR` gates [2].



(a) When the input voltage is high, the output is low.





(b) When the input voltage is low, the output is high.

Figure 7: This is one implementation of a `NOT` gate using two transistors. Note how the output voltage level is always opposite the input.

[3] A `NOR` gate is defined such that `A NOR B` is equivalent to `NOT (A OR B)`.

3 Volatile Memory

From an implementation perspective, all memory fits into two broad categories: volatile and non-volatile memory. *Volatile memory*, or *powered memory*, refers to memory that stores data temporarily; it stores states as signals in electric circuits, so it is very fast but only retains information as long as it has power running to it. In contrast, non-volatile memory stores data long term, even if the memory does not have a power supply.

At first, the fact that powered memory requires power may seem like a fatally bad quality. However, (barring some highly unexpected breakthrough) powered memory is much faster than unpowered memory, since it is based around flipping signal states of circuits rather than physical states of materials. Because of its speed, volatile memory is used for storing programs and the data they use while they are actually being run; it is the basis for both main memory and the memory on processor caches and registers.

3.1 SRAM

The first type of volatile memory that we're examining is called Static Random Access Memory (SRAM). Random Access Memory (RAM) is simply memory where it is possible to access any location in memory without having to go through all of the other locations. For example, a cassette tape is not random access because you need to scroll through the whole tape to get to a memory location at the end of the tape, while a book is random access because you can flip to any page you want without looking at all of the others. The meaning of the "Static" part of "Static RAM" will become clear once I describe how SRAM is implemented.

3.1.1 Role in computers

SRAM is both very fast and expensive, so it is used when a need for speed is paramount. Recall the requirements of the different memory categories from section 1; since SRAM is powered memory, it could only be used for main memory or for the memory on board the processor. Enough storage space is required for main memory that it would be too expensive to use SRAM for it, so SRAM is more or less exclusively used in computers' processors. In the case of the CPU - the "Central Processing Unit," where most of the computer's calculations take place - memory is used in two ways: as registers and as caches. Registers are the memory most tightly integrated with the processor's calculating circuitry; before any operation is run using some data, that data is loaded into the registers. The purpose of caches is simply to get the data from main memory to the registers as quickly as possible. Caches pull likely relevant data from main memory so that, with any luck, the processor gets to load data into the registers from the

cache instead of the much slower main memory.[4]

3.1.2 Implementation: Latch-based memory

Now for one of the fundamental questions of this paper: how do you get a circuit to "remember" information? In the case of SRAM, the answer is simple: make a circuit that has a feedback loop using logic gates. The circuit's state will represent the 1 or 0 value of the bit. Figure 8 shows a simple example of a feedback loop circuit called a one time latch. The name is appropriate because, as soon as the circuit is set to 1, it remains set to 1 and there is no way to reset it to 0 short of disconnecting the power source. Obviously a one time latch is not a viable way to store data, but the principle it operates on- using a feedback loop to store a bit as a circuit state- is a good one.

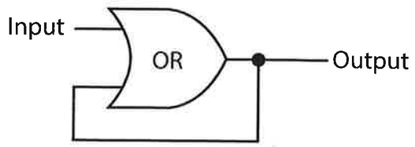


Figure 8: This is the diagram for a one time latch, a simple example of a circuit using a feedback loop to store data. As soon as the input is set to 1, the output gets permanently set to 1 (that is, as long as there is a power source).

The next step up from a one time latch is a set-reset latch, which is depicted in Figure 9. This latch improves dramatically over the one time latch because it has the ability to be reset to 0 after it has been set to 1. However if you look closely at a set-reset latch's behavior given different inputs, you'll find that it has its own share of problems. There are four possible combinations of the inputs RS: 11, 10, 01, and 00. Tracing through the circuit diagram (and starting from various initial conditions) shows that there are no issues with the first three RS combinations. However, the $R = 0 = S$ state poses some problems; if the circuit is in that state and then R and S are simultaneously flipped to 1, then the circuit will oscillate indefinitely between two states. This problem can be avoided by being careful to only flip one of the input bits at a time and only using the four acceptable circuit states shown in Table 1. However, there is a better way.

[4]Technically, caches are implemented using SRAM and registers are not, but the differences are very slight; the typical register implementation only really differs from SRAM in that registers are more tightly integrated with the main logic of the processor. Specifically, there is a direct bit line between each register location and the calculation circuits. [3]

Action	Reset	Set	Output	Output′
set bit to 1	0	1	1	0
reset bit to 0	1	0	0	1
maintain previous state	1	1	0	1
maintain previous state	1	1	1	0

Table 1: These are the four useful states of a set-reset latch being used as a bit of memory. Note that Set and Reset both being set to 1 results in the previous state of the circuit being maintained, no matter what that previous state was.

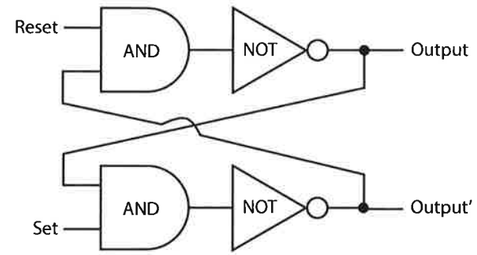


Figure 9: Set-reset latches enable the circuit state to be reset to 0 after it has been set to 1. To set Output = 1, make Reset = 0 and Set = 1. To reset the output to 0, turn Reset to one and then turn Set to zero.

Enter the D-latch. As shown in Figure 10, a D-latch consists of a set-reset latch with a little bit of extra circuitry tacked on the front. In a D-latch, the set S and reset R inputs are not controlled directly; instead, there are two new inputs: E for "enable," and D for "data" [2]. When $E = 1$, the D input sets the value of the circuit, but when $E = 0$, changing the value of D has no effect on the value stored by the circuit's feedback loop. This ability to enable or disable writing data to the bit is critical for ensuring that the computer doesn't have problems like trying to read a bit before it has been written.[5]

[5]In fact, every action a processor takes is regulated by a rapidly oscillating, highly regular clock. Having every action aligned with a clock pulse means that the computer's behavior is controllable and predictable; without the clock, it would be next to impossible to make sure various processor operations happen in the right sequence/ at the right time.

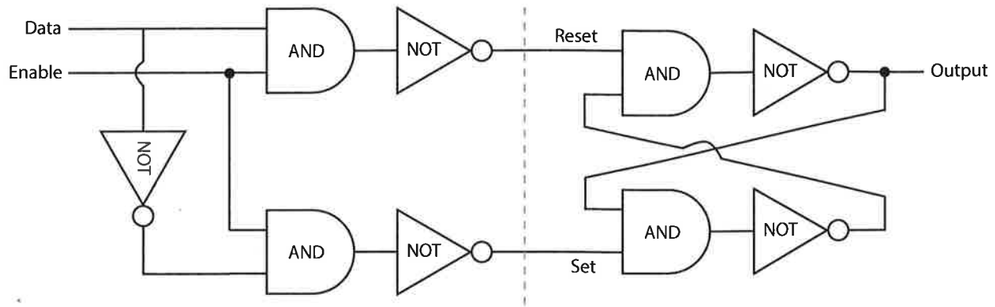


Figure 10: The D-latch is the basic bit for both SRAM and register memory.

SRAM is essentially a grid of D-latch memory cells, however there is an important implementation detail that is worth mentioning. Each transistor added to a circuit increases the cost, manufacturing difficulty, etc. of that circuit, so simply implementing a D-latch by replacing each logic gate with its constituent transistors is less than ideal. Using all NANDs or NORs as described earlier is helpful, but researchers have managed to simplify the circuit even more. Through an ingenious arrangement of transistors, an SRAM D-latch-equivalent memory cell can be implemented with only six transistors! Despite that great achievement, however, SRAM is still more bulky and more expensive (albeit faster) than the other prominent type of volatile memory: dynamic RAM.

3.2 DRAM

Like SRAM, DRAM uses memory cells that store bits as circuit states. Unlike SRAM, however, DRAM memory cells do not employ feedback loops or even logic gates to store the bit state. Instead, DRAM memory cells are capacitor-based. Capacitors are one of the most simple electronics components; they merely consist of two conductors separated by an insulator. When a voltage is applied across a capacitor, charge builds up on each plate corresponding to the applied voltage. As you can see in Figure 11, DRAM memory cells take advantage of this property to effectively store the bit value by charging and discharging a capacitor, such that when the capacitor is charged, the cell has a bit value of `1`. Ideally, this charge would stick around until the two ends of the capacitor get connected via a circuit. However, in practice, the insulator connecting the two plates is not perfect, so the charge from each plate leaks to the other side. For modern DRAM cells, the capacitance is roughly 30 fF, and the leakage current is about 1 fA [4]. Obviously it would be unacceptable for all capacitors set to `1` gradually switch back to `0`, so DRAM memory cells are forced to refresh the charge on their capacitors approximately every 32-64 ms [4]. Naturally those refresh rates are certainly playing it safe. The voltage in such systems is on the order

of 1V, so since

$$Q = VC \tag{5}$$

and

$$I = \frac{\Delta Q}{\Delta t} = \frac{\Delta VC}{\Delta t}\,, \tag{6}$$

it would actually take about $\Delta t = 30$s for a DRAM cell to lose all of its charge [4]. Still, there is much competition to find the best insulator to put between the capacitor plates to mitigate leakage but perform well.

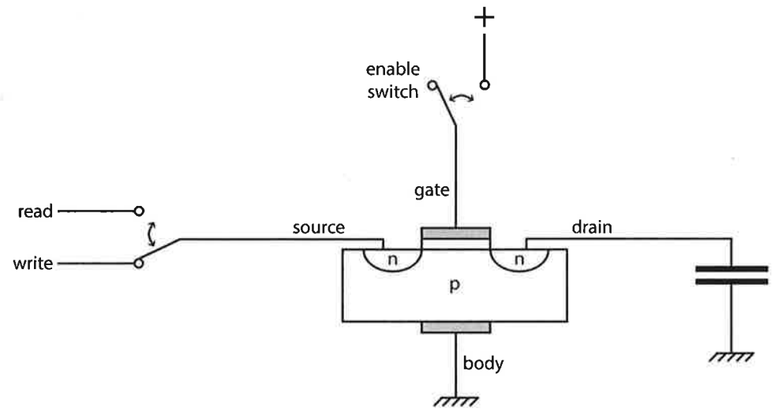


Figure 11: This is the basic schematic for a DRAM memory cell. If there is charge on the capacitor, then the cell holds a `1`.

4 Non-volatile Memory

Unlike volatile memory, technologies that fall under the broad umbrella of "non-volatile memory" do not require power to retain information and are therefore used as secondary memory and removable storage. There are many types of non-volatile memory ranging from simple magnetic tapes to exotic things like (very recently) DNA; I am going to discuss the most common in personal computers: magnetic disk drives and flash memory.

4.1 Magnetic Disk Drives

For many years now, hard disk drives have been the standard choice for secondary storage in personal computers.[6] The fact that they are random-access and nonvolatile coupled with their ability to compactly store increasingly massive amounts of data makes them excellent for storing all of a computer's files and applications long-term.

4.1.1 Implementation: Magnetic domains act as bits

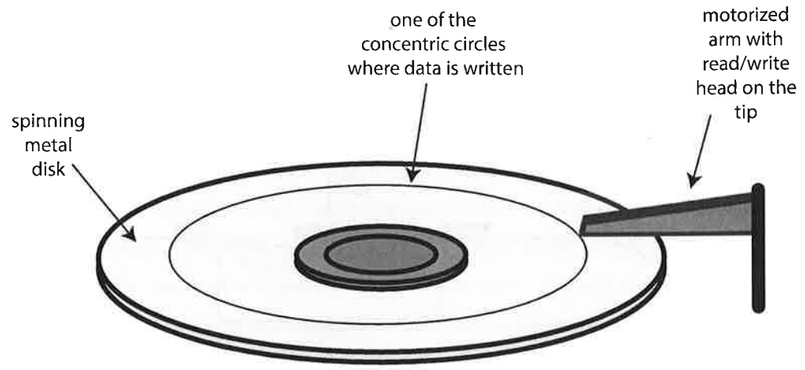
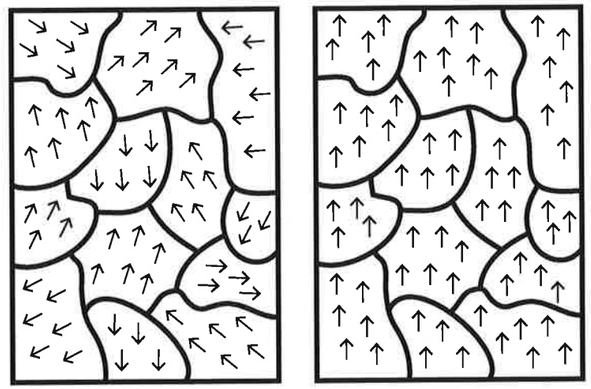


Figure 12: This is a standard disk drive with the basic components labeled.

As shown in Figure 12, the basic design of a hard disk drive consists of a spinning disk and a motorized arm that moves over the top of the disk. A tiny device on the end of the arm reads and writes the data, which the disk actually stores. Incredibly, each bit of data is represented by the direction of magnetization of a tiny region on the disk's surface, which is coated in a ferromagnetic material like iron or cobalt [5, 2]. Ferromagnetic materials are helpfully divided up into many small *magnetic domains*, regions that each act like a tiny magnet. Under normal conditions, these magnetic domains point in many random directions and effectively cancel out each other's effects, as shown in Figure 13(a). However, when exposed to an external magnetic field, as in Figure 13(b), these domains will align to the direction of the magnetic field, and, critically, they will remain aligned even after the external field is no longer present due to hysteresis in the response of magnetization to the strength of the external field.

[6] Spurred on by the mobile electronics industry, flash memory now has a strong share of the market, but disk drives aren't going away any time soon.



(a) (b)

Figure 13: Magnetic domains of a ferromagnetic material align to an external magnetic field. Magnetic disk drives take advantage of this property to store a bit as the direction of magnetization.

To encode data using these regions of aligned domains, you simply have to choose one direction of alignment, eg. North-South, to represent `0` and let the opposite direction represent `1`. Using this system, writing a bit is fairly straightforward: an electromagnet generates an external magnetic field and the bit region's magnetic domains align to it. Switching the polarity of the electromagnet is as simple as changing the direction of the inducing current, and so one electromagnet can write both 1s and 0s. In the basic hard drive design, reading the data back is simply the opposite of writing it [2]. Instead of a current loop generating a magnetic field to align the disk domains, the rapidly moving (but relatively weak) magnetic field of the aligned domains induces a current back in the electromagnet. Figure 14 illustrates this simplified, single electromagnet read/write process.

Of course, I should note that the design described is a gross oversimplification of how modern hard disk drives actually operate. As you would expect, getting increasingly huge amounts of data into a small space requires some extremely impressive engineering. In fact, over 250 steps- many of which are at the edge of manufacturing sophistication- are required to build a single read/write head [5]. The sheer density of the data demands incredible precision from the read/write head, and challenges arise from the fact that the bit domains are so small and close. Because of this, reading and writing are actually performed by distinct components on the read/write head, so that each can be optimized for its specific task [5]. Reading sensors in particular differ significantly from the simple model; they achieve more sophisticated sensing by taking advantage of giant magnetoresistance- very large magnetically induced resistance that

occurs in certain configurations of layered materials [6, 5]. Other notable advances include: writing sensors that use thermal energy to help flip bits, better signal processing that can handle increasingly noisy signals, and- very recently- methods of writing bits perpendicular to the surface of the disk [5]. The tiny size of each bit poses another big challenge: timing. In a fairly typical 7200 rpm, 3.5 in (disk diameter), 80 gigabit/in^2 areal density hard drive, a single bit is about 30 nm [5]. Those are the units you will see when purchasing a HDD; converting to more useful units tells us that the hard drive has a 0.0889 m diameter and spins at 120 rotations/s. By using the rotational kinematic equation

$$v = 2\pi\omega r \ , \tag{7}$$

which relates angular velocity ω in revolutions/s to linear velocity v and radius r in SI units, we find that a bit at the outer edge of the hard drive has a linear velocity of approximately 33.5 m/s. This means that a read/write head with a 60 nm active sensing region only "sees" the 30 nm bit for about 4.5 ns, about 1.3 times the time it takes light to travel a meter in a vacuum [5]. This number is really only a kind of upper bound to single bit read time; it becomes much more impressive when you repeat the calculation for even more dense, state of the art 15,000 rpm consumer drives and factor in some other details, such as the fact that the 60 nm read area can actually read two bits at once thanks to advanced signal processing techniques [5].

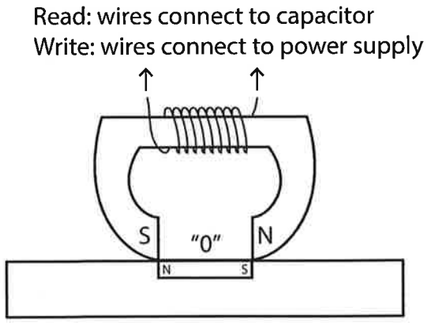


Figure 14: A single head is cleverly used to both read an write data on a hard disk drive. In "write mode" the electromagnet is connected to a power supply and gets activated in whichever direction is needed. In "read mode" that power supply is substituted for a capacitor that picks up a charge when the fast-moving magnetic domains induce a current in the electromagnet.

4.2 Flash Memory

Hard disk drives may be the traditional standard for secondary storage, but as mobile applications have demanded secondary storage that is less susceptible to damage from being flung onto a table or bouncing around in someone's pocket or purse, flash memory has dropped in price and surged in popularity. Flash's speed is such that it is being used for much more than just secondary storage; it is now very common as removable storage (thumb drives) and even as low-level memory that traditionally has been write-once and read-only. In fact, flash is, in some sense, a descendant of some of the old read-only memories, but, as we are about to see, the physics behind flash is also similar in many respects to the volatile memory technologies we've already examined [7].

4.2.1 Implementation: Floating-gate transistors

Like DRAM, a flash bit is essentially the charge held on a conductor. However, flash clearly doesn't store its charge on the metal plate of a capacitor, because the charge would not be held long enough. Instead, flash memory is based around a special transistor, called a *floating gate transistor*, which has an extra "floating" gate that is isolated from the rest of the rest of the transistor, as shown in Figure 15. Used under normal conditions, it is easy to see how a floating gate transistor would behave more or less like the MOSFETs described earlier. The electric field might have to be a bit stronger to create the n-channel or the p-channel because the main gate is further away from the body semiconductor, but otherwise the operation would be pretty similar.

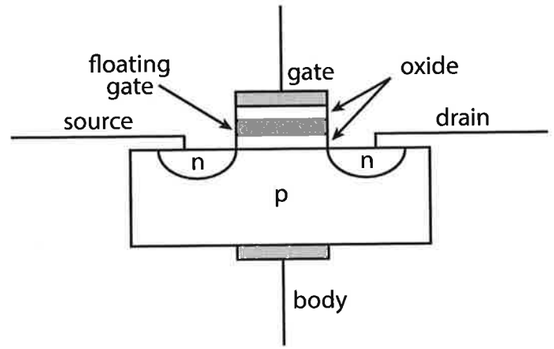


Figure 15: This is the internal structure of a floating gate transistor, the key component behind a flash memory bit.

However, if there is a charge somehow placed on the floating gate, the floating gate transistor will behave differently. Instead of taking the DRAM approach and directly using the stored charge when reading

data, flash memory simply detects how the stored charge impacts the device's operation. Specifically, the threshold voltage at which the transistor "turns on" the controlled current changes. The degree to which this threshold voltage changes can be determined by modeling the floating gate transistor as a group of capacitors, which makes sense because it just consists of conductors and semiconductors separated by insulating layers [7]. Using the relationship $Q = VC$ from equation 5 for each effective capacitor where the floating gate is one plate, I find that the the total charge on the floating gate Q_{FG} is

$$Q_{FG} = C_G(\phi_{FG} - \phi_G) + C_S(\phi_{FG} - \phi_S) + C_D(\phi_{FG} - \phi_D) + C_B(\phi_{FG} - \phi_B) \ , \tag{8}$$

where C_G, C_S, C_D, and C_B are the capacitances between the floating gate and the gate, source, drain, and body, respectively, and the ϕs are the potentials of each component. Solving for the floating gate potential ϕ_{FG} yields

$$\phi_{FG} = \frac{Q_{FG}}{C_T} + \frac{C_G}{C_T}\phi_G + \frac{C_S}{C_T}\phi_S + \frac{C_D}{C_T}\phi_D + \frac{C_B}{C_T}\phi_B \ , \tag{9}$$

where I have defined

$$C_T \equiv C_G + C_S + C_D + C_B \ . \tag{10}$$

By having the source and body both grounded, equation 9 becomes

$$V_{FG} = \frac{Q_{FG}}{C_T} + \frac{C_G}{C_T}V_G + \frac{C_D}{C_T}V_D \text{ [7].} \tag{11}$$

This means that the threshold voltage measured between ground and the floating gate, V_T^{FG}, is related to the threshold voltage measured between ground and the gate, V_T^G, by

$$V_T^{FG} = \frac{Q_{FG}}{C_T} + \frac{C_G}{C_T}V_T^G \ , \tag{12}$$

which is easily rearranged to get

$$V_T^G = \frac{C_T}{C_G}V_T^{FG} - \frac{Q_{FC}}{C_G} \ , \tag{13}$$

the practically measurable threshold voltage of the device [7]. Therefore, when there is a charge Q_{FG} on the floating gate, the threshold voltage V_T at which the floating gate transistor "turns on" changes by

$$\Delta V_T = -\frac{Q_{FC}}{C_G} \ . \tag{14}$$

The practical implication of this result is that having electrons stored on the floating gate measurably increases the threshold voltage of the floating gate transistor. This effect gives us a great way to sense whether or not a charge is stored on the floating gate: we can simply check to see if the floating gate transistor behaves normally or not. We smoothly vary the control voltage from 0 V to a high value; if

the transistor conducts at a normal value, the gate is uncharged, and the bit is in the 1 state [7, 2]. If, however, the transistor conducts at a higher voltage than normal, the bit value is 0.

So reading the bit can be easily achieved, but there is still one obvious challenge: how can we charge the electrically insulted floating gate to actually write data? Quantum mechanics provides the solution. It turns out that the insulation surrounding the floating gate is sufficiently thin (approx. 10 nm) that, when a great enough voltage is applied between the gate and the body, electrons will tunnel through the insulating oxide to the floating gate in a process called Fowler-Nordheim tunneling [7, 2]. Similarly, erasing is accomplished by using a high negative voltage to drive the electrons back off of the floating gate [7]. All of this tunneling and high applied voltages will cause the system to break down eventually, but advances in material science and manufacturing processes have rendered flash a more than viable form of nonvolatile memory [7].

5 Conclusion

From quantum tunneling to CMOS feedback loops, the physical underpinnings of digital memory are fascinating and diverse. Although the technologies I've described in this paper are highly refined and serve millions of computers well, it is important to remember that bits are just simple abstract 1s or 0s. We're constantly looking to invent creative new physical systems to implement bits in fast, dense, and reliable ways. Recent developments range from continued improvements in transistor and magnetic domain density to more exotic new technologies that include bits stored in DNA sequences and single, carefully controlled iron atoms.

Even an older, mundane computer is a phenomenal physical system. Think about it. To work on this paper, my laptop has to: take voltage signals induced by thousands of tiny magnetic regions on a rapidly spinning disk, move them to constantly refreshing capacitors, and then store them in looping circuits, which in turn are only able to work because some materials have more available electrons than others. Perhaps somewhere in the process, I move to a different computer that forgoes the metal disk in favor of exploiting the fact that electrons have some probability of being in places they really shouldn't be classically. It's incredible- and yet thoroughly explained by physics.

Acknowledgments

I would like to thank Prof. Melissa Eblen-Zayas for advising me, Prof. Matt Wiebold and Peter Bumcrot for reviewing this paper, and the Carleton College physics department for preparing me to research this topic.

References

[1] David J. Roulston. *An Introduction to the Physics of Semiconductor Devices.* Oxford University Press, 1999.

This book gives a lot of good background on semiconductors and MOSFETs as well as a little bit about their applications. It goes into more depth than my other general semiconductor and transistor sources.

[2] Michael G. Raymer. *The Silicon Web: Physics for the Internet Age.* Taylor & Francis, 2009.

This textbook is designed to teach an introductory Physics class using examples relevant to computers. It gives a good overview of many of the topics in this paper. In particular, ch. 3 has information on the mechanics involved with hard drives, ch. 5 describes some relevant E&M- including the physics behind DRAM, ch. 10 goes into depth about semiconductors and transistors, and ch. 11 discusses a lot of memory topics: latches, SRAM, DRAM, flash memory, hard drives, and optical storage.

[3] David A. Patterson and John L. Hennessy. Storage and other I/O topics. In *Computer Organization and Design, 4^{th} edition.* Morgan Kaufmann Publishers, 2009.

This textbook explains the fundamental (and some of the not so fundamental) ways computers are designed, from a low-level computer science perspective. Additionally, the bonus sections included on the CD (removable storage!) that comes with the book are particularly helpful in describing the circuits behind SRAM and register files.

[4] Bruce Jacob, Spencer Ng, and David Wang. *Memory Systems: Cache, DRAM, Disk.* Morgan Kaufmann, 2008.

This book goes into depth about the three technologies in its title. It includes physics, but mostly is written at the architecture level.

[5] J.R. Childress and R.E. Fontana Jr. Magnetic recording read head sensor technology. *C.R. Physique*, (6), 2005.

This article provides an excellent, pretty in-depth explanation of how the read/write head works on magnetic disk drives, with specific emphasis on the reading mode. It also contains some key details about the engineering and physics behind other components of HDDs.

[6] Class for Physics of the Royal Swedish Academy of Sciences. The discovery of giant magnetoresistance. 2007. `http://www.nobelprize.org/nobel_prizes/physics/laureates/2007/advanced-physicsprize2007.pdf`.

This is a background paper on the 2007 Nobel prize for the discovery of giant magnetoresistance. It has a lot of good information about the physics, but it does not talk much about magnetic sensor applications. This paper was very useful in understanding Childress and Fontana's paper on the read/write head of magnetic drives.

[7] P Pavan, R Bez, and P et al. Olivo. Flash memory cells - an overview. *Proceedings of the IEEE*, 85(8):1248–1271, August 1997. `http://www.google.com/url?sa=t&rct=j&q=flash%20memory%20cells%20-%20an%20overview&source=web&cd=1&cad=rja&ved=0CDIQFjAA&url=ftp%3A%2F%2F202.120.40.101%2FLegacy%2Fpaper%2FByBook%2FCAAQA(4th)%2F~index%2FPavan.1997.Flash%2520memory%2520cells-an%2520overview.pdf&ei=hgUOUb_4NsTbyAHEyYDICQ&usg=AFQjCNHi8SxiXHPWNlQ2E3jl2dw-vi4x7g&bvm=bv.41867550,d.aWc`.

This source, though older and a bit outdated on some of the details, provides a nice look at how flash memory cells work.

[8] John R. Taylor, Chris D. Zafiratos, and Michael A. Dubson. Ch. 14: Solids - Applications. In *Modern Physics for Scientists and Engineers, 2^{nd} edition.* Prentice Hall, 2004.

This chapter provides a solid background on the physics of semiconductors and transistors.

[9] Edward M. Purcell. McGraw-Hill, 1985.

This textbook is a good resource for getting an overview of electricity and magnetism related topics. I used it primarily as a general reference when other books didn't go into enough depth or when I wanted an alternate explanation of some phenomenon.