CHAPTER 1

Embedded Hardware Basics

Jack Ganssle
Tammy Noergaard

1.1 Lesson One on Hardware: Reading Schematics

This section is equally important for embedded hardware and software engineers. Before diving into the details, note that it is important for all embedded designers to be able to understand the diagrams and symbols that hardware engineers create and use to describe their hardware designs to the outside world. These diagrams and symbols are the keys to quickly and efficiently understanding even the most complex hardware design, regardless of how much or little practical experience one has in designing hardware. They also contain the information an embedded programmer needs to design any software that requires compatibility with the hardware, and they provide insight to a programmer as to how to successfully communicate the hardware requirements of the software to a hardware engineer.

There are several different types of engineering hardware drawings, including:

- *Block diagrams*, which typically depict the major components of a board (processors, buses, I/O, memory) or a single component (a processor, for example) at a systems architecture or higher level. In short, a block diagram is a basic overview of the hardware, with implementation details abstracted out. While a block diagram can reflect the actual physical layout of a board containing these major components, it mainly depicts how different components or units within a component function together at a systems architecture level. Block diagrams are used extensively throughout this book (in fact, Figures 1.5a–e later in this chapter are examples of block diagrams) because they are the simplest method by which to depict and describe the components within a system. The symbols used within a block diagram are simple, such as squares or rectangles for chips and straight lines for buses. Block diagrams are typically not detailed enough for a software designer to be able to write all the low-level software accurately enough to control the hardware (without a lot of headaches, trial and error, and even some burned-out hardware!). However, they are very useful in communicating a basic overview of the hardware, as well as providing a basis for creating more detailed hardware diagrams.
- *Schematics*. Schematics are electronic circuit diagrams that provide a more detailed view of all the devices within a circuit or within a single component—everything from

processors down to resistors. A schematic diagram is not meant to depict the physical layout of the board or component, but provides information on the flow of data in the system, defining what signals are assigned where—which signals travel on the various lines of a bus, appear on the pins of a processor, and so on. In schematic diagrams, *schematic symbols* are used to depict all the components within the system. They typically do not look anything like the physical components they represent but are a type of "shorthand" representation based on some type of schematic symbol standard. A schematic diagram is the most useful diagram to both hardware and software designers trying to determine how a system actually operates, to debug hardware, or to write and debug the software managing the hardware. See Appendix A for a list of commonly used schematic symbols.

- *Wiring diagrams*. These diagrams represent the bus connections between the major and minor components on a board or within a chip. In wiring diagrams, vertical and horizontal lines are used to represent the lines of a bus, and either schematic symbols or more simplified symbols (that physically resemble the other components on the board or elements within a component) are used. These diagrams may represent an approximate depiction of the physical layout of a component or board.

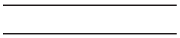
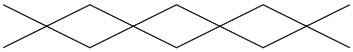
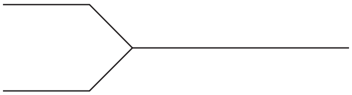
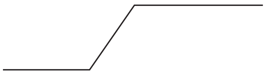
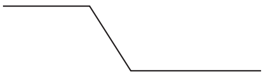
- *Logic diagrams/prints*. Logic diagrams/prints are used to show a wide variety of circuit information using logical symbols (AND, OR, NOT, XOR, and so on) and logical inputs and outputs (the 1's and 0's). These diagrams do not replace schematics, but they can be useful in simplifying certain types of circuits in order to understand how they function.

- *Timing diagrams*. Timing diagrams display timing graphs of various input and output signals of a circuit, as well as the relationships between the various signals. They are the most common diagrams (after block diagrams) in hardware user manuals and data sheets.

Regardless of the type, to understand how to read and interpret these diagrams, it is important to first learn the standard *symbols, conventions*, and *rules* used. Examples of the symbols used in timing diagrams are shown in Table 1.1, along with the conventions for input/output signals associated with each of the symbols.

An example of a timing diagram is shown in Figure 1.1. In this figure, each row represents a different signal. In the case of the signal rising and falling symbols within the diagram, the *rise time* or *fall time* is indicated by the time it takes for the signal to move from LOW to HIGH or vice versa (the entire length of the diagonal line of the symbol). In comparing two signals, a delay is measured at the center of the rising or falling symbols of each signal being compared. In Figure 1.1, there is a fall time delay between signals B and C and signals A and C in the first falling symbol. In comparing the first falling symbol of signals A and B in the figure, no delay is indicated by the timing diagram.

Table 1.1: Timing diagrams symbol table.[1.1]

Symbol	Input Signals	Output Signals
	Input signal must be valid	Output signal will be valid
	Input signal doesn't affect system, will work regardless	Indeterminate output signal
	Garbage signal (nonsense)	Output signal not driven (floating), tristate, HiZ, high impedance
	If the input signal rises:	Output signal will rise
	If the input signal falls:	Output signal will fall

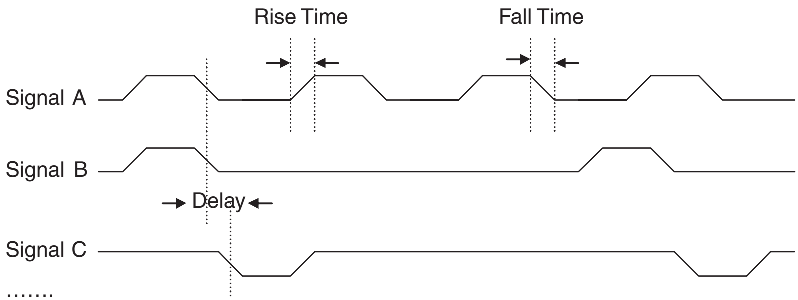


Figure 1.1: Timing diagram example.

Schematic diagrams are much more complex than their timing diagram counterparts. As introduced earlier this chapter, schematics provide a more detailed view of all the devices within a circuit or within a single component. Figure 1.2 shows an example of a schematic diagram.

In the case of schematic diagrams, some of the conventions and rules include:

- A *title section* is located at the bottom of each schematic page, listing information that includes, but is not limited to, the name of the circuit, the name of the hardware engineer responsible for the design, the date, and a list of revisions made to the design since its conception.
- The use of *schematic symbols* indicating the various components of a circuit (see Appendix A).

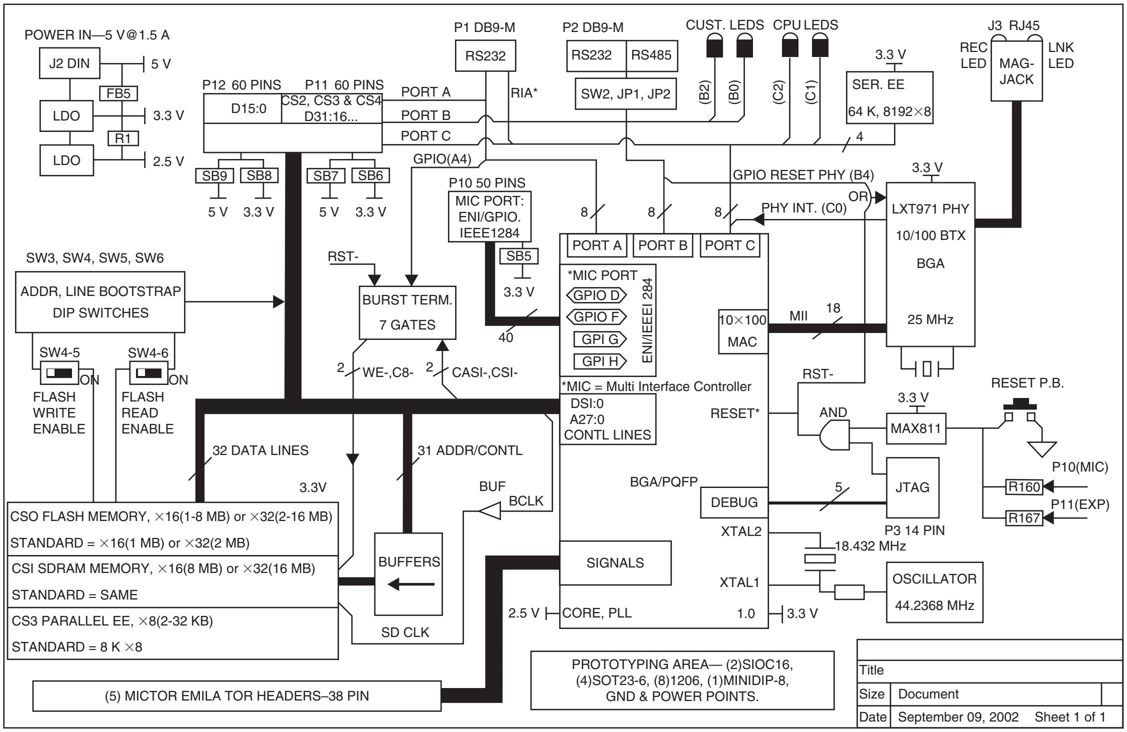


Figure 1.2: Schematic diagram example.[1.2]

- Along with the assigned symbol comes a *label* that details information about the component (i.e., size, type, power ratings, etc.). Labels for components of a symbol, such as the pin numbers of an IC, signal names associated with wires, and so forth are usually located outside of the schematic symbol.

- *Abbreviations* and *prefixes* are used for common units of measurement (i.e., *k* for kilo or 103, *M* for mega or 106) and these prefixes replace writing out the units and larger numbers.

- *Functional groups* and subgroups of components are typically separated onto different pages.

- *I/O* and *voltage source/ground terminals*. In general, positive voltage supply terminals are located at the top of the page, and negative supply/ground at the bottom. Input components are usually on the left, and output components are on the right.

At the very least, the block and schematic diagrams should contain nothing unfamiliar to anyone working on the embedded project, whether they are coding software or prototyping the hardware. This means becoming familiar with everything from where the name of the diagram is located to how the states of the components shown within the diagrams are represented.

One of the most efficient ways of learning how to learn to read and/or create a hardware diagram is via the *Traister* and *Lisk* method[1.3], which involves:

Step 1. Learning the basic symbols that can make up the type of diagram, such as timing or schematic symbols. To aid in the learning of these symbols, rotate between this step and steps 2 and/or 3.

Step 2. Reading as many diagrams as possible until reading them becomes boring (in that case, rotate between this step and steps 1 and/or 3) or comfortable (so there is no longer the need to look up every other symbol while reading).

Step 3. Writing a diagram to practice simulating what has been read, again until it becomes either boring (which means rotating back through steps 1 and/or 2) or comfortable.

1.2 The Embedded Board and the von Neumann Model

In embedded devices, all the electronics hardware resides on a board, also referred to as a *printed wiring board (PW)* or *printed circuit board (PCB)*. PCBs are often made of thin sheets of fiberglass. The electrical path of the circuit is printed in copper, which carries the electrical signals between the various components connected on the board. All electronic components that make up the circuit are connected to this board, either by soldering, plugging into a socket, or some other connection mechanism. All the hardware on an embedded board is located in the hardware layer of the Embedded Systems Model (see Figure 1.3).

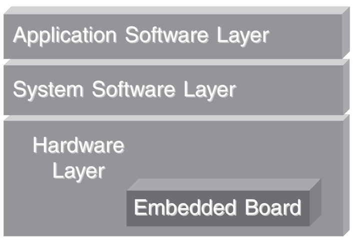


Figure 1.3: Embedded board and the Embedded Systems Model.

At the highest level, the major hardware components of most boards can be classified into five major categories:

- *Central processing unit (CPU).* The master processor.
- *Memory*. Where the system's software is stored.
- *Input device(s).* Input slave processors and relative electrical components.

- *Output device(s).* Output slave processors and relative electrical components.
- *Data pathway(s)/bus(es).* Interconnects the other components, providing a "highway" for data to travel on from one component to another, including any wires, bus bridges, and/or bus controllers.

These five categories are based on the major elements defined by the *von Neumann model* (see Figure 1.4), a tool that can be used to understand any electronic device's hardware architecture. The von Neumann model is a result of the published work of John von Neumann in 1945, which defined the requirements of a general-purpose electronic computer. Because embedded systems are a type of computer system, this model can be applied as a means of understanding embedded systems hardware.

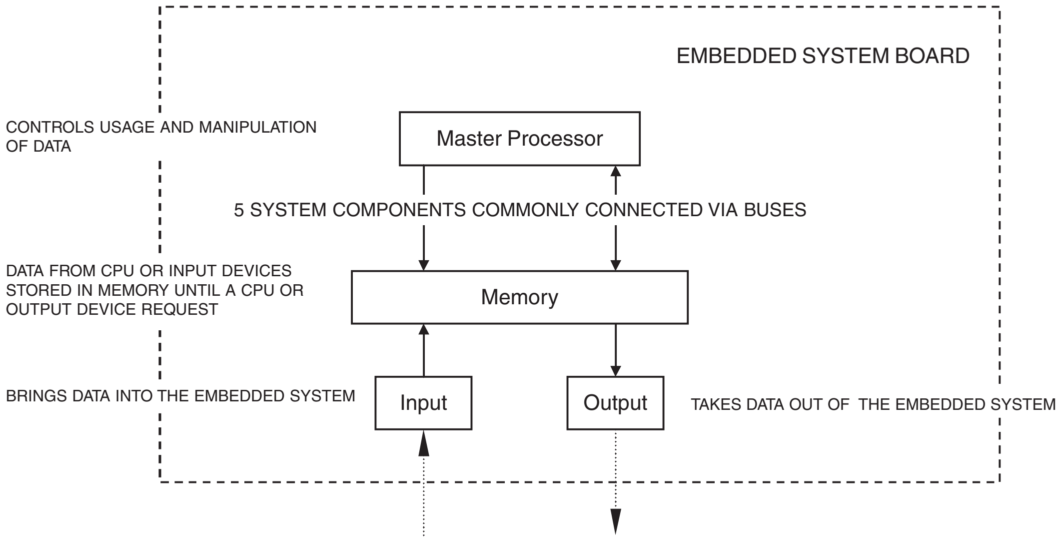
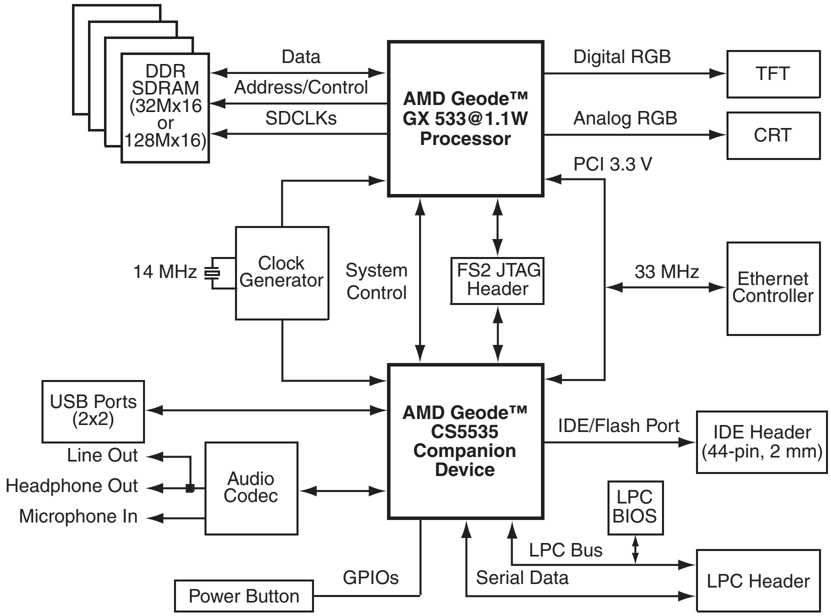


Figure 1.4: Embedded system board organization.[1.4]

Based on the von Neumann architecture model (also referred to as the Princeton architecture).

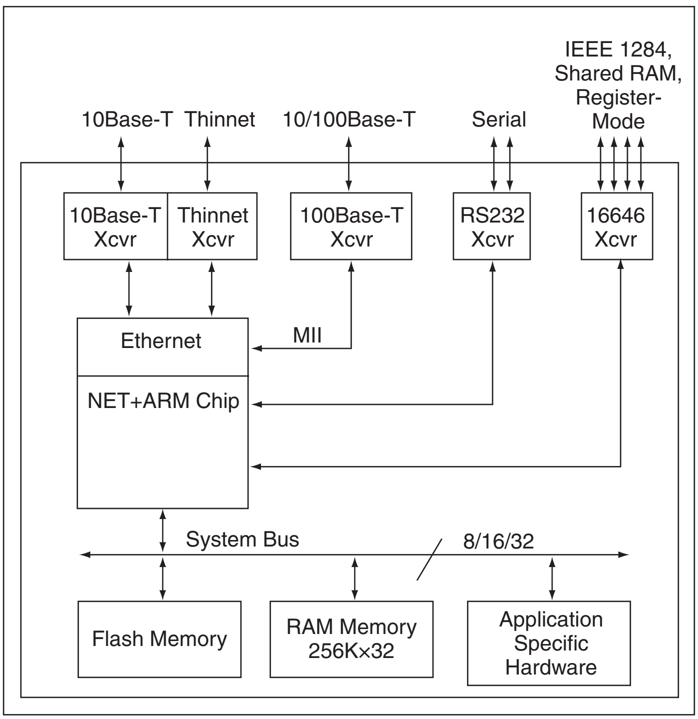
While board designs can vary widely, as demonstrated in the examples of Figures 1.5a–d, all the major elements on these embedded boards—*and on just about any embedded board*—can be classified as either the master CPU(s), memory, input/output, or bus components.

To understand how the major components on an embedded board function, it is useful to first understand what these components consist of and why. All the components on an embedded board, including the major components introduced in the von Neumann model, are made up of one or some combination of interconnected *basic electronic devices*, such as wires, resistors,



- **Master Processor:** Geode GX533@1.1w (x86)
- **Memory:** ROM (BIOS is located in), SDRAM
- **Input/Output Devices:** CS5535, Audio Codec...
- **Buses:** LPC,PCI

Figure 1.5a: AMD/National Semiconductor x86 reference board.[1.5]



- **Master Processor:** Net+ARM ARM7
- **Memory:** Flash, RAM
- **Input/Output Devices:** 10Base-T transceiver, Thinnet transceiver, 100Base-T transceiver, RS-232 transceiver, 16646 transceiver, …
- **Buses:** System Bus, MII, …

Figure 1.5b: Net Silicon ARM7 reference board.[1.6]

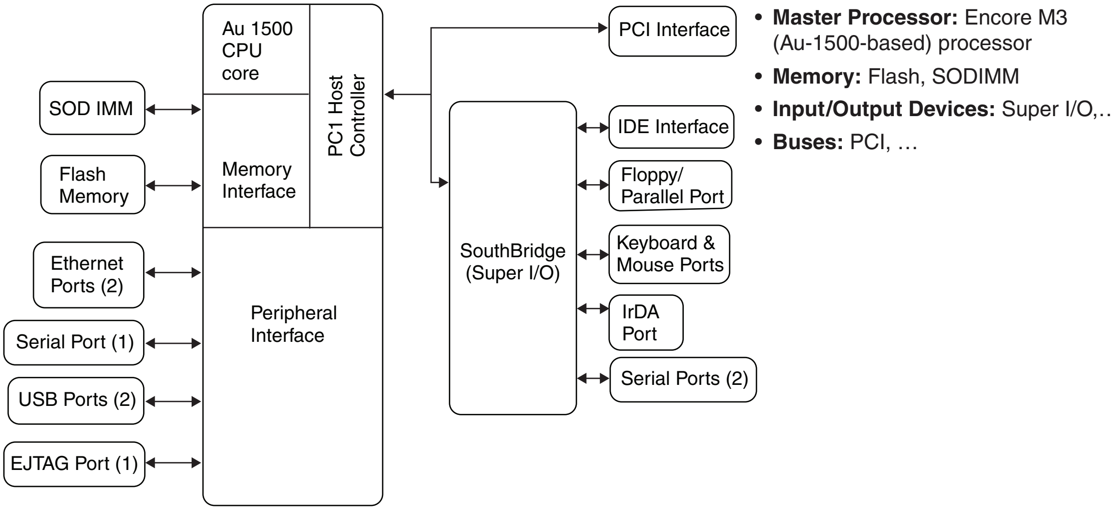


Figure 1.5c: Ampro MIPS reference board.[1.7]

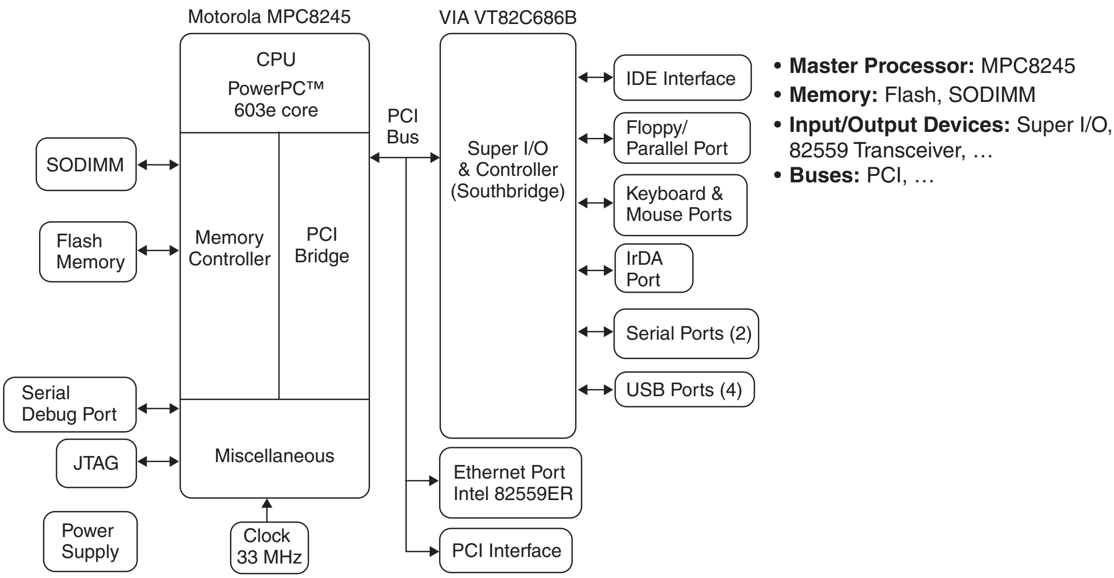


Figure 1.5d: Ampro PowerPC reference board.[1.8]

capacitors, inductors, and diodes. These devices also can act to connect the major components of a board together. At the highest level, these devices are typically classified as either *passive* or *active* components. In short, passive components include devices such as wires, resistors, capacitors and inductors that can only receive or store power. Active components, on the other hand, include devices such as transistors, diodes, and integrated circuits (ICs) that are capable

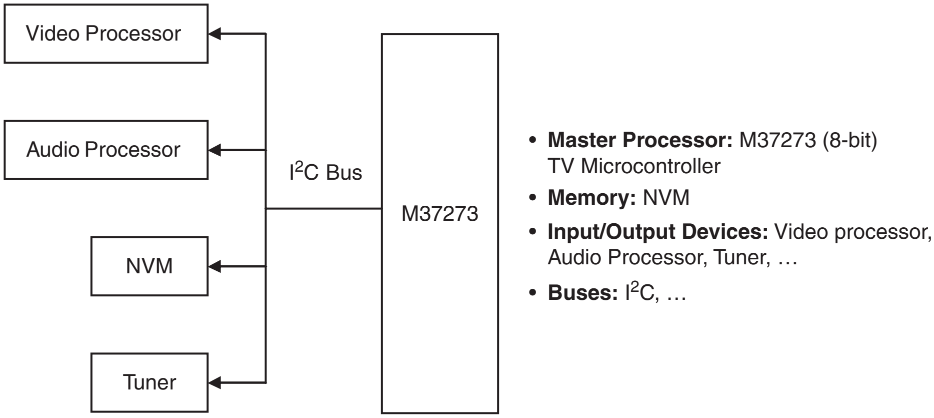


Figure 1.5e: Mitsubishi analog TV reference board.

of delivering as well as receiving and storing power. In some cases, active components themselves can be made up of passive components. Within the passive and active families of components, these circuit devices essentially differ according to how they respond to *voltage* and *current*.

1.3 Powering the Hardware

Power is the rate that energy is expended or work is performed. This means that in alternating current (AC) and direct current (DC) circuits, the power associated with each element on the board equals the current through the element multiplied by the voltage across the element ($P = VI$). Accurate power and energy calculations must be done for all elements on an embedded board to determine the power consumption requirements of that particular board. This is because each element can only handle a certain type of power, so AC-DC converters, DC-AC converters, direct AC-AC converters, and so on may be required. Also, each element has a limited amount of power that it requires to function, that it can handle, or that it dissipates. These calculations determine the type of voltage source that can be used on a board and how powerful the voltage source needs to be.

In embedded systems, both AC and DC voltage sources are used because each current generation technique has its pros and cons. AC is easier to generate in large amounts using generators driven by turbines turned by everything from wind to water. Producing large amounts of DC from electrochemical cells (batteries) is not as practical. Also, because transmitting current over long transmission lines results in a significant loss of energy due to the resistance of the wire, most modern electric company facilities transmit electricity to outlets in AC current, since AC can be transformed to lower or higher voltages much more easily than DC. With AC, a device called a *transformer*, located at the service provider, is used to efficiently transmit current over long distances with lower losses. The transformer is a device that transfers electrical energy from one circuit to another and can make changes to the current and voltage during the transfer. The service provider transmits lower levels of current at a higher voltage rate from the power plant, and then a transformer at the customer site decreases the voltage to the value required. On the flip side, at very high voltages, wires offer less resistance to DC than AC, thus making DC more efficient to transmit than AC over very long distances.

Some embedded boards integrate or plug into *power supplies*. Power supplies can be either AC or DC. To use an AC power supply to supply power to components using only DC, an AC-to-DC converter can be used to convert AC to the lower DC voltages required by the various components on an embedded board, which typically require 3.3, 5, or 12 volts.

> **Note:** Other types of converters, such as DC-to-DC, DC-to-AC, or direct AC-to-AC can be used to handle the required power conversions for devices that have other requirements.

Other embedded boards or components on a board (such as nonvolatile memory, discussed in more detail in Chapter 5) rely on *batteries* as voltage sources, which can be more practical for providing power because of their size. Battery-powered boards don't rely on a power plant for energy, and they allow portability of embedded devices that don't need to be plugged into an outlet. Also, because batteries supply DC current, no mechanism is needed to convert AC to DC for components that require DC, as is needed with boards that rely on a power supply and outlet supplying AC. Batteries, however, have a limited life and must be either recharged or replaced.

1.3.1 A Quick Comment on Analog vs. Digital Signals

A digital system processes only digital data, which is data represented by only 0's and 1's. On most boards, two voltages represent "0" and "1," since all data is represented as some combination of 1's and 0's. No voltage (0 volts) is referred to as ground, VSS, or low, and 3, 5, or 12 volts are commonly referred to as VCC, VDD, or high. All signals within the system are one of the two voltages or are transitioning to one of the two voltages. Systems can define "0" as low and "1" as high, or some range of 0–1 volts as LOW and 4–5 volts as HIGH, for instance. Other signals can base the definition of a "1" or "0" on edges (low to high) or (high to low).

Because most major components on an embedded board, such as processors, inherently process the 1's and 0's of digital signals, a lot of embedded hardware is digital by nature. However, an embedded system can still process analog signals, which are continuous—that is, not only

1's and 0's but values in between as well. Obviously, a mechanism is needed on the board to convert analog signals to digital signals. An analog signal is digitized by a sampling process, and the resulting digital data can be translated back into a voltage "wave" that mirrors the original analog waveform.

Real-World Advice

Inaccurate Signals: Problems with Noise in Analog and Digital Signals

One of the most serious problems in both the analog and digital signal realm involves noise distorting incoming signals, thus corrupting and affecting the accuracy of data. Noise is generally any unwanted signal alteration from an input source, any part of the input signal generated from something other than a sensor, or even noise generated from the sensor itself. Noise is a common problem with analog signals. Digital signals, on the other hand, are at greater risk if the signals are not generated locally to the embedded processor, so any digital signals coming across a longer transmission medium are the most susceptible to noise problems.

Analog noise can come from a wide variety of sources—radio signals, lightning, power lines, the microprocessor, or the analog sensing electronics themselves. The same is true for digital noise, which can come from mechanical contacts used as computer inputs, dirty slip rings that transmit power/data, limits in accuracy/dependability of input source, and so forth.

The key to reducing either analog or digital noise is: (1) to follow basic design guidelines to avoid problems with noise. In the case of analog noise, this includes not mixing analog and digital grounds, keeping sensitive electronic elements on the board a sufficient distance from elements switching current, limiting length of wires with low signal levels/high impedance, etc. With digital signals, this means routing signal wires away from noise-inducing high current cables, shielding wires, transmitting signals using correct techniques, etc. (2) to clearly identify the root cause of the problem, which means exactly what is causing the noise. With point (2), once the root cause of the noise has been identified, a hardware or software fix can be implemented. Techniques for reducing analog noise include filtering out frequencies not needed and averaging the signal inputs, whereas digital noise is commonly addressed via transmitting correction codes/parity bits and/or adding additional hardware to the board to correct any problems with received data.

—Based on the articles "Minimizing Analog Noise" (May 1997),"Taming Analog Noise" (November 1992), and "Smoothing Digital Inputs" (October 1992), by Jack Ganssle, in Embedded Systems Programming Magazine.

1.4 Basic Electronics

In this section, we will review some electronics fundamentals.

1.4.1 DC Circuits

DC means *direct current*, a fancy term for signals that don't change. They're flatlined, like a corpse's EEG or the output from a battery (Figure 1.6). Your PC's power supply makes DC out of the building's AC (alternating current) mains. All digital circuits require DC power supplies.

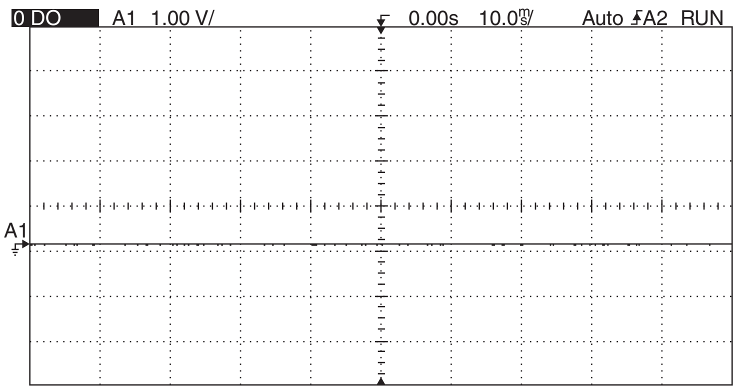


Figure 1.6: A DC signal has a constant, unvarying amplitude.

1.4.1.1 Voltage and Current

We measure the quantity of electricity using voltage and amperage, but both arise from more fundamental physics. Atoms that have a shortage or surplus of electrons are called *ions*. An ion has a positive or negative charge. Two ions of opposite polarity (one plus, meaning it's missing electrons, and the other negative, with one or more extra electrons) attract each other. This attractive force is called the *electromotive force*, commonly known as EMF.

Charge is measured in *coulombs*, where one coulomb is 6.25×10^{18} electrons (for negative charges) or protons for positive ones.

An *ampere* is one coulomb flowing past a point for one second. *Voltage* is the force between two points for which one ampere of current will do one *joule* of work, a joule per second being one watt.

Figure 1.7: A VOM, even an old-fashioned analog model like this $10 Radio Shack model, measures DC voltage as well or better than a scope.

But few electrical engineers remember these definitions, and none actually use them.

An old but still apt analogy uses water flow through a pipe: Current would be the amount of water flowing through a pipe per unit of time, whereas voltage is the pressure of the water.

The unit of current is the ampere (amp), though in computers an amp is an awful lot of current. Most digital and analog circuits require much less. Table 1.2 shows the most common nomenclatures.

Table 1.2: Common nomenclatures.

Name	Abbreviation	Number of Amps	Where Likely Found
amp	A	1	Power supplies; very high-performance processors may draw many tens of amps
milliamp	mA	.001 amp	Logic circuits, processors (tens or hundreds of mA), generic analog circuits
microamp	µA	10^{-6} amp	Low-power logic, low-power analog, battery-backed RAM
picoamp	pA	10^{-12} amp	Very sensitive analog inputs
femtoamp	fA	10^{-15} amp	The cutting edge of low-power analog measurements

Most embedded systems have a far less extreme range of voltages. Typical logic and microprocessor power supplies range from a volt or 2–5 volts. Analog power supplies rarely exceed plus and minus 15 volts. Some analog signals from sensors might go down to the millivolt (.001 volt) range. Radio receivers can detect microvolt-level signals, but they do this using quite sophisticated noise-rejection techniques.

1.4.1.2 Resistors

As electrons travel through wires, components, or accidentally through a poor soul's body, they encounter *resistance*, which is the tendency of the conductor to limit electron flow.

A vacuum is a perfect resistor: no current flows through it. Air's pretty close, but since water is a decent conductor, humidity does allow some electricity to flow in air.

Superconductors are the only materials with zero resistance, a feat achieved through the magic of quantum mechanics at extremely low temperatures, on the order of that of liquid nitrogen and colder. Everything else exhibits some resistance, even the very best wires. Feel the power cord of your 1500 watt ceramic heater—it's warm, indicating some power is lost in the cord due to the wire's resistance.

We measure resistance in ohms; the more ohms, the poorer the conductor. The Greek capital omega (Ω) is the symbol denoting ohms.

Resistance, voltage, and amperage are all related by the most important of all formulas in electrical engineering. Ohm's Law states:

$$E = I \times R$$

where E is voltage in volts, I is current in amps, and R is resistance in ohms. (EEs like to use E for volts because it indicates electromotive force.)

What does this mean in practice? Feed one amp of current through a one-ohm load and there will be one volt developed across the load. Double the voltage and, if resistance stays the same, the current doubles.

Though all electronic components have resistance, a *resistor* is a device specifically made to reduce conductivity (Figure 1.8 and Table 1.3). We use them everywhere. The volume control on a stereo (at least, the nondigital ones) is a resistor whose value changes as you rotate the knob; more resistance reduces the signal and hence the speaker output.

What happens when you connect resistors together? For resistors in series, the total effective resistance is the sum of the values:

$$R_{\text{eff}} = R_1 + R_2$$

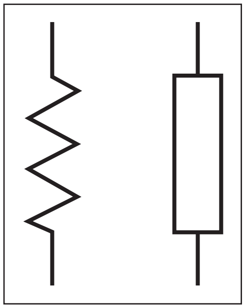


Figure 1.8: The squiggly thing on the left is the standard symbol used by engineers to denote a resistor on their schematics. On the right is the symbol used by engineers in the United Kingdom. As Churchill said, we are two peoples divided by a common language.

Table 1.3: Range of values for real-world resistors.

Name	Abbreviation	Ohms	Where Likely Found
milliohm	m Ω	.001 ohm	Resistance of wires and other good conductors
ohm	Ω	1 ohm	Power supplies may have big dropping resistors in the few to tens of ohms range
hundreds of ohms			In embedded systems, it's common to find resistors in the few hundred ohm range used to terminate high-speed signals
kiloohm	k Ω or just k	1000 ohms	Resistors from a half-k to a hundred or more k are found all over every sort of electronic device; "pullups" are typically a few k to tens of k
megaohm	M Ω	10^6 ohms	Low signal-level analog circuits
hundreds of M Ω		10^8++ ohms	Geiger counters and other extremely sensitive apps; rarely seen since resistors of this size are close to the resistance of air

For two resistors in parallel, the effective resistance is:

$$R_{\text{eff}} = \frac{R_1 \times R_2}{R_1 + R_2}$$

(Thus, two identical resistors in parallel are effectively half the resistance of either of them: two 1 ks is 500 ohms. Now add a third: that's essentially a 500-ohm resistor in parallel with a 1k, for an effective total of 333 ohms.)

The general formula for more than two resistors in parallel (Figure 1.9) is:

$$R_{\text{eff}} = \frac{1}{\dfrac{1}{R_1} + \dfrac{1}{R_2} + \dfrac{1}{R_3} + \dfrac{1}{R_4} + \cdots}$$

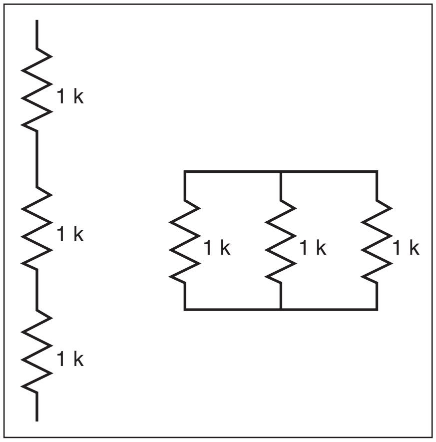


Figure 1.9: The three series resistors on the left are equivalent to a single 3000-ohm part. The three paralleled on the right work out to one 333-ohm device.

Manufacturers use color codes to denote the value of a particular resistor. Although at first this may seem unnecessarily arcane, in practice it makes quite a bit of sense. Regardless of orientation, no matter how it is installed on a circuit board, the part's color bands are always visible (Figure 1.10 and Table 1.4).

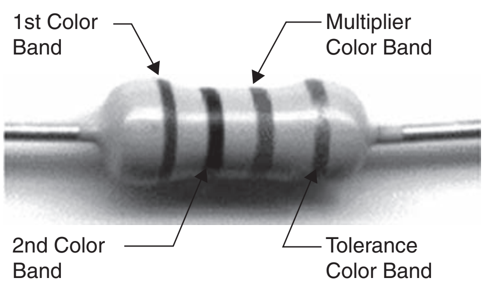


Figure 1.10: This black-and-white photo masks the resistor's color bands. However, we read them from left to right, the first two designating the integer part of the value, the third band giving the multiplier. A fourth gold (5%) or silver (10%) band indicates the part's tolerance.

Table 1.4: The resistor color code. Various mnemonic devices designed to help one remember these are no longer politically correct; one acceptable but less memorable alternative is Big Brown Rabbits Often Yield Great Big Vocal Groans When Gingerly Slapped.

Color band	Value	Multiplier
Black	0	1
Brown	1	10
Red	2	100
Orange	3	1000
Yellow	4	10,000
Green	5	100,000
Blue	6	1,000,000
Violet	7	Not used
Gray	8	Not used
White	9	Not used
Gold (third band)		÷10
Silver (third band)		÷100

The first two bands, reading from the left, give the integer part of the resistor's value. The third is the multiplier. Read the first two bands' numerical values and multiply by the scale designated by the third band. For instance: brown black red = 1 (brown) 0 (black) times 100 (red), or 1000 ohms, more commonly referred to as 1 k. Table 1.5 has more examples.

Table 1.5: Examples showing how to read color bands and compute resistance.

First Band	Second Band	Third Band	Calculation	Value (Ohms)	Commonly Called
Brown	Red	Orange	12 × 1000	12,000	12 k
Red	Red	Red	22 × 100	2,200	2.2 k
Orange	Orange	Yellow	33 × 10,000	330,000	330 k
Green	Blue	Red	56 × 100	5,600	5.6 k
Green	Blue	Green	56 × 100,000	5,600,000	5.6 M
Red	Red	Black	22 × 1	22	22
Brown	Black	Gold	10 ÷10	1	1
Blue	Gray	Red	68 × 100	6,800	6.8 k

Resistors come in standard values. Novice designers specify parts that do not exist; the experienced engineer knows that, for instance, there's no such thing as a 1.9 k resistor. Engineering is a very practical art; one important trait of the good designer is using standard and easily available parts.

1.4.1.3 Circuits

Electricity always flows in a loop. A battery left disconnected discharges only very slowly because there's no loop, no connection of any sort (other than the nonzero resistance of humid air) between the two terminals. To make a lamp light, connect one lead to each battery terminal; electrons can now run in a loop from the battery's negative terminal, through the lamp, and back into the battery.

There are only two types of circuits: series and parallel. All real designs use combinations of these. A *series circuit* connects loads in a circular string; current flows around through each load in sequence (Figure 1.11). In a series circuit, the current is the same in every load.

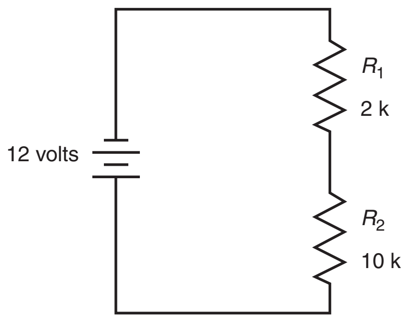


Figure 1.11: In a series circuit, the electrons flow through one load and then into another. The current in each resistor is the same; the voltage dropped across each depends on the resistor's value.

It's easy to calculate any parameter of a series circuit. In Figure 1.11, a 12-volt battery powers two series resistors. Ohm's Law tells us that the current flowing through the circuit is the voltage (12 in this case) divided by the resistance (the sum of the two resistors, or 12 k).

Total current is thus:

$$I = V \div R = (12\ volts) \div (2000 + 10{,}000\ ohms) = 12 \div 12000 = 0.001\ amp = 1\ mA$$

(Remember that *mA* is the abbreviation for milliamps.)

So what's the voltage across either of the resistors? In a series circuit, the current is identical in all loads, but the voltage developed across each load is a function of the load's resistance and the current. Again, Ohm's Law holds the secret. The voltage across R_1 is the current in the resistor times its resistance, or:

$$V_{R_1} = I_{R_1} = 0.001\ amps \times 2000\ ohms = 2\ volts$$

Since the battery places 12 volts across the entire resistor string, the voltage dropped on R_2 must be 12 – 2, or 10 volts. Don't believe that? Use Mr. Ohm's wonderful equation on R_2 to find:

$$V_{R_2} = I_{R_2} = 0.001\ amps \times 10{,}000\ ohms = 10\ volts$$

It's easy to extend this to any number of parts wired in series.

Parallel circuits have components wired so both pins connect (Figure 1.12). Current flows through both parts, though the amount of current depends on the resistance of each leg of the circuit. The voltage on each component, though, is identical.

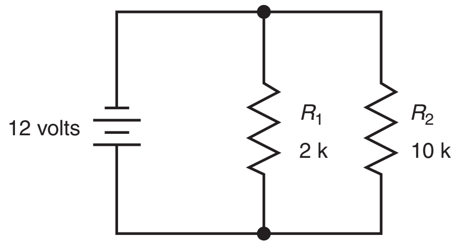


Figure 1.12: R_1 and R_2 are in parallel, both driven by the 12-volt battery.

We can compute the current in each leg much as we did for the series circuit. In the preceding case, the battery applies 12 volts to both resistors. The current through R_1 is:

$$I_{R_1} = 12\ volts \div 2{,}000\ ohms = 12 \div 2000 = 0.006\ amps = 6\ mA$$

Through R_2:

$$I_{R_2} = 12\ volts \div 10{,}000\ ohms = 0.0012\ amps = 1.2\ mA$$

Real circuits are usually a combination of series and parallel elements (Figure 1.13). Even in these more complex, more realistic cases, it's still very simple to compute anything one wants to know.

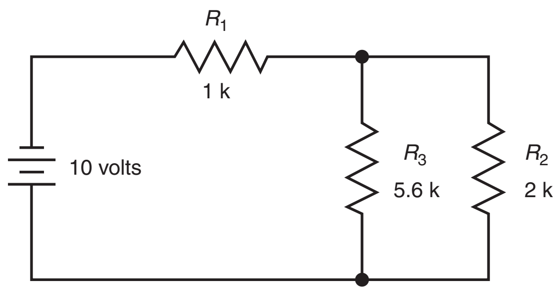


Figure 1.13: A series/parallel circuit.

Let's analyze the circuit shown in Figure 1.13. There's only one trick: cleverly combine complicated elements into simpler ones. Let's start by figuring the current flowing out of the battery. It's much too hard to do this calculation until we remember that two resistors in parallel look like a single resistor with a lower value.

Start by figuring the current flowing out of the battery and through R_1. We can turn this into a series circuit (in which the current flowing is the same through all the components) by replacing R_3 and R_2 with a single resistor with the same effective value as these two paralleled components. That's:

$$R_{\text{EFF}} = \frac{R_2 \times R_3}{R_1 + R_3} = \frac{5600 \times 2000}{5600 + 2000} = 1474 \ ohms$$

So the circuit is identical to one with two series resistors: R_1, still 1 k, and R_{EFF} at 1474 ohms. Ohm's Law gives the current flowing out of the battery and through these two resistors:

$$i = \frac{V}{R_1 + R_{\text{EFF}}} = \frac{10}{1000 + 1474} = 0.004 \ amps = 4 \ mA$$

Ohm's Law remains the font of all wisdom in basic circuit analysis and readily tells us the voltage dropped across R_1:

$$V = iR_1 = 0.004 \ amps \times 1000 \ ohms = 4 \ volts$$

Clearly, since the battery provides 10 volts, the voltage across the paralleled pair R_2 and R_3 is 6 volts.

1.4.1.4 Power

Power is the product of voltage and current and is expressed in *watts*. One watt is one volt times one amp. A *milliwatt* is a thousandth of a watt; a *microwatt* is a millionth.

You can think of power as the total amount of electricity present. A thousand volts sounds like a lot of electricity, but if there's only a microamp available, that's a paltry milliwatt—not much power at all.

Power is also current^2 times resistance:

$$\text{P} = I^2 \times R$$

Electronic components like resistors and ICs consume a certain amount of volts and amps. An IC doesn't move, make noise, or otherwise release energy (other than exerting a minimal amount of energy in sending signals to other connected devices), so almost all the energy consumed gets converted to heat. All components have maximum power dissipation ratings; exceed these at your peril.

If a part feels warm it's dissipating a reasonable fraction of a watt. If it's hot but you can keep your finger on it, then it's probably operating within specs, though many analog components want to run cooler. If you pull back, not burned, but the heat is too much for your finger, then in most cases (be wary of the wimp factor; some people are more heat sensitive than others) the device is too hot and either needs external cooling (heat sink, fan, etc.), has failed, or your circuit exceeds heat parameters. A burn or near burn or discoloration of the device means there's trouble brewing in all but exceptional conditions (e.g., high-energy parts like power resistors).

A PC's processor has so many transistors, each losing a bit of heat, that the entire part might consume and eliminate 100+ watts. That's far more than the power required to destroy the chip. Designers expend a huge effort in building heat sinks and fans to transfer the energy in the part to the air.

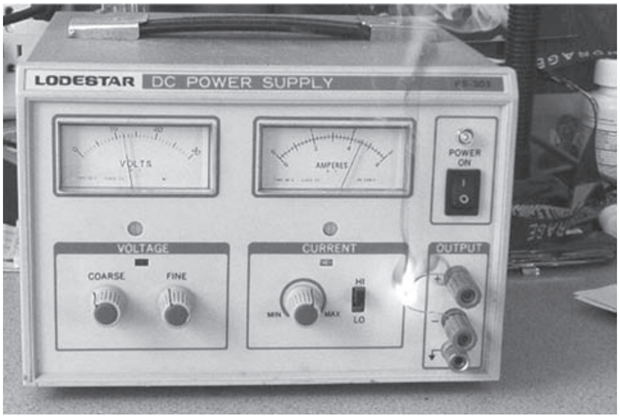


Figure 1.14: This 10-ohm resistor, with 12 volts applied, draws 833 mA. P = I^2R, so it's sucking about 7 watts. Unfortunately, this particular part is rated for ¼ watt max, so it is on fire. Few recent college grads have a visceral feel for current, power, and heat, so this demo makes their eyes go like saucers.

The role of heat sinks and fans is to remove the heat from the circuits and dump it into the air before the devices burn up. The fact that a part dissipates a lot of energy and wants to run hot is not bad as long as proper thermal design removes the energy from the device before it exceeds its max temp rating (Figure 1.14).

1.4.2 AC Circuits

AC is short for *alternating current*, which is any signal that's not DC. AC signals vary with time. The mains in your house supply AC electricity in the shape of a sine wave: the voltage varies from a large negative to a large positive voltage 60 times per second (in the United States and Japan) or 50 times per second (in most of the rest of the world).

AC signals can be either *periodic*, which means they endlessly and boringly repeat forever, or *aperiodic*, the opposite. Static from your FM radio is largely aperiodic since it's quite random. The bit stream on any address or data line from a micro is mostly aperiodic, at least over short times, as it's a complex changing pattern driven by the software.

The rate at which a periodic AC signal varies is called its *frequency*, which is measured in *hertz* (Hz for short). One Hz means the waveform repeats once per second. A thousand Hz is a kHz (kilohertz), a million Hz is the famous MHz by which so many microprocessor clock rates are defined, and a billion Hz is a GHz.

The reciprocal of Hz is *period*. That is, where the frequency in hertz defines the signal's repetition rate, the period is the time it takes for the signal to go through a cycle.

Mathematically:

$$\text{Period in seconds} = 1 \div \text{frequency in Hz}$$

Thus, a processor running at 1 GHz has a clock period of 1 nanosecond—one billionth of a second. No kidding. In that brief flash of time, even light goes but a bare foot. Though your 1.8 GHz PC may seem slow loading Word, it's cranking instructions at a mind-boggling rate.

Wavelength relates a signal's period—and thus its frequency—to a physical "size." It's the distance between repeating elements and is given by:

$$\text{Wavelength in meters} = \frac{c}{\mathit{frequency}} = \frac{300{,}000{,}000\ \mathit{meters/second}}{\mathit{frequency\ in\ Hz}}$$

where c is the speed of light.

An FM radio station at about 100 MHz has a wavelength of 3 meters. AM signals, on the other hand, are around 1 MHz, so each sine wave is 300 meters long. A 2.4-GHz cordless phone runs at a wavelength a bit over 10 cm.

As the frequency of an AC signal increases, things get weird. The basic ideas of DC circuits still hold but need to be extended considerably. Just as relativity builds on Newtonian mechanics to describe fast-moving systems, electronics needs new concepts to properly describe fast AC circuits.

Resistance, in particular, is really a subset of the real nature of electronic circuits. It turns out that there are three basic kinds of resistive components; each behaves somewhat differently. We've already looked at resistors; the other two components are capacitors and inductors.

Both of these parts exhibit a kind of resistance that varies depending on the frequency of the applied signal; the amount of this "AC resistance" is called *reactance*.

1.4.2.1 Capacitors

A *capacitor*, colloquially called the "cap," is essentially two metal plates separated from each other by a thin insulating material. This insulation, of course, means that a DC signal cannot flow through the cap. It's like an open circuit.

But in the AC world, strange things happen. It turns out that AC signals can make it across the gap between the two plates; as the frequency increases, the effective resistance of this gap decreases. This resistive effect is called *reactance*; for a capacitor it's termed *capacitive reactance* (Figure 1.15). There's a formula for everything in electronics; for capacitive reactance it's:

$$X_c = \frac{1}{2\pi fc}$$

where:

X_c = capacitive reactance
f = frequency in Hz
c = capacitance in farads

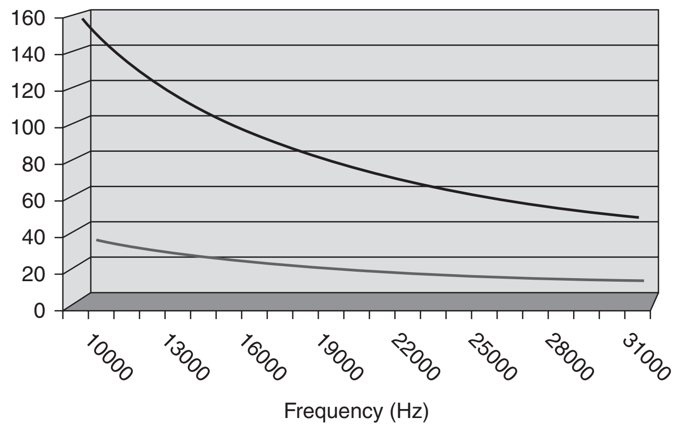


Figure 1.15: Capacitive reactance of a 0.1 μF cap (top) and a 0.5 μF cap (bottom curve). The vertical axis is reactance in ohms. See how larger caps have lower reactances, and as the frequency increases reactance decreases. In other words, a bigger cap passes AC better than a smaller one, and at higher frequencies all caps pass more AC current. Not shown: at 0 Hz (DC), reactance of all caps is essentially infinite.

Capacitors thus pass only *changing* signals (Table 1.6). The current flowing through a cap is:

$$I = \frac{dV}{dt}$$

(If your calculus is rusty or nonexistent, this simply means that the current flow is proportional to the change in voltage over time.)

In other words, the faster the signal changes, the more current flows.

Table 1.6: Range of values for real-world capacitors.

Name	Abbreviation	Farads	Where Likely Found
picofarad	pF	10^{-12} farad	Padding caps on microprocessor crystals, oscillators, analog feedback loops.
microfarad	μF	10^{-6} farad	Decoupling caps on chips are about .01 to .1 μF; low-freq decoupling runs about 10 μF, big power supply caps might be 1000 μF.
farad	F	1 farad	One farad is a huge capacitor and generally does not exist. A few vendors sell "supercaps" that have values up to a few farads, but these are unusual. Sometimes used to supply backup power to RAM when the system is turned off.

In real life there's no such thing as a perfect capacitor. All leak a certain amount of DC and exhibit other more complex behavior. For that reason, there's quite a range of different types of parts.

In most embedded systems you'll see one of two types of capacitors (Figure 1.16). The first are the polarized ones, devices which have a plus and a minus terminal. Connect one backward and the part will likely explode!

Polarized devices have large capacitance values: tens to thousands of microfarads. They're most often used in power supplies to remove the AC component from filtered signals. Consider the equation of capacitive reactance: large cap values pass lower-frequency signals efficiently. Typical construction today is from a material called *tantalum*; seasoned EEs often call these devices *tantalums*. You'll see tantalum caps on PC boards to provide a bit of bulk storage of the power supply.

Smaller caps are made from a variety of materials. These have values from a few picofarads to a fraction of a microfarad. They're often used to "decouple" the power supply on a PCB (i.e., to short high-frequency switching from power to ground, so the logic signals don't get coupled into the power supply). Most PCBs have dozens or hundreds of these parts scattered around.

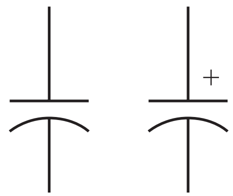


Figure 1.16: Schematic symbols for capacitors. The one on the left is a generic, generally low-valued (under 1 μF) part. On the right the plus sign shows that the cap is polarized. Installed backward, it's likely to explode.

We can wire capacitors in series and in parallel; compute the total effective capacitance using the rules opposite those for resistors. So, for two caps in parallel, sum their values to get the effective capacitance. In a series configuration the total effective capacitance is:

$$C_{eff} = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \cdots}$$

Note that this rule is for figuring the total capacitance of the circuit, *not* for computing the total reactance. More on that shortly.

One useful characteristic of a capacitor is that it can store a charge. Connect one to a battery or power supply and it will store that voltage. Remove the battery and (for a perfect, lossless part) the capacitor will still hold that voltage. Real parts leak a bit; ones rated at under 1 μF or so discharge rapidly. Larger parts store the charge longer.

Interesting things happen when you wire a cap and a resistor in series. The resistor limits current to the capacitor, causing it to charge slowly. Suppose the circuit shown in Figure 1.17 is dead, no voltage at all applied. Now turn on the switch. Though we've applied a DC signal, the sudden transition from 0 to 5 volts is AC.

Current flows due to the $I = \frac{dV}{dt}$ rule; *dV* is the sudden edge from flipping the switch. But the input goes from an AC-edge to steady-state DC, so current stops flowing pretty quickly. How fast? That's defined by the circuit's *time constant*.

A resistor and capacitor in series is colloquially called an *RC circuit*. The graph shows how the voltage across the capacitor increases over time. The time constant of any circuit is pretty well approximated by:

$$t = RC$$

for *R* in ohms, *C* in farads, and *t* in seconds.

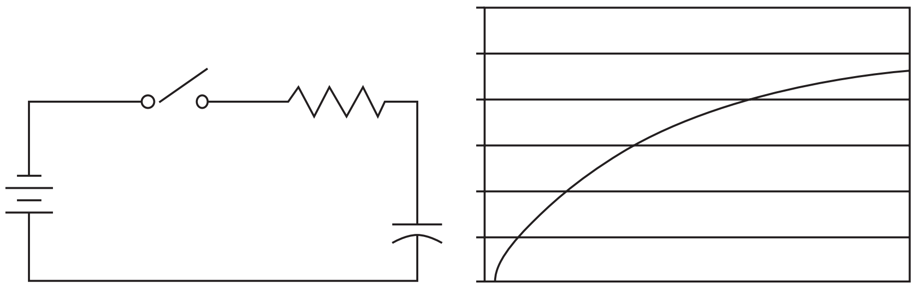


Figure 1.17: Close the switch and the voltage applied to the RC circuit looks like the top curve. The lower graph shows how the capacitor's voltage builds slowly with time, headed asymptotically toward the upper curve.

This formula tells us that after *RC* seconds the capacitor will be charged to 63.2% of the battery's voltage. After another *RC* seconds, another 63.2%, for a total of 86.5%.

Analog circuits use a lot of RC circuits; in a microprocessor it's still common to see them controlling the CPU's reset input. Apply power to the system and all the logic comes up, but the RC's time constant keeps reset asserted low for a while, giving the processor time to initialize itself.

The most common use of capacitors in the digital portion of an embedded system is to *decouple* the logic chips' power pins. A medium value part (0.01 to 0.1 μF) is tied between power and ground very close to the power leads on nearly every digital chip. The goal is to keep power supplied to the chips as clean as possible—close to a perfect DC signal.

Why would this be an issue? After all, the system's power supply provides a nearly perfect DC level. It turns out that as a fast logic chip switches between zero and one it can draw immense amounts of power for a short, subnanosecond, time. The power supply cannot respond quickly enough to regulate that, and since there's some resistance and reactance between the supply and the chip's pins, what the supply provides and what the chip sees are somewhat different. The decoupling capacitor shorts this very high-frequency (i.e., short transient) signal on Vcc to ground. It also provides a tiny bit of localized power storage that helps overcome the instantaneous voltage drop between the power supply and the chip.

Most designs also include a few tantalum bulk storage devices scattered around the PC board, also connected between Vcc and ground. Typically these are 10 to 50 μF each. They are even more effective bulk storage parts to help minimize the voltage drop chips would otherwise see.

You'll often see very small caps (on the order of 20 pF) connected to microprocessor drive crystals. These help the device oscillate reliably.

Analog circuits make many wonderful and complex uses of caps. It's easy to build integrators and differentiators from these parts, as well as analog hold circuits that memorize a signal for a short period of time. Common values in these sorts of applications range from 100 pF to fractions of a microfarad.

1.4.2.2 Inductors

An *inductor* is, in a sense, the opposite of a capacitor. Caps block DC but offer diminishing resistance (really, reactance) to AC signals as the frequency increases. An inductor, on the other hand, passes DC with zero resistance (for an idealized part), but the resistance (reactance) increases proportionately to the frequency.

Physically an inductor is a coil of wire and is often referred to as a *coil*. A simple straight wire exhibits essentially no inductance. Wrap a wire in a loop and it's less friendly to AC signals. Add more loops, or make them smaller, or put a bit of ferrous metal in the loop, and inductance increases. Electromagnets are inductors, as is the field winding in an alternator or motor.

An *iron core* inductor is wound around a slug of metal, which increases the device's inductance substantially (Figure 1.18).

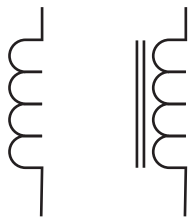


Figure 1.18: Schematic symbols of two inductors. The one on the left is an "air core"; the one on the right is an "iron core."

Inductance is measured in henries (H). *Inductive reactance* is the tendency of an inductor to block AC and is given by:

$$X_L = 2\pi Lf$$

where:

X_L = Inductive reactance
f = frequency in Hz
L = inductance in henries

Clearly, as the frequency goes to zero (DC), reactance does as well.

Inductors follow the resistor rules for parallel and series combinations: add the value (in henries) when in series, and use the division rule when in parallel.

Inductors are much less common in embedded systems than are capacitors, yet they are occasionally important. The most common use is in switching power supplies. Many datacomm circuits use small inductors (generally millihenries) to match the network being driven.

Power supplies usually have a *transformer* which reduces the AC mains (from the wall) to a lower voltage more appropriate for embedded systems (Figure 1.19).

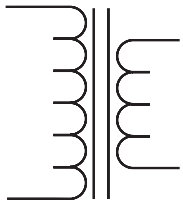


Figure 1.19: The schematic symbol for a transformer.

Transformers are two inductors wrapped around each other, with an iron core. The input AC generates a changing magnetic field, which induces a voltage in the output ("secondary") inductor.

If both inductors have the same number of wire loops, the output voltage is the same as the input. If the secondary has fewer loops, the voltage is less.

Sometimes signals, especially those flowing off a PC board, will have a *ferrite bead* wrapped around the wire. These beads are small cylinders (a few mm long) made of a ferromagnetic material. Like all inductors, they help block AC so are used to minimize noise of signal wires.

1.4.3 Active Devices

Resistors, capacitors and inductors are the basic *passive* components, passive meaning "dumb." The parts can't amplify or dramatically change applied signals. By contrast, *active* parts can clip, amplify, distort, and otherwise change an applied signal. The earliest active parts were vacuum tubes, called "valves" in the UK.

Consider the schematic in Figure 1.20, which is a single tube that contains two identical active elements, each called a *triode*, as each has three terminals. Tubes are easy to understand; let's see how one works.

A *filament* heats the cathode, which emits a stream of electrons. They flow through the grid, a wire mesh, and are attracted to the plate. Electrons are negatively charged, so applying a very small amount of positive voltage to the grid greatly reduces their flow. This is the basis of amplification: a small control signal greatly affects the device's output.

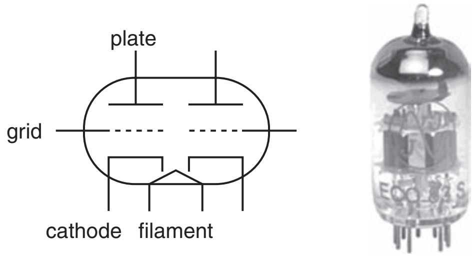


Figure 1.20: On the left, a schematic of a dual triode vacuum tube. The part itself is shown on the right.

Of course, in the real world tubes are almost unheard of today. When Bardeen, Brattain, and Shockley invented the *transistor* in 1947 they started a revolution that continues today. Tubes are power hogs, bulky and fragile. *Transistors*—also three-terminal devices that amplify—seem to have no lower limit of size and can run on picowatts (Figure 1.21).

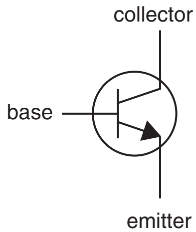


Figure 1.21: The schematic diagram of a bipolar NPN transistor with labeled terminals.

A transistor is made from a single crystal, normally of silicon, into which impurities are doped to change the nature of the material. The tube description showed how it's a voltage-controlled device; bipolar transistors are current-controlled.

Writers love to describe transistor operation by analogy to water flow or to the movement of holes and carriers within the silicon crystal. These are at best poor attempts to describe the quantum mechanics involved. Suffice to say that, in Figure 1.21, feeding current into the base allows current to flow between the collector and emitter.

And that's about all you need to know to get a sense of how a transistor amplifier works. The circuit shown in Figure 1.22 is a trivialized example of one. A microphone—which has a tiny output—drives current into the base of the transistor, which amplifies the signal, causing the lamp to fluctuate in rhythm with the speaker's voice.

A real amplifier might have many cascaded stages, each using a transistor to get a bit of amplification. A radio, for instance, might have to increase the antenna's signal by many millions before it gets to the speakers.

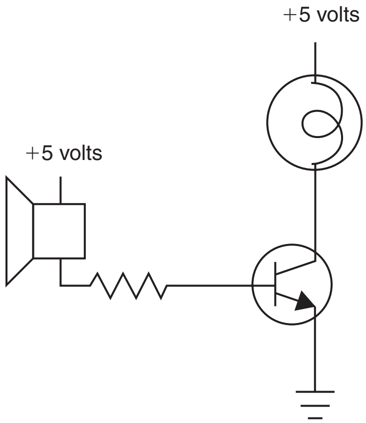


Figure 1.22: A very simple amplifier.

Transistors are also switches, the basic element of digital circuits. The previous circuit is a simplified—but totally practical—NOR gate (Figure 1.23). When both inputs are zero, both transistors are off. No current flows from their collectors to emitters, so the output is 5 volts (as supplied by the resistor).

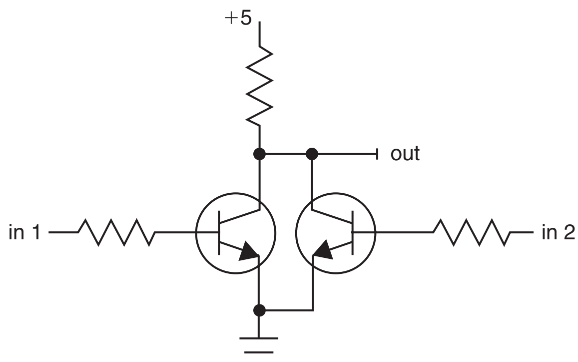


Figure 1.23: A NOR gate circuit.

If either input goes to a high level, the associated transistor turns on. This causes a conduction path through the transistor, pulling "out" low. In other words, any input going to a one gives an output of zero. Table 1.7 illustrates the circuit's behavior.

It's equally easy to implement any logic function.

The circuit we just analyzed would work; in the 1960s all "RTL" integrated circuits used exactly this design. But the gain of this approach is very low. If the input dawdles between a zero and a one, so will the output. Modern logic circuits use very high amplification factors,

Table 1.7: Truth table.

in1	in2	out
0	0	1
0	1	0
1	0	0
1	1	0

so the output is either a legal zero or one, not some in-between state, no matter what input is applied.

The silicon is a conductor, but a rather lousy one compared to a copper wire. The resistance of the device between the collector and the emitter changes as a function of the input voltage; for this reason active silicon components are called *semiconductors*.

Transistors come in many flavors; the one we just looked at is a bipolar part, characterized by high power consumption but (typically) high speeds. Modern ICs are constructed from MOSFET—Metal Oxide Semiconductor Field Effect Transistor—devices, or variants thereof (Figure 1.24). A mouthful? You bet. Most folks call these transistors FETs for short.

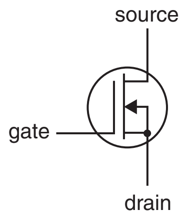


Figure 1.24: The schematic diagram of a MOSFET.

A FET is a strange and wonderful beast. The gate is insulated by a layer of oxide from a silicon channel running between the drain and source. No current flows from the gate to the silicon channel. Yet putting a bias voltage (like a tube, a FET is a voltage device) on the gate creates an electrostatic field that reduces current flow between the other two terminals. Again, *no current flows from the gate*. And when turned on, the source-drain resistance is much lower than in a bipolar transistor. This means the part dissipates little power, a critical concern when putting millions of these transistors on a single IC.

A *diode* is a two-terminal semiconductor that passes current in one direction only. In Figure 1.25, a positive voltage will flow from the left to the right, but not in the reverse

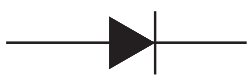


Figure 1.25: The schematic symbol for a diode.

direction. This seems a little thing, but it's incredibly useful. Figure 1.26 shows a circuit that implements an OR gate without a transistor.

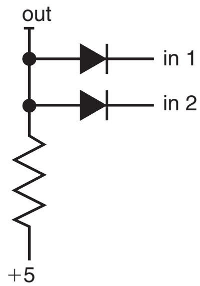


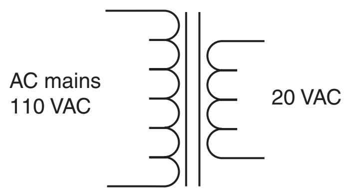
Figure 1.26: A diode OR circuit.

If both inputs are logic one, the output is a one (pulled up to +5 by the resistor). Any input going low will drag the output low as well. Yet the diodes ensure that a low-going input doesn't drag the other input down.

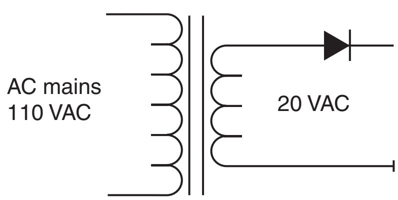
1.5 Putting It Together: A Power Supply

A power supply is a simple yet common circuit that uses many of the components we've discussed. The input is 110 volts AC (or 220 volts in Europe, 100 in Japan, 240 in the UK). Output might be 5 volts DC for logic circuits. How do we get from high voltage AC input to 5 volts DC?

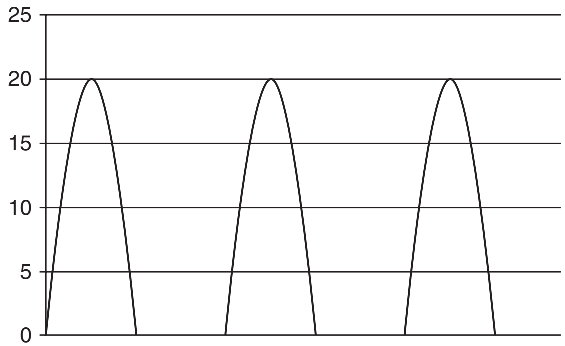
The first step is to convert the AC mains to a lower voltage AC, as follows:



Now let's turn that lower voltage AC into DC. A diode does the trick nicely:

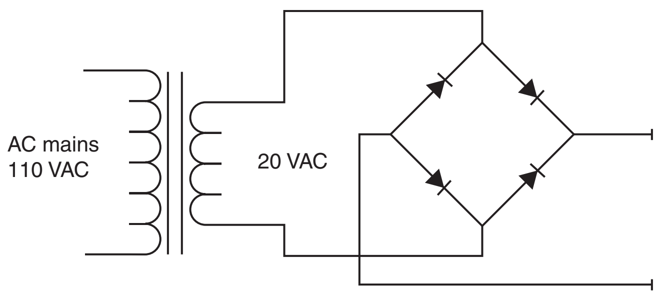


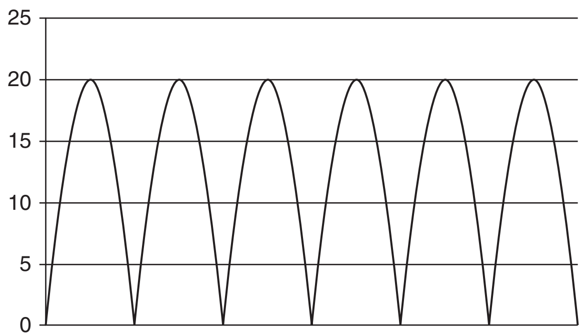
The AC mains are a sine wave, of course. Since the diode conducts in one direction only, its output looks like:



This isn't DC … but the diode has removed all the negative-going parts of the waveform.

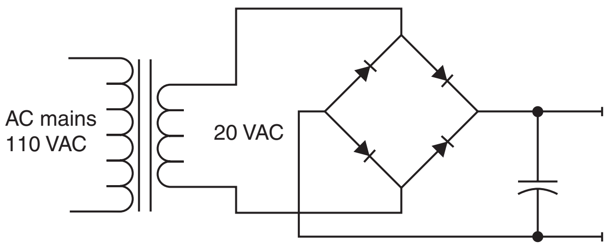
But we've thrown away half the signal; it's wasted. A better circuit uses four diodes arranged in a *bridge* configuration as follows:





The bridge configuration ensures that two diodes conduct on each half of the AC input, as shown above. It's more efficient and has the added benefit of doubling the apparent frequency, which will be important when we're figuring out how to turn this moving signal into a DC level.

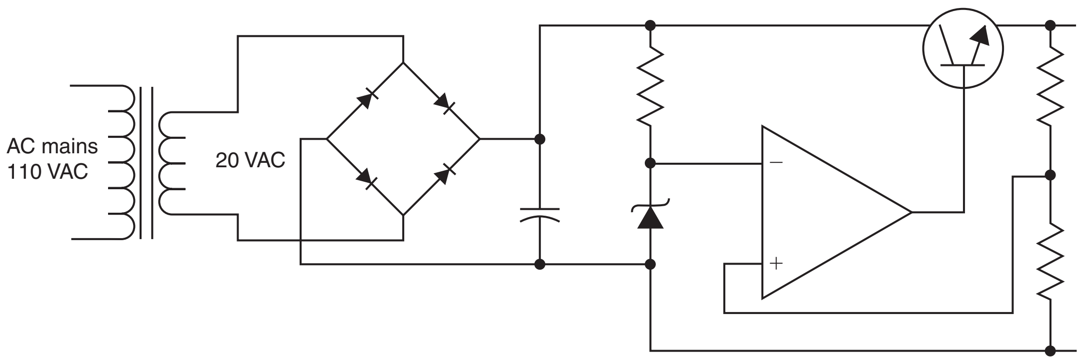
The average of this signal is clearly a positive voltage; if only we had a way to create an average value. Turns out that a capacitor does just that:



A huge-value capacitor filters best—typical values are in the thousands of microfarads.

The output is a pretty decent DC wave, but we're not done yet. The load—the device this circuit will power—will draw varying amounts of current. The diodes and transformer both have resistance. If the load increases, current flow goes up, so the drop across the parts will increase (Ohm's Law tells us $E = IR$, and as I goes up, so does E). Logic circuits are very sensitive to fluctuations in their power, so some form of *regulation* is needed.

A regulator takes varying DC in and produces a constant DC level out. For example:



The odd-looking part in the middle is a *zener diode*. The voltage drop across the zener is always constant, so if, for example, this is a 3-volt part, the intersection of the diode and the resistor will *always* be 3 volts.

The regulator's operation is straightforward. The zener's output is a constant voltage. The triangle is a bit of magic—an error amplifier circuit—that compares the zener's constant voltage to the output of the power supply (at the node formed by the two resistors). If the output voltage goes up, the error amplifier applies less bias to the base of the transistor, making it conduct less … and lowering the supply's output. The transistor is key to the circuit; it's sort of like a variable resistor controlled by the error amp.

If, say, 20 volts of unregulated DC go into the transistor from the bridge and capacitor, and the supply delivers 5 volts to the logic, there's 15 volts dropped across the transistor. If the supply provides even just two amps of current, that's 30 watts (15 volts times two amps) dissipated by that semiconductor—a lot of heat! Careful heatsinking will keep the device from burning up.

1.5.1 The Scope

The oscilloscope (colloquially known as the "scope") is the most basic tool used for trouble-shooting and understanding electronic circuits. Without some understanding of this most critical of all tools, you'll be like a blind person trying to understand color.

The scope has only one function: it displays a graph of the signal or signals you're probing (Figure 1.27). The horizontal axis is usually time; the vertical is amplitude, a fancy electronics term for voltage.

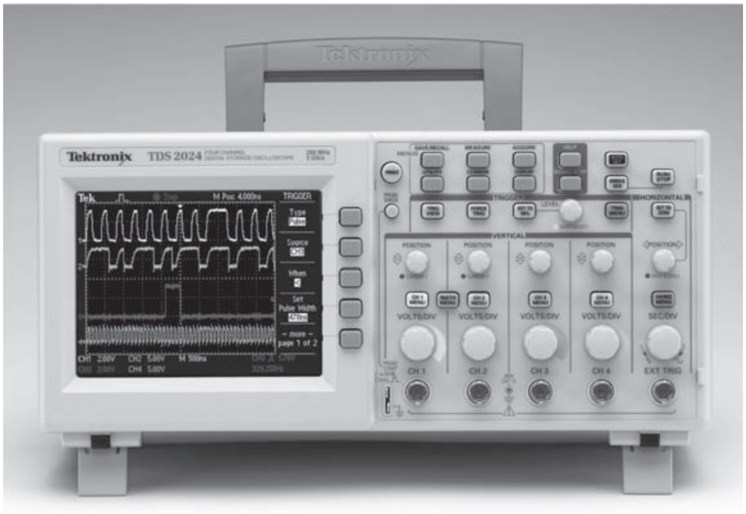


Figure 1.27: A sea of knobs. Don't be intimidated. There's a logical grouping to these. Master them and wow your friends and family. Photo courtesy of Tektronix, Inc.

1.5.2 Controls

In Figure 1.28, note first the two groups of controls labeled "vertical input 1" and "vertical input 2." This is a two-channel scope, by far the most common kind, which allows you to sample and display two different signals at the same time.

The vertical controls are simple. "Position" allows you to move the graphed signal up and down on the screen to the most convenient viewing position. When you're looking at two signals it allows you to separate them, so they don't overlap confusingly.

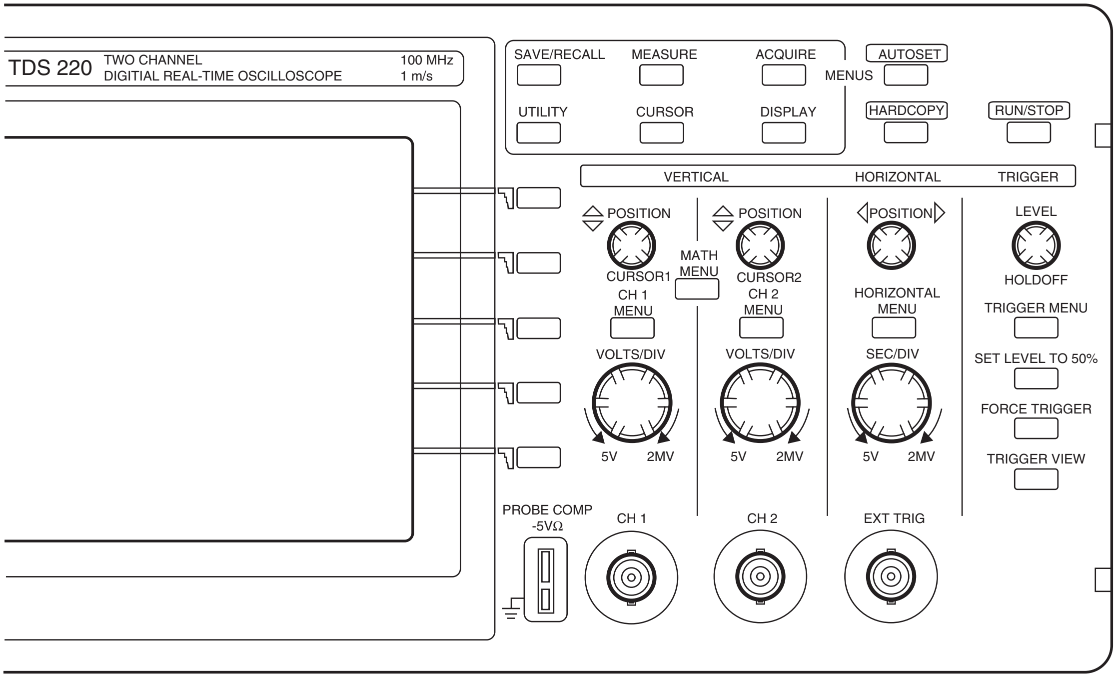


Figure 1.28: Typical oscilloscope front panel. Picture courtesy Tektronix, Inc.

"Volts/div" is short for volts-per-division. You'll note the screen is a matrix of 1 cm by 1 cm boxes; each is a "division." If the "volts/div" control is set to 2, then a two-volt signal extends over a single division. A five-volt signal will use 2.5 divisions. Set this control so the signal is easy to see. A reasonable setting for TTL (5-volt) logic is 2 volts/div.

The "coupling" control selects "DC"—which means what you see is what you get. That is, the signal goes unmolested into the scope. "AC" feeds the input through a capacitor; since caps cannot pass DC signals, this essentially subtracts DC bias (Figure 1.29).

The "mode" control lets us look at the signal on either channel, or both simultaneously.

Now check out the horizontal controls. These handle the scope's "time base," so called because the horizontal axis is always the time axis.

The "position" control moves the trace left and right, analogously to the vertical channel's knob of the same name.

"Time/div" sets the horizontal axis' scale. If set to 20 nsec/div, for example, each cm on the screen corresponds to 20 nsec of time. Figure 1.30 shows the same signal displayed using two different time base settings; it's more compressed in the left picture simply because at 2000 μsec/div more pulses occur in the 1 cm division mark.

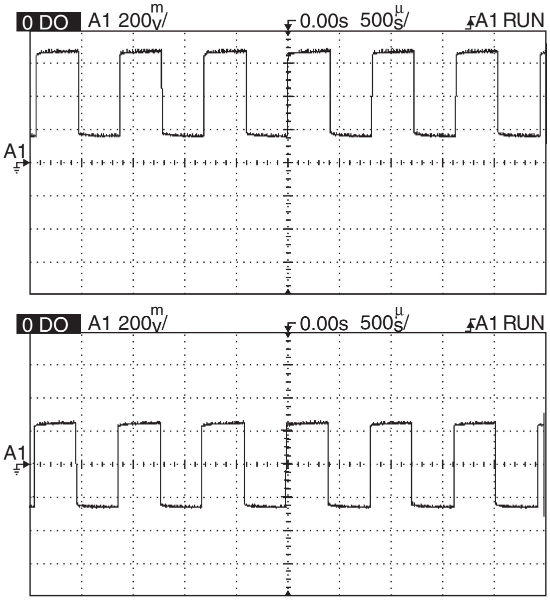


Figure 1.29: The signal is an AC waveform riding on top of a constant DC signal. On the left we're observing it with the scope set to DC coupling; note how the AC component is moved up by the amount of DC (in other words, the total signal is the DC component + the AC). On the right we've changed the coupling control to "AC"; the DC bias is removed and the AC component of the signal rides in the middle of the screen.

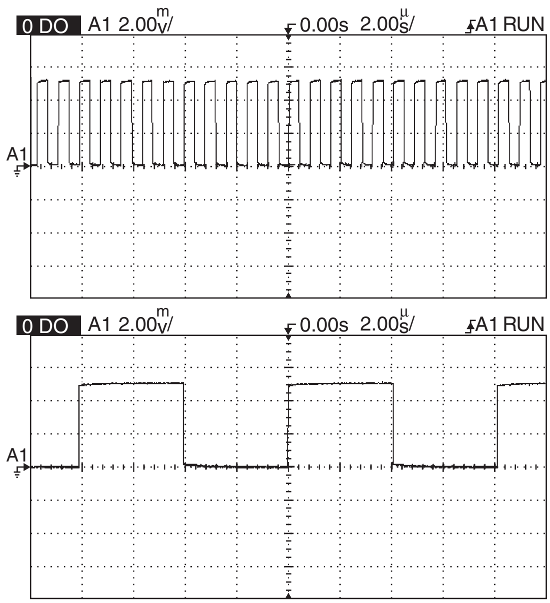


Figure 1.30: The left picture shows a signal with the time base set to 2000 µsec/division; the right is the same signal, but now we're sweeping at 200 µsec/division. Though the data is unchanged, the signal looks compressed. Also note that the 5-volt signal extends over 2.5 vertical boxes, since the gain is set to 2 volts/div. The first rule of scoping is to know the horizontal and vertical settings.

The last bank of knobs—those labeled "trigger"—are perhaps the most important of all. Though you see a line on the screen, it's formed by a dot swept across from left to right, repeatedly, at a very high speed. How fast? The dot moves at the speed you've set in the time/div knob. At 1 sec/div the dot takes 10 seconds to traverse the normal 10 cm-wide scope screen. More usual speeds for digital work are in the few microseconds to nanosecond range, so the dot moves faster than any eye can track.

Most of the signals we examine are more or less repetitive: it's pretty much the same old waveform over and over again. The trigger controls tell the scope when to start sweeping the dot across the screen. The alternative—if the dot started on the left side at a random time—would result in a very quickly scrolling screen, which no one could follow.

Twiddling the "trigger level" control sets the voltage at which the dot starts its inexorable left-to-right sweep. Set it to 6 volts and the normal 5-volt logic signal will never get high enough that the dot starts. The screen stays blank. Crank it to zero and the dot runs continuously, unsynchronized to the signal, creating a scrambled mess on the scope screen.

Set trigger level to 2 volts or so, and as the digital signal traverses from 0 to 5 volts the dot starts scanning, synchronizing to the signal.

It's most dramatic to learn how this control works when you're sampling a sine wave. As you twirl the knob clockwise (from a low trigger voltage to a higher one) the displayed sine wave shifts to the left. That is, the scan starts later and later since the triggering circuit waits for an ever-increasing signal voltage before starting.

"Trigger Menu" calls up a number of trigger selection criteria. Select "trigger on positive edge" and the scope starts sweeping when the signal goes from a low level through the trigger voltage set with the "Trigger Level" knob. "Trigger on negative edge" starts the sweep when the signal falls from a high level through the level.

Every scope today has more features than normal humans can possibly remember, let alone use. Various on-screen menus let you do math on the inputs (add them and so on), store signals that occur once, and much, much more. The instrument is just like a new PC application. Sure, it's nice to read the manual, but don't be afraid to punch a lot of buttons and see what happens. Most functions are pretty intuitive.

1.5.3 Probes

A "probe" connects the scope to your system. Experienced engineers' fingers are permanently bent a bit, warped from too many years holding the scope probe in hand while working on circuit boards. Though electrically the probe is just a wire, in fact there's a bit of electronics magic inside to propagate signals without distortion from your target system to the scope.

So too for any piece of test equipment. The tip of the scope probe is but one of the two connections required between the scope and your target system. A return path is needed, a ground (Figure 1.31). If there's no ground connection the screen will be nuts, a swirling mass of meaningless scrolling waveforms.

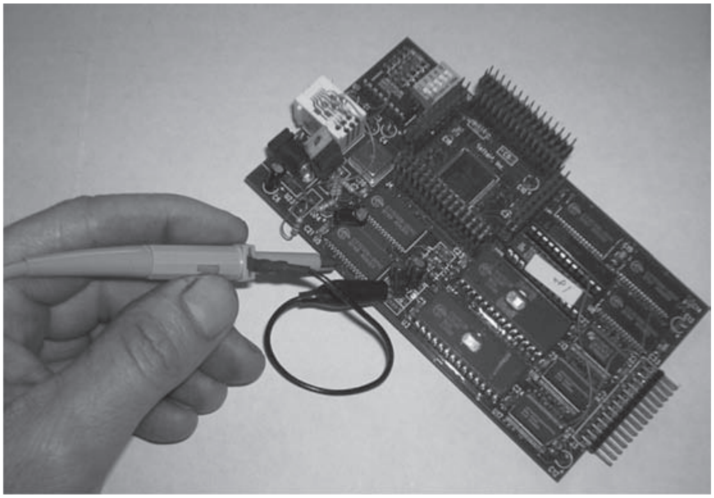


Figure 1.31: Always connect the probe's ground lead to the system.

Yet often we'll see engineers probing nonchalantly without an apparent ground connection. Oddly, the waves look fine on the scope. What gives? Where's the return path?

It's in the lab wall. Most electric cords, including the one to the scope and possibly to your target system, have three wires. One is ground. It's pretty common to find the target grounded to the scope via this third wire, going through the wall outlets. Of one thing be sure: even if this ground exists, it's ugly. It's a marginal connection at best, especially when dealing with high-speed logic signals or low level noise-sensitive analog inputs. Never, ever count on it even when all seems well. Every bit of gear in the lab, probably in the entire building, shares this ground. When the Xerox machine on the third floor kicks in, the big inductive spike from the motor starting up will distort the scope signal.

No scope will give decent readings on high-speed digital data unless it is *properly* grounded. I can't count the times technicians have pointed out a clock improperly biased 2 volts above ground, convinced they found the fault in a particular system, only to be bemused and embarrassed when a good scope ground showed the signal in its correct 0 to 5 volt glory. Ground the probe and thus the scope to your target using the little wire that emits from the end of the probe. As circuits get faster, shorten the wire. The very shortest ground lead results in the least signal distortion (see Figure 1.32.)

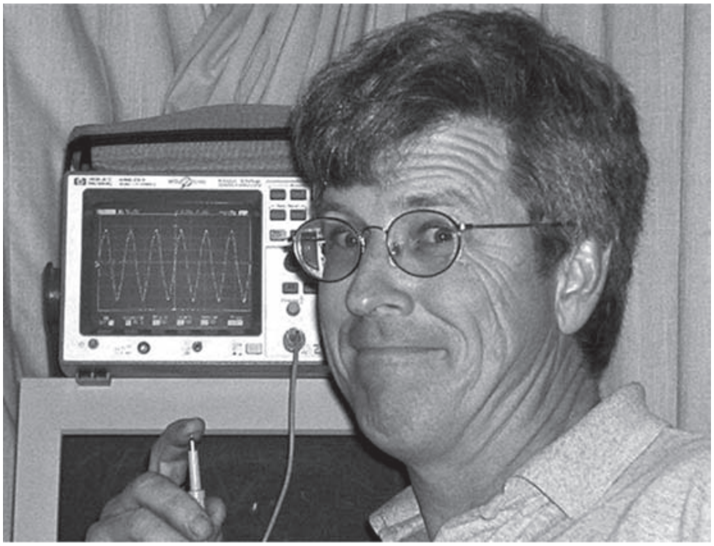


Figure 1.32: Here we probe a complex nonembedded circuit. Note the displayed waveform. A person is an antenna that picks up the 60 Hz hum radiated from the power lines in the walls around us. Some say engineers are particularly sensitive (though not their spouses).

Yet most scope probes come with crummy little lead alligator clips on the ground wire that are impossible to connect to an IC. The frustrated engineer might clip this to a clip lead that has a decent "grabber" end. Those extra 6–12 inches of ground may very well trash the display, showing a waveform that is not representative of reality. It's best to cut the alligator clip off the probe and solder a micrograbber on in its place.

One of the worst mistakes we make is neglecting probes. Crummy probes will turn that wonderful 1-GHz instrument into junk. After watching us hang expensive probes on the floor, mixed in with all sorts of other debris, few bosses are willing to fork over the $150 that Tektronix or Agilent demands. But the $50 alternatives are junk. Buy the best and take good care of them (see Figure 1.33.)

Figure 1.33: Tektronix introduced the 545 scope back in the dark ages; a half-century later, many are still going strong. Replace a tube from time to time and these might last forever. About the size of a two-drawer file cabinet and weighing almost 100 pounds, they're still favored by Luddites and analog designers.

Endnotes

[1.1] *Embedded Microcomputer Systems*, Valvano, p. 509.

[1.2] Net Silicon, "Net50BlockDiagram."

[1.3] *Beginner's Guide to Reading Schematics*, Traister and Lisk, p. 49.

[1.4] H. Malcolm, *Foundations of Computer Architecture.* Additional references include Stallings, W., *Computer Organization and Architecture*, Prentice Hall, fourth edition, 1995; Tanenbaum, A. S., *Structured Computer Organization*, Prentice Hall, third edition, 1990; Baron, R. J. and Higbie L., *Computer Architecture*, Addison-Wesley, 1992; Kane, G. and Heinrich, J., *MIPS RISC Architecture*, Prentice-Hall, 1992; and Patterson, D. A., and Hennessy, J. L., *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann, third edition, 2005.

[1.5] National Semiconductor, "Geode User Manual," Rev. 1, p. 13.

[1.6] Net Silicon, "Net+ARM40 Hardware Reference Guide," pp. 1–5.

[1.7] "EnCore M3 Embedded Processor Reference Manual," Revision A, p. 8.

[1.8] "EnCore PP1 Embedded Processor Reference Manual," Revision A, p. 9.