Chapter 1

The Digital Abstraction

1.1 The Digital Revolution

Digital systems are pervasive in modern society. Some uses of digital techonolgoy are obvious - such as a personal computer or a network switch. However, there are also many hidden applications of digital technology. When you speak on the phone, in almost all cases your voice is being digitized and transmitted via digital communications equipment. When you play a music CD, the music, recorded in digital form, is processed by digital logic to correct errors and improve the audio quality. When you watch TV, the image is processed by digital electronics to improve picture quality (and for HDTV the transmission is digital as well). If you have a TiVo (or other PVR) you are recording video in digital form. DVDs are compressed digital video recordings. When you play a DVD you are digitally decompressing and processing the video. Most radios - cell phones, wireless networks, etc... - use digital signal processing to implement their modems. The list goes on.

Most modern electronics uses analog circuitry only at the edge - to interface to a physical sensor or actuator. As quickly as possible, signals from a sensor (e.g., a microphone) are converted into digital form and all real processing, storage, and transmission of information is done in digital form. The signals are converted back to analog form only at the output - to drive an actuator (e.g., a speaker).

Not so long ago the world was not so digital. In the 1960s digital logic was found only in expensive computer systems and a few other niche applications. All TVs, radios, music recordings, and telephones were analog.

The shift to digital was enabled by the scaling of integrated circuits. As integrated circuits become more complex, more sophisticated signal processing became possible. This signal processing was only possible using digital logic. The complexity of the modulation, error correction, compression, and other techniques were not feasible in analog technology. Only digital logic with its ability to perform a complex computation without accumulating noise and its

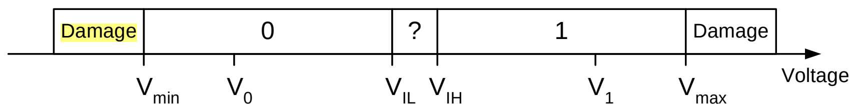

Figure 1.1: Encoding of two symbols, 0 and 1, into voltage ranges. Any voltage in the range labeled 0 is considered a 0 symbol. Any voltage in the range labeled 1 is considerd to be a 1 symbol. Voltages between the 0 and 1 ranges (the ? range) are undefined and represent neither symbol. Voltages outside the 0 and 1 ranges may cause permanent damage to the equipment receiving the signals.

ability to represent signals with arbitrary precision could implement this signal processing.

In this book we will look at how the digital systems that form such a large part of all of our lives function and how they are designed.

1.2 Digital Signals

Digital systems store, process, and transport information in digital form. That is the information is represented as discrete symbols that are encoded into ranges of a physical quantity. Most often we represent information with just two symbols, "0" and "1", and encode these symbols into voltage ranges as shown in Figure 1.1. Any voltage in the ranges labeled "0" and "1" represents a "0" or "1" symbol respectively. Voltages between these two ranges, the region labeled "?" are undefined and represent neither symbol. Voltages outside the ranges, below the "0" range or above the "1" range are not allowed and may permanently damage the system if they occur. We call signal encoded in the manner shown in Figure 1.1 a *binary* signal because it has two valid states.

Table 1.1 shows the JEDEC JESD8-5 standard for encoding a binary digital signal in a system with a 2.5V power supply. Using this standard, any signal with a voltage between -0.3V and 0.7 volts is considered to be a "0" and a signal with a voltage between 1.7V and 2.8V is considered to be a "1". Signals that don't fall into these two ranges are undefined. If a signal is below -0.3V or above 2.8V, it may cause damage[1].

Digital systems are not restricted to binary signals. One can generate a digital signal that can take on three, four, or any finite number of discrete values. However, there are few advantages to using more than two values and the circuits that store and operate on binary signals are simpler and more robust than their multi-valued counterparts. Thus, except for a few niche applications, binary signals are universal in digital systems today.

Digital signals can also be encoded using physical quantities other than voltage. Almost any physical quantity that can be easily manipulated and sensed

[1] The actual specification for V_{max} is $V_{DD} + 0.3$, where V_{DD}, the power supply, is allowed to vary between 2.3 and 2.7V.

Parameter	Value	Description
V_{min}	-0.3V	Absolute minimum voltage below which damage occurs
V_0	0.0V	Nominal voltage representing logic "0"
V_{OL}	0.2V	Maximum output voltage representing logic "0"
V_{IL}	0.7V	Maximum voltage considered to be a logic "0" by a module input
V_{IH}	1.7V	Minimum voltage considered to be a logic "1" by a module input
V_{OH}	2.1V	Minimum output voltage representing logic "1"
V_1	2.5V	Nominal voltage representing logic "1"
V_{max}	2.8V	Absolute maximum voltage above which damage occurs

Table 1.1: Encoding of binary signals for 2.5V LVCMOS logic. Signals with voltage in $[-0.3, 0.7]$ are considered to be a 0 signals with voltage in $[1.7, 2.8]$ are considered to be a 1. Voltages in $[0.7, 1.7]$ are undefined. Voltages outside of $[-.3, 2.8]$ may cause permanent damage.

can be used to represent a digital signal. Systems have been built using electrical current, air or fluid pressure, and physical position to represent digital signals. However, the the tremendous capability of manufacturing complex systems at low cost as CMOS integrated circuits has made voltage signals universal today.

1.3 Digital Signals Tolerate Noise

The main reason that digital systems have become so pervasive, and what distinguishes them from *analog* systems is that they can process, transport, and store information without it being distorted by noise. This is possible because of the discrete nature of digital information. A binary signal represents either a "0" or a "1". If you take the voltage that represents a "1", V_1, and disturb it with a small amount of noise, ϵ, it still represents a "1". There is no loss of information with the addition of noise, until the noise gets large enough to push the signal out of the "1" range. In most systems it is easy to bound the noise to be less than this value.

Figure 1.2 compares the effect of noise on an analog system (Figure 1.2(a)) and a digital system (Figure 1.2(b)). In an analog system information is represented by an analog votage, V. For example, we might represent temperature (in degrees Fahrenheit) with voltage according to the relation $V = 0.2(T - 68)$. So a temperature of 72.5 degrees is represented by a voltage of 900mV. This representation is continuous; every voltage corresponds to a different temperature. Thus, if we disturb the signal V with a noise voltage ϵ, the resulting signal

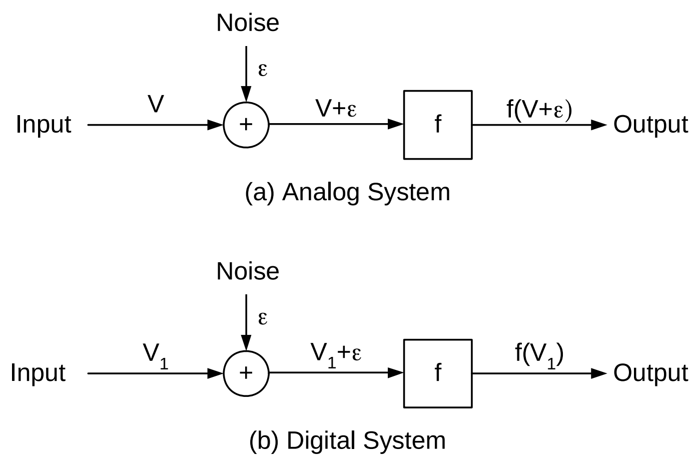

Figure 1.2: Effects of noise in analog and digital systems. (a) In an analog system perturbing a signal V by noise ϵ results in a degraded signal $V + \epsilon$. Operating on this degraded signal with a function f gives a result $f(V+\epsilon)$ that is different from the result of operating on the signal without noise. (b) In a digital system, adding noise ϵ to a signal V_1 representing a symbol, 1, gives a signal $V_1 + \epsilon$ that still represents the symbol 1. Operating on this signal with a function f gives the same result $f(V_1)$ as operating on the signal without the noise.

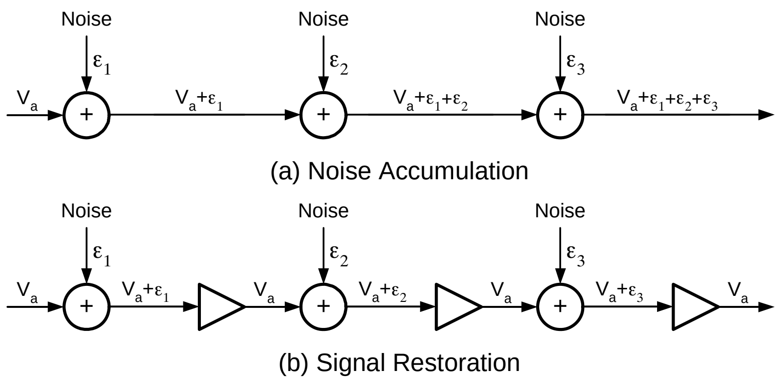

Figure 1.3: Restoration of digital signals. (a) Without restoration signals accumulate noise and will eventually accumulate enough noise to cause an error. (b) By restoring the signal to its proper value after each operation noise is prevented from accumulating.

$V + \epsilon$ corresponds to a different temperature. If $\epsilon = 100$mV, for example, the new signal $V + \epsilon = 1V$ corresponds to a temperature of 73 degrees ($T = 5V + 68$) which is different from the original temperature of 72.5 degrees.

In a digital system, on the other hand, each bit of the signal is represented by a voltage, V_1 or V_0 depending on whether the bit is "1" or "0". If a noise source perturbs a digital "1" signal V_1 for example, as shown in Figure 1.2(b), the resulting voltage $V_1 + \epsilon$ still represents a "1" and applying a function to this noisy signal gives the same result as applying a function to the original signal. Moreover, if a temperature of 72 is represented by a three-bit digital signal with value 010 (see Figure 1.6(c)), the signal still represents a temperature of 72 even after all three bits of the signal are disturbed by noise - as long as the noise is not so great as to push any bit of the signal out of the valid range.

To prevent noise from accumulating to the point where it pushes a digital signal out of the valid "1" or "0" range, we periodically restore digital signals as illustrated in Figure 1.3. After transmitting, storing and retrieving, or operating on a digital signal, it may be disturbed from its nomimal value V_a (where a is 0 or 1) by some noise ϵ_i. Without restoration (Figure 1.3(a)) the noise accumulates after each operation and eventually will overwhelm the signal. To prevent accumulation, we restore the signal after each operation. The restoring device, which we call a *buffer*, outputs V_0 if its input lies in the "0" range and V_1 if its output lies in the "1" range. The buffer, in effect, restores the signal to be a pristine 0 or 1, removing any additive noise.

This capability of restoring a signal to its noiseless state after each operation enables digital systems to carry out complex high-precision processing. Analog systems are limited to performing a small number of operations on relatively

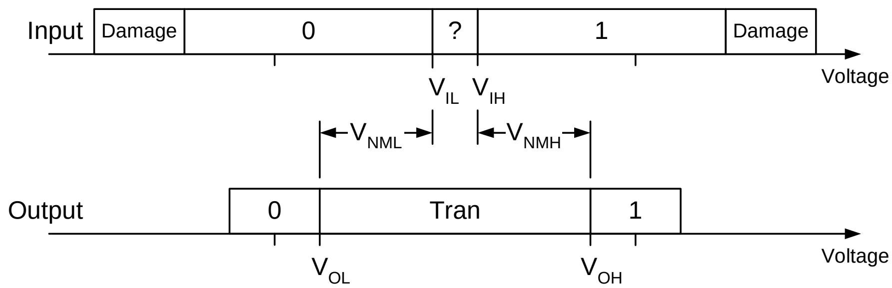

Figure 1.4: Input and output voltage ranges. (Top) Inputs of logic modules interpret signals as shown in Figure 1.1. (Bottom) Outputs of logic modules restore signals to narrower ranges of valid voltages.

low-precision signals because noise is accumulated during each operation. After a large number of operations the signal is swamped by noise. Since all voltages are valid analog signals there is no way to restore the signal between operations. Analog systems are also limited in precision. They cannot represent a signal with an accuracy finer than the background noise level. Digital systems on the other hand can perform an indefinite number of operations and, as long as the signal is restored after each operation, no noise is accumulated. Digital systems can also represent signals of arbitrary precision without corruption by noise.[2]

In practice, buffers, and other restoring logic devices, do not guarantee to output exactly V_0 or V_1. Variations in power supplies, device parameters, and other factors lead the outputs to vary slightly from these nominal values. As illustrated in the bottom half of Figure 1.4, all restoring logic devices guarantee that their 0 outputs fall into a 0 range that is narrower than the input 0 range and similarly for 1 outputs. Specifically, all 0 signals are guaranteed to be less than V_{OL} and all 1 signals are guaranteed to be greater than V_{OH}. To ensure that the signal is able to tolerate some amount of noise, we insist that $V_{OL} < VIL$ and that $V_{IH} < V_{OH}$. For example, the values of V_{OL} and V_{OH} for 2.5V LVCMOS are shown in Table 1.1. We can quantify the amount of noise that can be tolerated as the *noise margins* of the signal:

$$\begin{aligned} V_{NMH} &= V_{OH} - V_{IH}, \\ V_{NML} &= V_{IL} - V_{OL}. \end{aligned} \tag{1.1}$$

While one might assume that a bigger noise margin would be better, this is not necessarily the case. Most noise in digital systems is induced by signal transitions and hence tends to be proportional to the signal swing. Thus, what is really important is the *ratio* of the noise margin to the signal swing, $\frac{V_{NM}}{V_1 - V_0}$

[2]Of course one is limited by analog input devices in acquiring real-world signals of high precision.

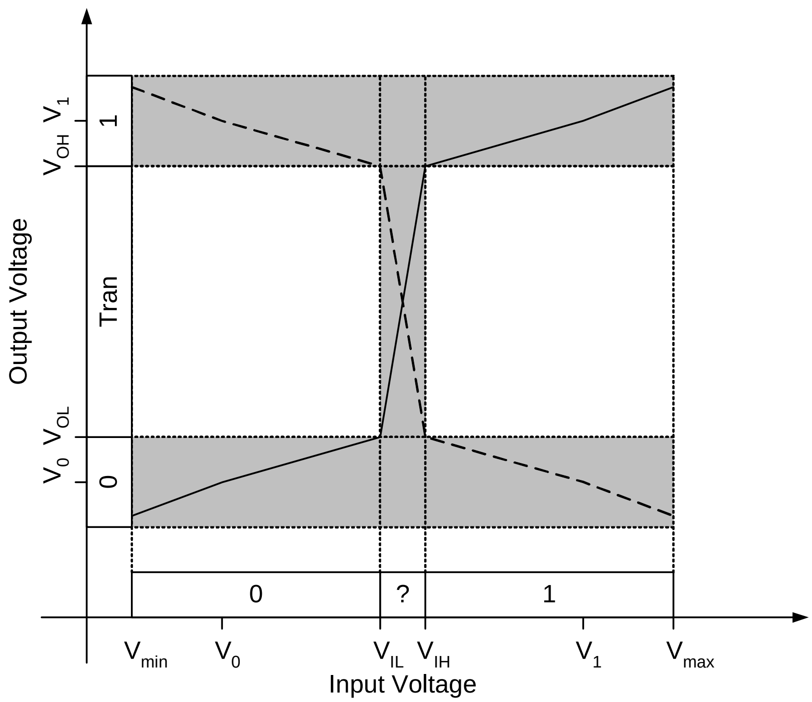

Figure 1.5: DC transfer curve for a logic module. For an input in the valid ranges, $V_{min} \leq V_{in} \leq V_{IL}$ or $V_{IH} \leq V_{in} \leq V_{max}$, the output must be in the valid output ranges $V_{out} \leq V_{OL}$ or $V_{OH} \leq V_{out}$. Thus, all valid curves must stay in the shaded region. This requires that the module have gain > 1 in the invalid input region. The solid curve shows a typical transfer function for a non-inverting module. The dashed curve shows a typical transfer function for an inverting module.

rather than the absolute magnitude of the noise margin. We will discuss noise in more detail in Chapter 5.

Figure 1.5 shows the relationship between DC input voltage and output voltage for a logic module. The horizontal axis shows the module input voltage and the vertical axis shows the module output voltage. To conform to our definition of *restoring* the the transfer curve for all modules must lie entirely within the shaded region of the figure so that a input signal in the valid 0 or 1 range will result in an output signal in the narrower output 0 or 1 range. Non-inverting modules, like the buffer of Figure 1.3 have transfer curves similar to the solid line. Inverting modules have transfer curves similar to the dashed line. In either case, gain is required to implement a restoring logic module. The

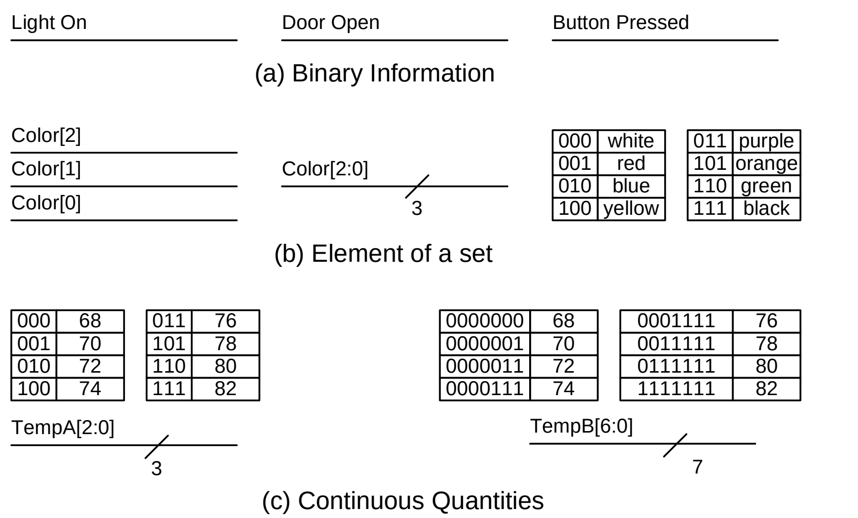

Figure 1.6: Representing information with digital signals. (a) binary-valued predicates are represented by a single-bit signal. (b) elements of sets with more than two elements are represented by a group of signals. In this case one of eight colors is denoted by a three-bit signal `Color[2:0]`. (c) A continuous quantity, like temperature, is *quantized* and the resulting set of values is encoded by a group of signals. Here one of eight temperatures can be encoded as a three-bit signal `TempA[2:0]` or as a seven-bit *thermometer-coded* signal `TempB[6:0]` with at most one transition from 0 to 1.

absolute value of the maximum slope of the signal is bounded by

$$\max \left| \frac{dV_{out}}{dV_{in}} \right| \geq \frac{V_{OH} - V_{OL}}{V_{IH} - V_{IL}}. \tag{1.2}$$

From this we conclude that restoring logic modules must be active elements capable of providing gain.

1.4 Digital Signals Represent Complex Data

Some information is naturally binary in nature and can be represented with a single binary digital signal (Figure 1.6(a)). Truth propositions or predicates fall into this category. For example a single signal can indicate that a door is open, a light is on, a seatbelt is buckled, or a button is pressed.

Often we need to represent information that is not binary in nature: a day of the year, the value and suit of a playing card, the temperature in a