1 Introduction to Jitter

Imagine a lunar eclipse is to occur later tonight at around 1:57 am when you are fast asleep. You decide to set your camera to capture the event, but being uncertain of the actual time, you would set the camera to start a few minutes before the nominal time and run it for a few minutes after 1:57 am. In this process, you have included a margin around the nominal time so as to minimize the risk of not capturing the event on your camera.

Imagine you have an important meeting at 8:00 am sharp, but you are uncertain about your watch being a bit too fast or too slow. Determined not to be late, you decide to arrive by 7:55 am according to your watch, just in case your watch is a bit slow. This is indeed a practical way to deal with uncertainty in time.

Imagine you have a ticket for the bullet train in Tokyo that leaves the train station at 1:12 pm sharp. You are at the station and on the right platform, and you take the train that leaves the station at 1:12 pm sharp. But a few minutes later, you notice that your watch is two minutes fast compared to the time being displayed on the train, and you realize that you are on the 1:10 pm train and you are moving towards a different destination.

Missing an event, being late or early, and getting on the wrong train are all consequences of timing uncertainties in our daily lives. In digital circuits, we deal with very similar situations when we try to time events by a clock that has its own timing uncertainty, called *jitter*, and in doing so we may miss an event, such as not capturing critical data, or cause bit errors, e.g., capturing the wrong data. Our goal, however, is to prevent such errors from occurring, or to minimize their probabilities. We do this first by carefully studying the nature of these uncertainties and modeling their characteristics.

This chapter provides a few concrete examples from electronic circuits and systems where timing uncertainties have a profound impact on the accuracy of their operation. Our goal here is to qualitatively introduce the concept of jitter and intuitively explain how it should be characterized and dealt with. A formal definition of jitter, its types, and its full characterization will be presented in Chapter 2.

1.1 What Is Clock Jitter?

This section provides an intuitive explanation of two fundamental jitter concepts: period jitter and absolute jitter.

1.1.1 Period Jitter

Most microprocessors (μP) and digital signal processing (DSP) units work with clock signals to time the execution of instructions. A basic instruction such as a shift by one bit may take a single clock cycle, whereas addition and subtraction instructions may take three to four clock cycles, and more complex instructions, such as multiplication, may take tens of clock cycles to complete. In all these cases, the underlying assumption is that the clock is accurate; that is, the clock cycles are all identical in duration. In reality, however, the cycles of a clock are only identical with a finite (not infinite) accuracy. This is because the clock signals are generated using physical devices that inevitably have some uncertainty or randomness associated with them. For example, a clock may be generated by three identical CMOS inverters in a loop, as shown in Figure 1.1. If we assume t_{pd} is the propagation delay of each inverter in response to a transition at the input, then it takes $3t_{pd}$ for a state "0" at node v_{out} to transition to state "1" and another $3t_{pd}$ to transition back to "0". In total, it takes $6t_{pd}$ for the clock to complete one cycle, and therefore we can write $T_{nom} = 6t_{pd}$ where T_{nom} represents a nominal clock cycle. However, strictly speaking, t_{pd} is not constant, but rather a random variable that results in a different propagation delay every time the inverter is used. This randomness is inherent in any physical device that deals with the motion of electrons at temperatures above zero degrees Kelvin.

If we accept that t_{pd} is a random variable, then the clock period, which is the sum of six random variables, is also a random variable that deviates from its nominal (or expected) value. The good news is that these random variables may have tight distributions and, as such, may not adversely affect the circuit operations, especially at low clock frequencies where the period is much larger than the deviations. However, as we increase the clock frequency, the same absolute deviation in period may compromise circuit operation.

A typical clock frequency of 4GHz, for example, corresponds to a nominal period of 250ps, but this period may have a normal (Gaussian) distribution with a standard

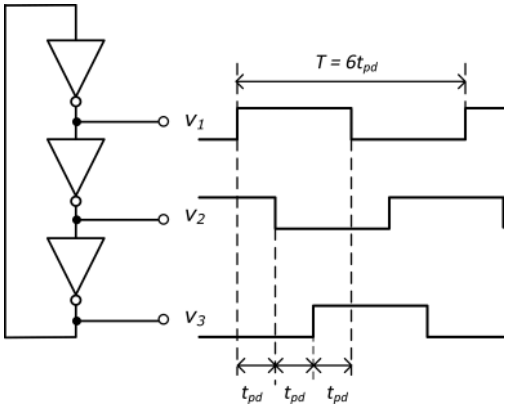

Figure 1.1 A simple ring oscillator and its voltage waveforms.

deviation of σ = 1ps. This implies that a clock cycle is close to 250ps but may deviate from this value with a probability that is a decreasing function of the deviation magnitude. For example, as we will see in Chapter 9 (Table 9.1), the probability of the period being smaller than 243ps is about 10^{-12}. What does this mean for designs using this clock? A straightforward answer is that we must ensure that any task that is supposed to complete in one clock cycle must do so assuming the worst-case clock cycle, that is, 243ps (not the nominal 250ps). This will ensure that, with a high probability ($1-10^{-12}$), all the tasks are completed within one clock cycle even when one clock cycle is 7ps shorter than the nominal cycle. If we were to reduce the probability of failure (i.e., the probability of failing to meet the timing), we could simply design for a 242ps minimum cycle time while working with a nominal cycle time of 250ps. This reduces the probability of failure to 6.22×10^{-16}, which corresponds to one error in every 4.6 days (assuming 4GHz operation).

The concept we just described, i.e., the deviation of a clock cycle from its nominal value, is referred to as *clock period jitter*, or *period jitter* for short. A concept closely related to period jitter is *N-period jitter*, which is the deviation of a time interval consisting of N consecutive periods of the clock from the time interval of N nominal periods. We will formally define period jitter, N-period jitter, and other types of jitter for a clock signal in Chapter 2. Let us now provide an intuitive explanation for *absolute jitter*.

1.1.2 Absolute Jitter

Period jitter is only one method to characterize the timing uncertainties in a clock signal. This method is particularly useful in digital circuits where we are concerned with the time period we need to complete a task. In other applications, such as in clock and data recovery, where the clock edge is used to sample (capture) data, we are concerned with the instants of time, not with time durations! This is because sampling occurs at an instant of time, not over a period of time. In this case, it is the absolute time of the clock edge (at which sampling occurs) that matters, not the interval between the edges (i.e., the clock period). In these applications, it is important for the clock's edges to be precise; i.e., for them not to deviate from their ideal locations by certain amount. Let us elaborate on this further.

In an ideal clock, if we assume the first rising edge occurs at time $t = 0$, then the subsequent rising edges will occur at exactly $t = kT$, where k is a positive integer and T is the period of the ideal clock. In a non-ideal clock, the rising edges, t_k, will deviate from their ideal values kT. We refer to this deviation as the clock's *absolute jitter*, with the word *absolute* signifying the deviation in instants of time as opposed to deviations in intervals of time, as in period jitter.

The reader notes that absolute jitter and period jitter are closely connected concepts. Indeed, we will see in Chapter 2 how we can obtain one from the other.

Figure 1.2 shows an example of a jittery clock signal (CK) along with an ideal clock signal (CK_{ideal}). The absolute jitter can be abstracted from the clock signal and shown separately as a function of time. From this example, absolute jitter appears to be a

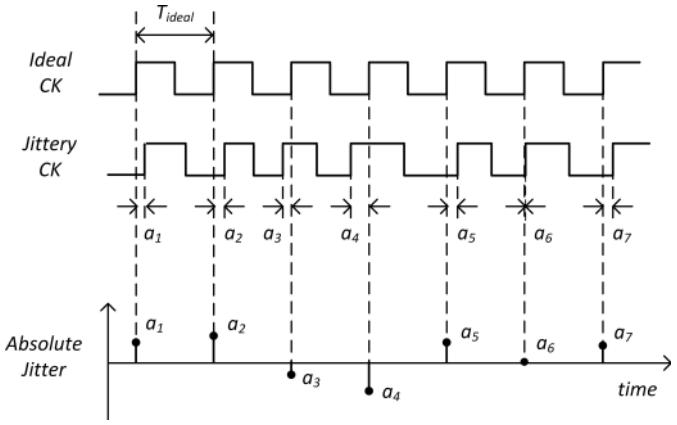

Figure 1.2 Ideal clock versus jittery clock waveforms.

discrete-time random signal. We will see later in Chapter 2 that, similar to any random signal, absolute jitter has a well-defined spectrum and, as such, may contain a continuous range of frequency components.

1.1.3 Intentional Jitter

Jitter is not always an unwanted property of a clock signal. In some applications, such as in spread-spectrum clocking, we intentionally add carefully controlled jitter to a clean clock so as to shape its power spectral density.

It is well known that a clean clock, by virtue of being a periodic signal, has energy peaks at its fundamental frequency and its second- and higher-order harmonics. At clock frequencies in excess of 100MHz, some of the harmonics may have a wavelength that is of the same order of magnitude as the length of the wire carrying the clock signal between chips on the same board or between boards. If left unattenuated, the signal energy of these harmonics may radiate and cause electromagnetic interference (EMI) with neighboring electronic components (mostly the components on the same PCB). For this reason, it is desirable to limit the amount of radiation a clock signal can produce. This radiation level is directly related to the peaks in the clock signal power spectrum.

Spread-Spectrum Clocking is a technique that spreads the peak energy over a larger frequency band, thereby reducing the peak values and their associated EMI. This is accomplished by adding a controlled amount of jitter to the clock. The jitter profile is typically in the form of a frequency offset that goes up and down linearly with time. Figure 1.3 shows an example of a frequency offset which increases linearly over a certain period of time (kT) and then decreases over the same period (kT). This increase in clock frequency (assuming $\Delta f \ll f_0$) is equivalent to a decrease in the period by a factor of $2\Delta f/f_0$.

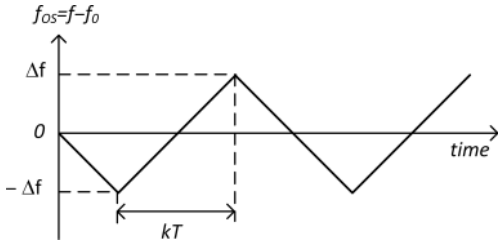

Figure 1.3 Frequency offset (f_{OS}) as a function of time.

Another example of using intentional jitter is in characterizing clock and data recovery circuits for their robustness to jitter. In this application, we intentionally add jitter to the incoming data and observe the system's robustness (in terms of its bit error rate) as a function of jitter frequency. We discuss this example in further detail in Chapter 7.

1.2 What Is Data Jitter?

Consider a D flip-flop with data input D, clock input CK, and output Q, where D is captured and transferred to Q at the rising edge of the clock. As a result, Q is synchronized with the clock, and therefore the clock jitter directly affects the timing of the data at the output. In clock and data recovery applications, it is common to refer to the duration of one data bit as a unit interval or UI. An ideal clock results in a data stream that has a fixed UI, where each UI is exactly equal to the clock period. We refer to this data stream as *ideal* or *clean* (i.e., the data without jitter). A non-ideal clock results in a jittery data stream where each UI is slightly different from the nominal UI. This situation is shown in Figure 1.4, where jittery data is compared against an ideal data stream. Note that, due to the random nature of the data sequence, data transitions do not occur at every UI edge. As such, jitter is not observable from data when there is no transition. However, by overlaying several UIs of data on top of each other, in what is known as an eye diagram, we can observe the overall characteristics of data jitter.

1.2.1 Eye Diagram

In clock and data recovery applications, the clock samples a jittery data waveform at instants that are one clock period (T) apart. These instants usually correspond to the rising edges of a clean clock, i.e., a clock with little or no jitter. Since the act of sampling repeats itself every T, we are interested in seeing how the data waveform looks if we chop it into segments of length T and overlay all the segments on top of each other. The resulting composite waveform, which resembles an eye, is known as an eye diagram.

Figure 1.5 shows the process by which we generate an eye diagram. This is essentially the data waveform as a function of time modulo 1T. The eye diagram clearly shows the

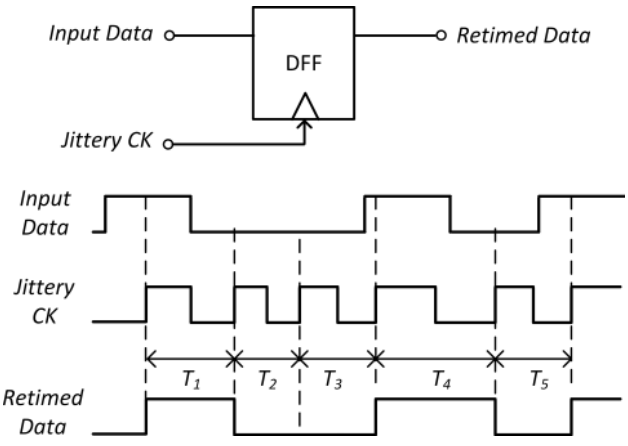

Figure 1.4 Input data is retimed with a jittery clock.

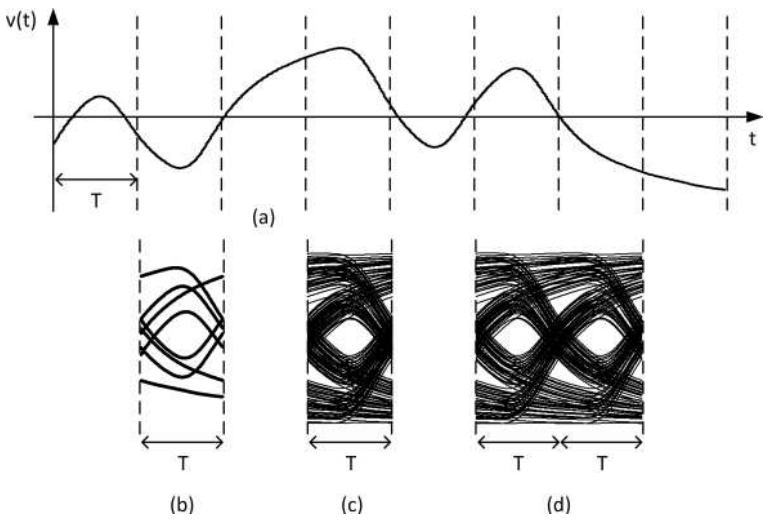

Figure 1.5 (a) A sketch of a voltage waveform as a function of time, (b) an eye diagram with eight traces, (c) an eye diagram with 500 traces, (d) an eye diagram showing two eyes (corresponding to two periods).

time instants at which the data "0" and "1" are farthest apart. This is the instant at which the eye height (also known as vertical eye opening) is at a maximum, and this is the instant at which we strive to sample the data by the rising edge of the clock.

The eye diagram also reveals how jitter can accumulate over time so as to close the eye in the horizontal direction (see Figure 1.6). Accordingly, the horizontal eye opening is an indication of the timing margin left for error-free operation.

When the noise, interference, attenuation, and jitter are excessive in a design, the eye may be completely closed: there is then no instant at which the data "1" and "0" can be

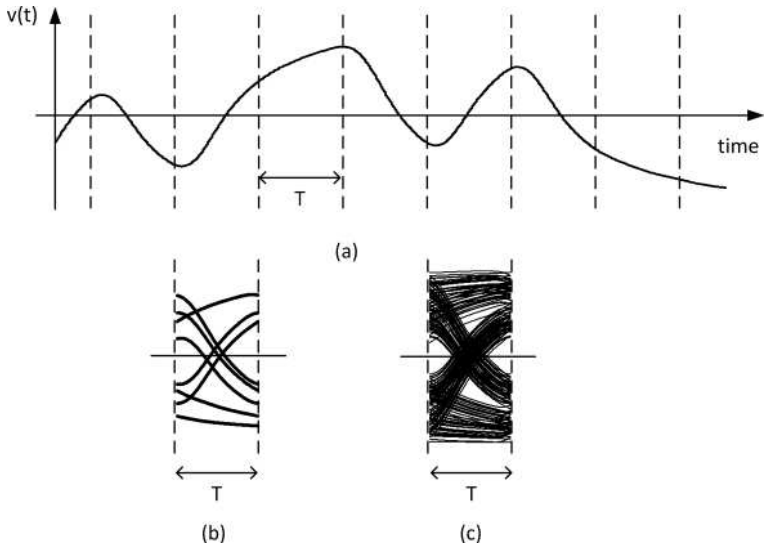

Figure 1.6 (a) A sketch of a voltage waveform as a function of time, (b) an eye diagram centered around the zero crossings (eight traces), (c) the same eye diagram using 500 traces.

consistently distinguished. These situations require equalization and jitter cleaning so as to open an otherwise closed eye for data detection.

1.2.2 Random Versus Deterministic Jitter

Jitter, in general, refers to any timing deviations from an ideal clock. These deviations may be random or deterministic depending on the underlying sources producing them. We have already provided several examples where random jitter is produced. Let us now consider an example that produces deterministic jitter. Consider a binary random sequence (produced using an ideal clock) that travels the length of a wire on a PCB. The wire acts as a low-pass filter and, as such, slows the data transitions, creating jitter. The jitter produced in this way is considered deterministic because one can predict the exact jitter if an accurate model of the wire is available. Note that this is in contrast with the random jitter produced by random movement of electrons, as we cannot predict the electron movement.

An easy way to characterize jitter is to look at its histogram or its probability density function (PDF). As shown in Figure 1.7, deterministic jitter is bounded (having a histogram with limited range) whereas random jitter is unbounded (having a Gaussian-like PDF). In most cases, several jitter sources are at work at the same time, producing composite jitter that includes both deterministic and random jitter. We will discuss this in depth in Chapters 2 and 7 and describe ways to decompose jitter into its components. Independent of the type of jitter, however, one can always characterize jitter, which is a discrete-time sequence, by its peak-to-peak or its root mean square (RMS) values.

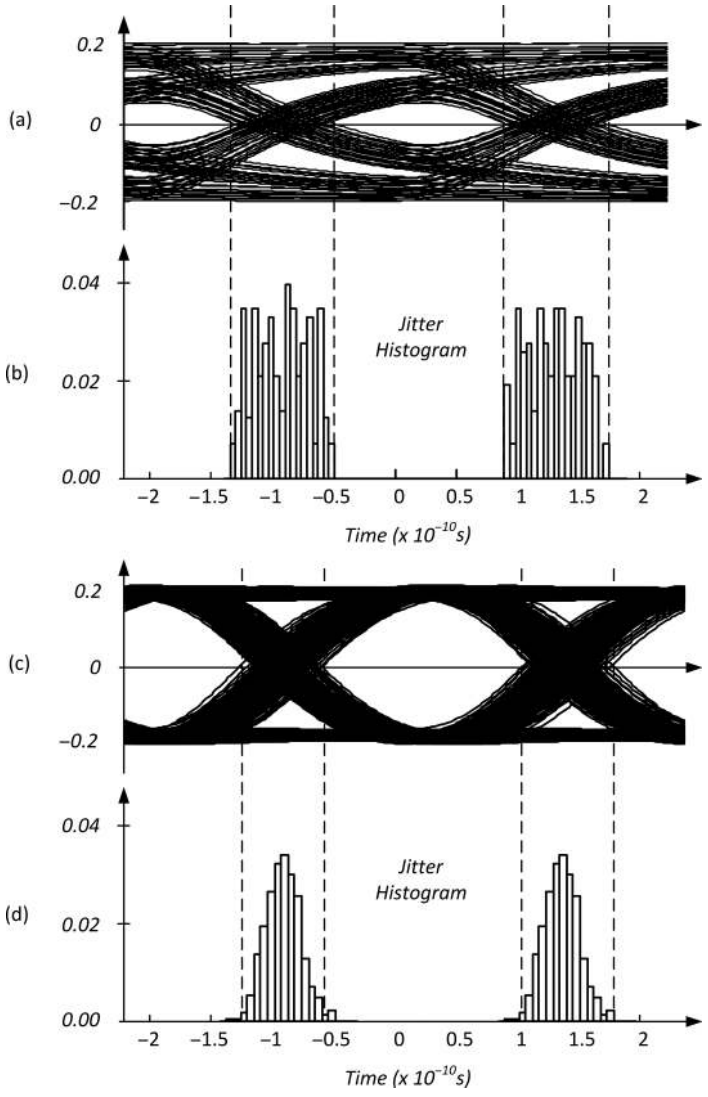

Figure 1.7 Eye diagrams and their corresponding jitter histograms (a) eye diagram with deterministic jitter only, (b) corresponding jitter histogram, (c) eye diagram with random jitter only, (d) corresponding jitter histogram.

1.3 Jitter in Measuring Time

We often measure time by observing the completion of a physical phenomenon or event. For example, in an hourglass (also known as a sandglass), shown in Figure 1.8, the event is the falling of sand from the top to the bottom compartment. The completion of this event marks one unit of time, which may be from a few minutes to a few hours depending on the hourglass design. Note that an hourglass has limited

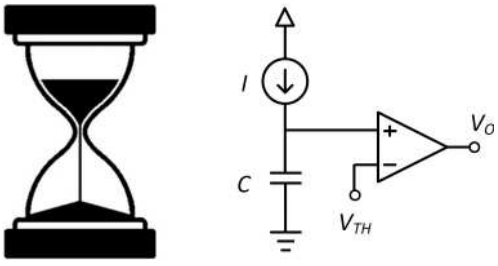

Figure 1.8 An hourglass behaves similarly to a current source charging a capacitor.

accuracy and limited resolution. The accuracy relates to the variation in time measurement as we repeat the same experiment. Depending on the size distribution of the grains of sand and their random arrangement, we will actually experience a slightly different time interval every time we use the hourglass. This variation (or uncertainty) is considered the jitter in the hourglass. The resolution relates to the minimum time interval (seconds or fractions of a second, for example) that could be measured using the hourglass.

If we replace the grains of sand with electrons and the glass with a capacitor, then we have an electronic version of an hourglass that can measure time with much better accuracy and far better resolution. Figure 1.8 shows an ideal current source charging a capacitor. The event in this case is defined as the time it takes for the capacitor to charge from 0V (corresponding to an empty capacitor) to a threshold voltage, V_{TH} ($V_{DD}/2$, for example). If the current source is ideal, having an amplitude of I_0, this event takes CV_{TH}/I_0 to complete. Accordingly, this advanced hourglass has several knobs to turn in order to adjust its resolution. For example, one could reduce the capacitance, increase the current, or lower the threshold voltage of the comparator. The accuracy, however is limited by the current noise that is inherent in the methods by which we implement a current source.

A very simple current source may be implemented using a PMOS transistor with its gate connected to ground, as shown in Figure 1.9. This current source differs from an ideal current source in two ways: first, the current it produces is not constant; it becomes smaller as the capacitor is charged from 0 to V_{TH}; and second, the current includes noise such as thermal noise due to the random movements of electrons. The voltage dependency of the current slows down the process of charging the capacitor as there is simply less current. This deviation in time, however, can easily be absorbed by calibrating the time unit. The current noise, on the other hand, will result in different measurements of time as we repeat the experiment. To see this, let us model our practical current source as a constant current source in parallel with a noise current source, as shown in Figure 1.9. The current is now the sum of the nominal current, I_0, and the noise current. The additional noise current will cause the voltage across the capacitor to deviate from its ideal waveform, which is simply a ramp, and to arrive faster or slower at V_{TH}. This time deviation is a random process, which we call jitter, and is directly related to the characteristics of the noise current (itself a random process).

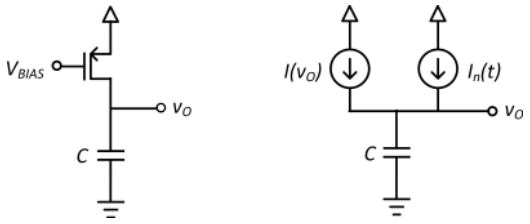

Figure 1.9 A simple PMOS current source charging a capacitor.

1.4 Jitter in a Ring Oscillator

The circuits we use to generate clock signals are called *oscillators*. One common type of oscillator, called a *ring oscillator*, consists of three inverters in a closed loop (a ring), as shown in Figure 1.1. We analyze this ring oscillator to explain intuitively how jitter is generated. A more rigorous treatment of this topic will be presented later in Chapter 4, once we mathematically define jitter and its characteristics.

As mentioned earlier in this chapter, the ring oscillator of Figure 1.1 produces oscillations (in voltage) with an expected period of $6t_{pd}$, where t_{pd} is the expected value of the propagation delay through each of the three stages. Let us now explore the properties of the delay of a CMOS inverter.

1.4.1 Jitter in Delay of a CMOS Inverter

When the input of an inverter rises from 0 to VDD, its output falls from VDD to 0. The high-to-low delay of an inverter (denoted by t_{phl}) is defined as the time elapsed between the input (at 50% of its swing) and the output (at 50% of its swing). The low-to-high delay of an inverter (denoted by t_{plh}) is defined similarly, as shown in Figure 1.10. If we assume the input transitions from low to high and high to low occur instantaneously, then t_{phl} and t_{plh} simply correspond to the time it takes for the output to reach 50% of its full swing. With this simplifying assumption, during the low-to-high transition of the output, only the PMOS transistor is ON, while during the high-to-low transition of the output, only the NMOS transistor is ON. Accordingly, we have two equivalent circuits to calculate t_{phl} and t_{plh}, as shown in Figure 1.11. In both circuits, two current sources charge or discharge the load capacitor. Of the two current sources in parallel, one is assumed to be deterministic and controlled by the gate voltage. The other source provides a random current corresponding to the thermal movement of electrons.

The deterministic parts of t_{phl} and t_{plh} are simply $C_L V_{TH}/I_0$. We refer to these as the *base delays* of the inverter. The remaining parts are two random variables which are also functions of time. We refer to these as the *excess delays*, or jitter, of the inverter. Jitter is random for the following reasons: if we measure two identical inverters at the same time, they exhibit different noise currents and hence result in different measurements of the delays. They are also time-dependent because if we attempt to measure t_{phl} (for