Chapter 2

Electrical Characteristics of Gates

In the ideal digital world we have considered up to now, all low logic signals have been considered to be 0V (ground) and all high logic signals have been considered to be at the power supply voltage (V_{CC}). In real life this is not the case. This chapter introduces data sheets for logic circuits and tells how to determine their electrical characteristics. From this it can then be determined just how they will perform when connected to other logic circuits and other electrical devices.

Before we begin our discussion of the electrical characteristics of digital logic circuits, a DISCLAIMER is in order. Much of this chapter focuses on TTL 7400-family logic circuits and their associated data sheet information. That said, it should be clearly understood that this is not meant to imply that TTL logic circuits are a major technology of choice for today's digital systems. On the contrary — TTL was popular in the 1970's and early 1980's when digital systems were built mainly from gate-level logic circuits on printed circuit boards. Today's systems are built principally from four kinds of digital components: (1) software programmable chips such as CPUs, DSPs, and Application-Specific Instruction Set Processors (ASIPs), (2) Field-Programmable Gate Arrays (FPGAs), (3) Application Specific Standard Products (ASSPs), and (4) memories. These are all based on high-integration VLSI design methods and fabrication technologies (mainly CMOS), and typically contain many million transistors per chip. In contast, TTL logic circuits were based mostly on bipolar technology and contained tens or hundreds of transistors per chip. Why, then, the initial focus on TTL parts? The main reason is that they are a simple vehicle with which to introduce data sheet-based DC and AC interfacing considerations. Further, their function is simple enough that the number of data sheet parameters required to discuss at this introductory stage is minimal. The assumption is that, once introduced to the basic concepts of electrical interfacing using TTL parts, the student can easily transfer that knowledge to more complex (and modern!) logic circuits and technologies. To that end, a number of the end-of-chapter homework exercises involve such modern technologies and parts.

2.1 DC Voltage Characteristics of a Typical Inverter

A *data sheet* is a document which describes a logic circuit and is usually supplied by the manufacturer of the logic circuit. Table 2.1 shows a portion of a data sheet for an inverter. It is the portion of the data sheet that deals with the static electrical characteristics of the inverter. These are usually called the inverter's 'DC' characteristics. Using the information found in the data sheet, you can determine the range of allowable input voltages and currents that the inverter will respond to. Further, you can determine the range of voltages and currents it will produce on its outputs.

Table 2.1: The DC Portion of a Data Sheet for an Inverter

	Quantity	Description	min	nom	max	units
1	V_{CC}	Supply voltage	4.75	5	5.25	V
2	V_{IH}	High-level input voltage	2			V
3	V_{IL}	Low-level input voltage			0.8	V
4	V_{OH}	V_{CC} = min, V_{IL} = 0.8V, I_{OH} = -0.4mA	2.4	3.4		V
5	V_{OL}	V_{CC} = min, V_{IH} = 2.0V, I_{OL} = 16mA		0.2	0.4	V
6	I_{OH}	High-level output current			-0.4	mA
7	I_{OL}	Low-level output current			16	mA
8	I_{IH}	V_{CC} = max, V_I = 2.4V			40	μA
9	I_{IL}	V_{CC} = max, V_I = 0.4V			-1.6	mA

From a schematic, one might assume that an inverter is a two-terminal device (a single input and a single output). In reality, it requires three inputs and produces one output. The additional two inputs are its power and ground terminals as shown in Figure 2.1.

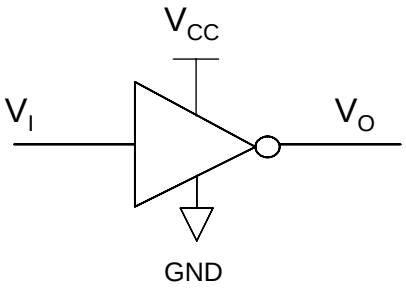


Figure 2.1: Schematic Symbol for an Inverter Showing Power and Ground Terminals

By convention, all voltages contained in a data sheet are referenced to the ground terminal. Line 1 of the data sheet specifies that allowable power supply voltages (V_{CC}) are from $4.75V$ to $5.25V$. The implication is that if V_{CC} falls outside this range, the inverter may not operate properly.

Line 2 of Table 2.1 (V_{IH}) states that any input voltage of $2V$ or greater will be interpreted as a valid '1' input. The implication is that any voltage below $2V$ will not be considered a valid '1' input. Line 3 (V_{IL}) states that any input voltage less than or equal to $0.8V$ will be interpreted as a valid '0' input.

These two relations are shown graphically in Figure 2.2. This figure shows that a wide range of voltages (from $2V - 5V$) will be interpreted as a logical '1', while a smaller range of voltages (from $0V - 0.8V$) will be interpreted as a logical '0'. Also of interest: the figures shows that the range of voltages between $0.8V$ and $2.0V$ is neither a '1' or a '0' — the behavior of the inverter is undefined for input voltages in this range.

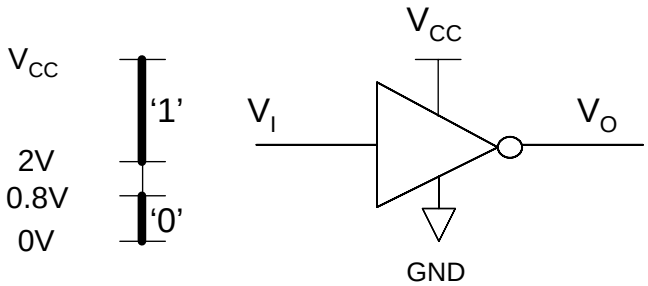


Figure 2.2: Inverter Input Voltage Ranges

In addition to accepting a range of input voltages for logical high and low values, the inverter will produce a range of voltages on its output as well. In the data sheet of Table 2.1, V_{OH} is guaranteed to be *at least* $2.4V$. That is, the data sheet promises that when the inverter is attempting to drive a '1' on its output, the actual voltage on the output wire will be at least $2.4V$. This is a guaranteed minimum voltage (worst case value). The same line in the data sheet shows that the *typical* output for a logical '1' will be about $3.4V$. Similarly, the data sheet promises that when the inverter is driving its output low (V_{OL}), the output voltage is guaranteed to be no greater than $0.4V$ worst case and $0.2V$ typical.

These two relations are shown graphically in Figure 2.3. The implication is that if we provide an inverter with inputs that are within the proper bounds for V_{IH} and V_{IL}, the inverter will respond by driving its output within the ranges specified for V_{OH} and V_{OL}. The figure also implies that for proper input values, an inverter will never drive its output to a value in the range $0.4V - 2.4V$.

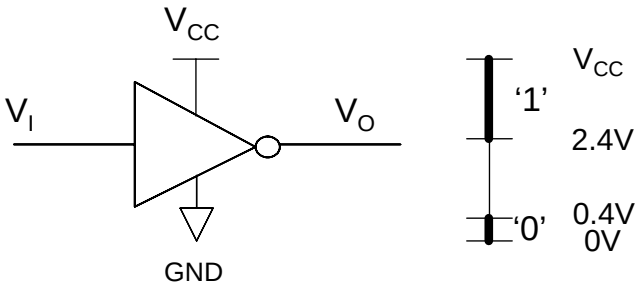


Figure 2.3: Inverter Output Voltage Ranges

2.1.1 Voltage Considerations for Interfacing Logic Gates

Using the information from the data sheet fragment of Table 2.1 it is straightforward to determine whether two such inverters can be successfully connected together into the circuit of Figure 2.4. To aid in this process, the the output and input voltage ranges for the two inverters are included below the circuit. On the left are the output voltages for Inverter1 and on the right are the legal input voltages for Inverter2.

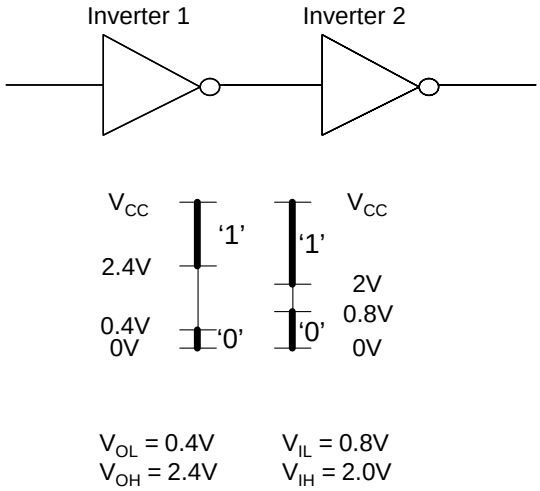


Figure 2.4: Sample Circuit for Voltage Interfacing Considerations

The question to be answered is this: "Is Inverter1 guaranteed to produce output voltages that are acceptable input voltages for Inverter2?" We begin by examining the voltages for a logic '0' value — Inverter1 is guaranteed output a logic '0' as a voltage between $0V - 0.4V$ while Inverter2 requires a logic '0' to be a voltage between $0V - 0.8V$. The figure shows graphically that the output voltage from Inverter1 is a subset of the legal Inverter2 '0' input voltage.

As a result, we conclude that, at least for purposes of logic '0' values, Inverter1 can indeed drive Inverter2. Similar reasoning shows that Inverter1's output is within the range expected by Inverter2 for a logic '1'. We thus can conclude that, based on DC voltage considerations, Inverter1 can indeed driver Inverter2 and the net behavior of the circuit would be as expected from a simple boolean truth table analysis of the composite circuit.

An alternative algebraic method for determining this is shown by the inequalities at the bottom of the figure. On the left the guaranteed output voltage ranges for Inverter1 are given (V_{OL} and V_{OH}), and on the right the required input voltage ranges for Inverter2 are given (V_{IL} and V_{IH}). Since both inverters share the same ground and power supply voltages, for proper operation the following two inequalities must hold: $V_{OL} \leq V_{IL}$ and $V_{OH} \geq V_{IH}$.

2.2 Current Characteristics of a Typical Inverter and Interfacing Considerations

The above analysis is insufficient to determine whether Inverter1 and Inverter2 can be successfully interfaced because it has ignored current considerations. Any logic gate has finite current drive capability on its output. Some logic gates consume or produce non-zero input currents, and thus a current-based analysis is also required to determine whether Inverter1 and Inverter2 are compatible in the configuration shown in Figure 2.4.

Returning back to Table 2.1, we see that the data sheet includes current values as well. The sixth line of the data sheet (I_{OH}) specifies that when driving its output high, the inverter is able to output a maximum of $0.4mA$ of current [1]. Further, the I_{IH} line in the data sheet indicates that in the worst case (when its input it high), the inverter will sink a maximum of $40\mu A$ of current. This is only 10% of the current that the inverter can output.

Similarly, when the inverter is driving a logic '0' on its output, the I_{OL} line of the data sheet indicates that the inverter can sink $16mA$ of current. The I_{IL} line further shows that the inverter can source $1.6mA$ of current to its input. Figure 2.5 shows the two inverter scenario once again but with these currents marked. The questions to be asked are these: "Can Inverter1 source enough current to drive Inverter2's input high and can it sink enough current to pull Inverter2's input low?" Since $|I_{OH}| \geq |I_{IH}|$ and $|I_{OL}| \geq |I_{IL}|$, we see that the circuit will work from a consideration of input and output currents.

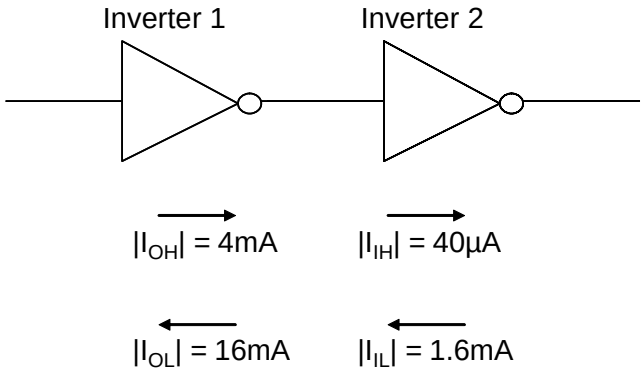


Figure 2.5: Sample Circuit for Current Interfacing Considerations

In fact, what we learn from this example, is that Inverter1 can drive up to 10 copies of Inverter2 without violating its output current capabilities. Put another way, we would say that Inverter1 *has a fanout limit of* 10 when driving circuits with the input characteristics of Inverter2.

2.3 Logic Families, I/O Standards and Interfacing Considerations

The analysis of Sections 2.1 and 2.2 to determine whether our two inverters can be interfaced to one another is simplified by the concepts of *logic families* and *I/O standards*.

[1] Note that it is listed as a $-0.4mA$ value in the data sheet — the convention is that when the inverter is sourcing current to its output wire the current is a negative value. Conversely, when the inverter is sinking current from its output wire the current is a positive value.

2.3.1 Logic Families

In the past, discrete TTL logic circuits were a popular way of creating digital logic circuits. These were small IC packages, each containing some small number of logic circuits. For example, Figure 2.6 is a graphical depiction of a dual in-line package (DIP) for the TTL 7404 part. The physical dimensions of the package are approximately $0.75in \times 0.3in$ and the package has fourteen pins around its periphery. These are numbered in the figure. Further, the relationship between the six inverters in the chip contained in the package and the package pins is also shown graphically in the figure.

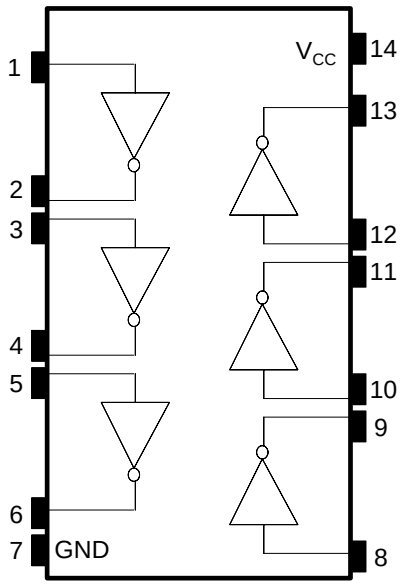


Figure 2.6: Depiction of DIP Package of Inverters (TTL 7404)

A logic family such as TTL is a collection of logic circuits, all based on the same integrated circuit technology. All of the circuits in a logic family are designed to be able to be interfaced to all the other circuits in the same logic family. Thus, if Inverter1 and Inverter2 from Figure 2.4 are known to be from the same logic family (the TTL 7400 series logic family, for example), it is known *a priori* that they are electrically compatible. Table 2.2 provides a small sampling of the parts from the popular TTL 7400 series of logic. As can be seen, the family contains both combinational as well as sequential circuits.

Table 2.2: Partial Listing of TTL 7400 Series Family Parts

Part No.	Description	Number of circuits per package
7400	2-input NAND gate	4
7404	Inverter	6
7408	2-input AND gate	4
7410	3-input NAND gate	3
7430	8-input NAND gate	1
7447	BCD to 7 Seg decoder/driver	1
7474	Rising Edge DFF with asynch preset/clear	4
74193	4-bit Binary up/down counter	1

Over the years, a number of variations on the 7400 family were produced, each with different speed and power characteristics. These were typically designated using additional letters in the part number. Thus, a 74S30 is an 8-input NAND gate from the S (Schottky) family. Similarly, an 'LS' is used to indicate the Low Power Schottky TTL logic family and 'ALS' is used to indicate the 'Advanced Low Power Schottky' family. Generally, the digits after the

family designator indicates the function ('04 for inverters, '08 for 2-input AND gates, $\cdots$). As new logic families were produced, their DC characteristics were often kept compatible with previous families. Table 2.3 shows the DC characteristics of three different logic families derived from TTL.

Table 2.3: Selected Logic Families' DC Characteristics

Logic Family	V_{CC}	V_{IH}	I_{IH}	V_{IL}	I_{IL}	V_{OH}	I_{OH}	V_{OL}	I_{OL}
TTL 74xx	4.75-5.25V	2V	40μA	0.8V	-1.6mA	2.4V	-0.4mA	0.4V	16mA
TTL 74Sxx	4.75-5.25V	2V	50μA	0.8V	-2.0mA	2.7V	-1.0mA	0.5V	20mA
TTL 74LSxx	4.75-5.25V	2V	20μA	0.8V	-0.4mA	2.7V	-0.4mA	0.5V	8mA

In addition to TTL logic families, CMOS logic families with similar logical operation but different power consumption and performance characteristics also exist. The interested reader is referred to Wikipedia on the WWW for a more detailed history and description of the various logic families created in the past and currently in use. Once there, search for 'logic families'.

2.3.2 I/O Standards

Today, few digital systems are built using TTL-like logic families of parts. Up to this point in this chapter, however, TTL data sheets have been used to help introduce the concepts associated with DC interfacing. Instead, many digital systems today are based around custom-designed integrated circuits or standard components like FPGA's, CPU's, and memories. In this case there is not the precise notion of a logic family to ensure interface compatibility. Rather, I/O standards have been created for this purpose.

An example of such an I/O standard is the LVTTL/LVCMOS standard and was created by the JEDEC standards body. Formally, this standard is called *Interface Standard for Nominal 3V/3.3V Supply Digital Integrated Circuits*. It is also known as JEDEC standard JESD8-B and, at the time of this writing, could be found on the WWW at *http://www.jedec.org/download/search/jesd8b.pdf*. It defines two DC interface standards specifically for $3.3V$ circuits: LVTTL and LVCMOS. The term LVTTL stands for *Low Voltage TTL* while the term LVCMOS stands for *Low Voltage CMOS*. These were designed to provide input and output voltages and currents compatible with $5V$ TTL parts or CMOS parts.

Table 2.4 summarizes the DC characteristics for LVTTL[2]. Note that the standard specifies *both* max and min values for V_{IH} and V_{IL}. Also note that, unlike TTL, it specifies no independent value for I_{OH} or I_{OL}. Rather, output current values are given as test conditions for V_{OH} and V_{OL}. For example, $V_{OH} = 2.4V$ but only if the output current is less than or equal to $2mA$.

Table 2.4: LVTTL DC Specifications

	Parameter	**Test Condition**	**min**	**max**	**units**
V_{DD}	Power supply voltage		3.0	3.6	V
V_{IH}	High-level input voltage		2	$V_{DD} + 0.3$	V
V_{IL}	Low-level input voltage		-0.3	0.8	V
I_{IN}	Input current	$V_{IN} = 0V$ or $V_{IN} = V_{DD}$		+/-5μ	A
V_{OH}	High-level output voltage	V_{DD} = min, I_{OH} = -2mA	2.4		V
V_{OL}	Low-level output voltage	V_{DD} = min, I_{OL} = 2mA		0.4	V

The same document that describes LVTTL also describes a standard called LVCMOS, which is summarized in Table 2.5. The power supply, input voltage, and input current requirements are identical for LVTTL and LVCMOS.

[2] V_{DD} as used in the specification is the power supply voltage. While the term V_{CC} is typically used for bipolar circuits, V_{DD} is commonly used for MOS.

Table 2.5: LVCMOS DC Specifications

	Parameter	Test Condition	min	max	units
V_{DD}	Power supply voltage		3.0	3.6	V
V_{IH}	High-level input voltage		2	$V_{DD} + 0.3$	V
V_{IL}	Low-level input voltage		-0.3	0.8	V
I_{IN}	Input current	$V_{IN} = 0V$ or $V_{IN} = V_{DD}$		+/-5μ	A
V_{OH}	High-level output voltage	V_{DD} = min, I_{OH} = -100μA	$V_{DD} - 0.2$		V
V_{OL}	Low-level output voltage	V_{DD} = min, I_{OL} = 100μA		0.2	V

The two major differences between LVTTL and LVCMOS include:

1. A larger output voltage swing is specified – to within $0.2V$ of both ground and V_{DD}. This is typical of the voltage swing for CMOS circuits.

2. The output current specification for V_{OH} and V_{OL} is very small – only 100μA. This is also consistent with typical CMOS circuits – a CMOS circuit presents a load that is mainly capacitive in nature. Thus, at steady state a CMOS circuit will have $I_{IN} = 0$.

When dealing with an I/O standard such as LVTTL or LVCMOS, interface feasibility analysis is done in the same way as shown in previous sections. As above, it consists of two parts: input/output voltage analysis and input/output current analysis.

Following the creation of the LVTTL and LVCMOS standards, there was a desire to further reduce power supply levels in integrated circuits. This became especially important in the 1990's as battery-powered mobile devices became more and more common and battery life became an important issue. As a result, similar standards for $2.5V$ and $1.8V$ operation were also created and documented as standards. Finally, in addition to these simple standards, a wide range of other I/O standards also have been created for other types of signalling such as differential I/O signalling, etc.

2.4 Noise Margins

Consider the scenario of an LVTTL circuit driving a conventional TTL circuit. The value of V_{OH} for the LVTTL circuit is $2V$, while the TTL circuit requires an input of at least $2V$. One could conclude that, as far as logic '1' values are concerned, this interfacing scenario is acceptable but barely so. However, the presence of noise in the system brings such a conclusion into question.

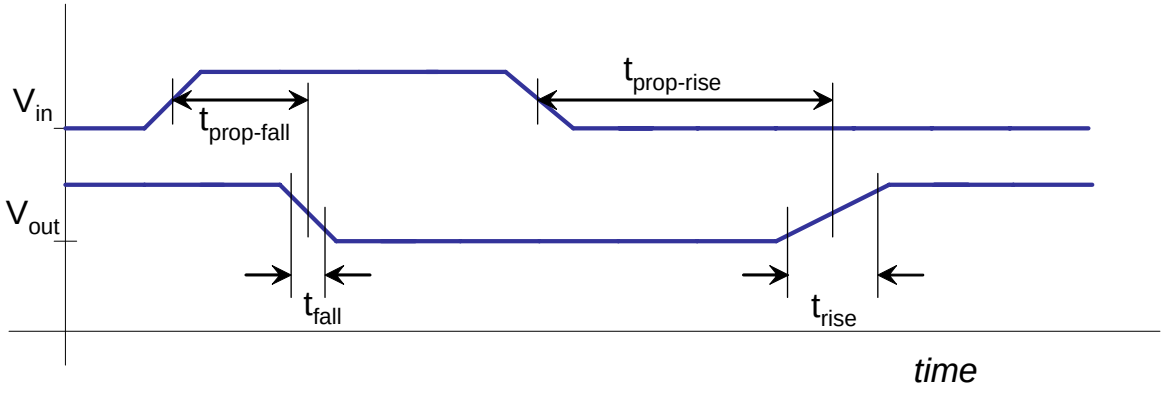
What is noise? Noise in this context refers to unwanted voltages on the signal wires in the system. As a simple example, consider the scenario where two wires in a circuit run physically parallel across the circuit for a long distance. Due to the physical arrangement of the wires, an unwanted capacitor structure is formed between them. This *capacitive coupling* between the two wires means that as the voltage on one wire changes, the voltage on the other wire may be affected. Thus, the nominal $2.0V$ value on the second wire may fall below $2.0V$, an invalid logic '1' value, and the circuit may not function correctly. In general, such a voltage deviation would be a *transient* effect, meaning that the gate driving the affected wire would be able to drive it back up to $2.0V$ within a short time. However, in the mean time, gates being driven by that wire may switch, leading to a circuit malfunction. This is just one example of noise in a system.

In Figure 2.4, however, one can see that there is a gap (or margin) between V_{OH} and V_{IH} of $0.4V$. That is, $V_{OH} = 2.4V$ for the driving inverter and $V_{IH} = 2.0V$ for the inverter being driven. In the capacitive coupling scenario just described, as long as the capacitvely induced noise on the wires was less than 0.4, there is no possibility of a circuit malfunction. Thus, for the circuit of Figure 2.4, one could say that "the high noise margin is $0.4V$". By similar reasoning, the low noise margin in Figure 2.4 is also $0.4V$ ($V_{OL} = 0.4V$ and $V_{IL} = 0.8V$). While determining the exact amount of noise margin desirable for a digital system is beyond the scope of what we will cover in this textbook, when performing an interfacing feasibility analysis you should be aware of the concepts of noise and noise margins.

2.5 Timing Characteristics of Logic Circuits

Another important consideration of gate behavior has to do with how *fast* a gate will react to changes on its inputs. Knowing the *delays* associated with the gates in your design allows you to determine how fast the design will run. That is the topic of this section.

Figure 2.7 shows a stylized representation of the *timing* behavior of a CMOS inverter. When the input (V_{in}) rises, the output (V_{out}) falls, and when the input falls the output rises. The drawing reflects *how long* it takes for the output to respond to changes on the input rather than final DC values.



$t_{prop\text{-}fall}$ and $t_{prop\text{-}rise}$ are measured from 50% of input swing to 50% of output swing

t_{rise} is measured from 10% of output swing to 90% of output swing
t_{fall} is measured from 90% of output swing to 10% of output swing

Figure 2.7: Detailed Timing Characteristics of an Inverter

There are two delays of interest in this drawing. The first is called *propagation delay* or t_{prop}. It is often defined as the time from when the input reaches the middle of its input swing to when the output reaches the middle of its output swing. Designers normally call this the *delay* associated with the gate. In the figure, $t_{prop-fall}$ is the propagation delay associated with a falling transition on the output, and $t_{prop-rise}$ is the propagation delay associated with a rising transition on the output. Note that $t_{prop-rise}$ is not necessarily the same as $t_{prop-fall}$. This is generally true for most gates — their rising and falling delays are different due to how they are constructed from silicon.

The figure also shows another set of delays called *rise time* and *fall time*. Rise time (t_{rise}) is typically defined as the time required for the output to go from 10% of its output swing to 90% of its output swing. Fall time (t_{fall}) is defined similarly. For some purposes, rise and fall times are important quantities in addition to propagation delays. Often, they are not needed (or even provided in a data sheet).

Table 2.6 summarizes the delay characteristics of a TTL 7404 inverter from a typical TTL data sheet and Figure 2.8 shows the corresponding timing waveforms. As can be seen, t_{PLH} in the table and figure is equivalent to $t_{prop-fall}$ from Figure 2.7 while t_{PHL} is equivalent to $t_{prop-rise}$. Further note that these propagation delays are measured from the $1.5V$ point in the input and output waveforms rather than precisely from the 50% points.

Table 2.6: Switching Characteristics of TTL 7404 Inverter

Parameter	Test Conditions	min	typ	max	units
t_{PLH}	R_L = 400 Ω, C_L = 15pF		12	22	ns
t_{PHL}	R_L = 400 Ω, C_L = 15pF		8	15	ns

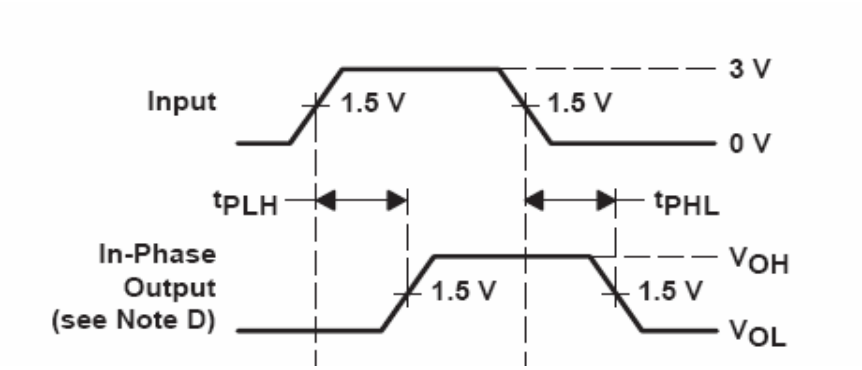


Figure 2.8: Inverter Timing Waveforms

2.5.1 Loading-Dependent Delay

An important point about the delays given in Table 2.6 is that specific test conditions were provided. That is, the gate's delay was measured when the gate was driving into a load containing both a resistance as well as a capacitance. Figure 2.9 shows the load used when measuring the delays of Table 2.6. The arrangement of capacitor, resistor, and diodes in this circuit is meant to approximate the kind of TTL load this circuit might encounter when used in typical designs.

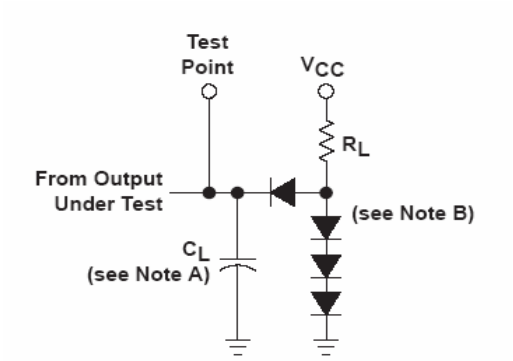


Figure 2.9: Load Circuit for Delays of Table 2.6

In contrast, Figure 2.10 shows the load circuit used for testing the delay of a recent DDR RAM device. The take-away information from these two examples is that delay values are *always* dependent on the attached load, and a proper interfacing analysis will take these load conditions into account.

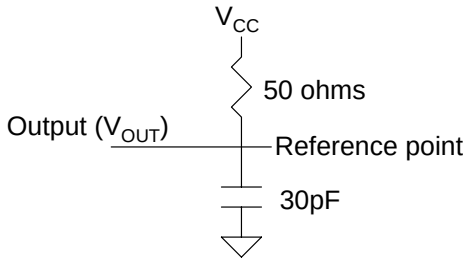


Figure 2.10: Load Circuit for Delays Testing of a DDR RAM

2.5.2 Capacitive vs. Resistive Loads

In Section 2.2 we saw that a TTL gate can only drive 10 similar gates. For example, $I_{OL} = 16mA$ for the driving gate and $I_{IL} = 1.6mA$ for the gate being driven. Thus, the maximum fanout of 10 for TTL devices is determined based

on current considerations. In this case, we say that a TTL gate presents a *resistive load* to the device trying to drive it. Put another way, the DC input current for a TTL gate is non-zero. If you exceed the fanout capacity for TTL, the gates will simply not work as intended — it is not a function of them working more slowly. Rather, they simply will not work at all.

In contrast, a CMOS gate presents an almost purely capacitive load to the gate attempting to drive it. Figure 2.11 shows a transistor-level diagram of a CMOS inverter driving another CMOS inverter. It also presents two equivalent circuits. One is for the case when $V_{IN} = 3.3V$ and therefore the first inverter's output is falling. The other is for the case when $V_{IN} = 0V$ and the first inverter's output is rising.

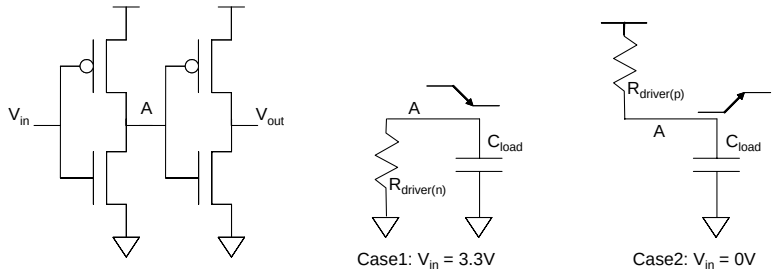


Figure 2.11: CMOS Inverter Driving Another CMOS Inverter and Equivalent Circuits

As can be seen, when a MOS FET is on it can be modelled roughly as a resistor and when a MOS FET is off it can be modelled roughly as an open circuit.

The delay on node 'A' can be computed approximately based on $R_{driver} \times C_{load}$ as in introductory circuits courses. The problem with this approach is that R_{driver} is not a constant value. For Case 1, when V_A is large, the effective R_{driver} is much smaller than when V_A approaches $0V$. Thus, an accurate calculation of the delay through this circuit is not possible using a simple RC model. Instead, a detailed circuit simulator such as SPICE is commonly used to determine delays.

However, the RC model does have some use for developing a qualitative understanding of delay vs. loading in CMOS circuits. For example, what would you expect the delay on the output of the first inverter in Figure 2.12 to be compared to that in Figure 2.11?

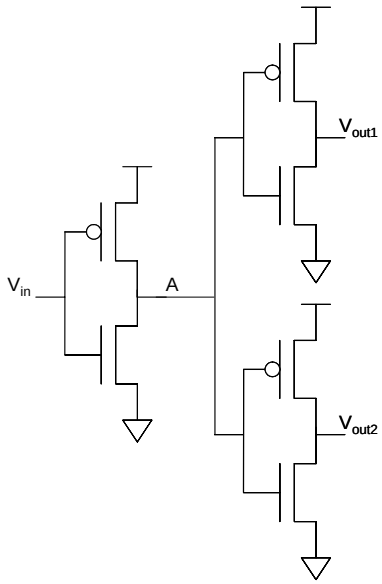


Figure 2.12: CMOS Loading-Dependent Delay Example

Intuition suggests that C_{load} for this second circuit is twice that of the original circuit (we may assume it is based roughly on how many transistor gates the inverter *drives*). Further, we know that the delay is dependent on $R_{driver} \times C_{load}$. Thus, doubling C_{load} would seem to double the delay on node 'A'.

So, what is the limit to how many CMOS loads a gate can drive? The numbers of Table 2.5 would indicate that, based on current considerations, a fanout of at least 20 is achievable (output current is 100μA, gate input current is less than 5μA. However, the intuitive RC analysis just presented would indicate that the delay on node 'A' in this circuit would be approximately 20 times that of Figure 2.11. In many cases, this delay will exceed system clocking requirements. Thus, it may not be feasible *from a timing viewpoint* to build such a large fanout into a design.

In summary, some loads are resistive, meaning their DC input current is non-zero. For these loads, DC current considerations often dictate allowable fanout. In addition to TTL gates, examples of resistive loads include LEDs and some kinds of relays and solenoids.

Other loads (especially CMOS loads) are mostly capacitive, meaning that their DC input current is close to zero. For these kinds of loads the limiting factor to fanout is usually not DC current considerations but rather timing considerations. Indiscriminately increasing the number of loads a CMOS gate drives will lead to circuits that run very slowly[3]. A range of CMOS logic families exist which are example of capacitive-load circuits. In addition, the majority of modern microprocessors, memories, and FPGAs are all based on CMOS technology. Finally, most custom integrated circuits are developed using CMOS technology. In these cases, design-for-speed is based on a careful analysis of the loading of each gate's outputs. The use of CAD tools to perform such timing analyses is the rule and such analyses are almost always performed immediately after synthesis and physical design (placement and routing) of the circuit.

2.6 A Final Note

The circuit course prerequisite for this chapter is basic linear circuits (voltage, current, R's, L's, and C's). As such, many details on the electrical properties of circuits were necessarily glossed over. For example, noise due to capacitive-coupling (crosstalk) was mentioned to motivate one source of noise in a circuit, but little else was said about crosstalk. A whole host of other important electrical effects were not even mentioned at all. For example, another source of noise in electrical systems occurs you consider wire inductance in conjunction with rapidly changing currents (as found in switching gate outputs). The resulting $V = L \times \frac{dI}{dt}$ effects are called ground bounce or rail collapse when the voltage change appears the power or ground leads. A somewhat related effect is known as ringing.

The very simple timing models discussed in this chapter further assumed that all wires were perfect (zero delay) conductors. In reality, all wires introduce delay on the signals they transmit. Two decades ago in the days of TTL this was largely ignored because the circuits were switching at such slow rates. With integrated circuits being clocked well over 1 GHz today, wire delays have become as significant as logic delays in the operation of circuits. Further, it is often the case that wires can no longer be modelled as simple bulk delays. At high switching speeds, *all* wires in a circuit behave like transmission lines to a certain effect due to their intrinsic R, L, and C characteristics.

A conscious decision was made in the writing of this textbook to limit the discussion to what would be appropriate for students having only 'Digital Design 1' and 'Circuits 1' courses as preparation. To fully understand many of these more advanced effects requires additional courses in electronics, electromagnetics (especially transmission lines), and integrated circuit technology. As such, these advanced concepts have been left for another text to cover.

[3] For example, a student project from one of the author's past classes ignored this and featured a single gate output driving 225 different subcircuits! Needless to say, the resulting circuit did *not* meet timing.

2.7 Chapter Summary

The key high-level points to master from this chapter include the following:

1. Binary logic values are actually voltages taken from a continuous range.
2. Both DC voltage and current considerations are used to determine whether two circuits can be successfully interfaced to one another.
3. Logic families were created to simplify interfacing.
4. Data sheets typically give all the information required to do DC interfacing feasibility analyses.
5. Noise is unwanted voltages on wires in electronic systems. Noise margins ensure that circuits will function correctly in the presence of limited amounts of noise.
6. Maximum fanout is defined as the number of gates a single gate can drive.
7. DC current considerations often limit the fanout of a logic gate. This especially true for resistive loads. Both DC current and timing considerations must usually be taken into account when determining fanout limits for gates driving mostly capacitive loads.
8. Output voltages, currents, and delays are always specfied in a data sheet in conjunction with the test conditions used. These test conditions specify, among other things, the power supply voltage used and the load used to perform the measurement.

The skills that should result from a study of this chapter include the following:

1. The ability to understand the DC and AC portions of a logic circuit data sheet.
2. The ability to perform a DC interfacing feasibility analysis based on both voltage and current considerations.
3. The ability to compute fanout limits for a circuit.
4. The ability to determine delays through a circuit from data sheet quantities.