

FED 101: Computer Engineering Module

Version 1.3

Dr. Sol Rosenstark

Department of Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, New Jersey

© 2001
New Jersey Institute of Technology
All rights reserved

Acknowledgements

I wish to express my gratitude to Dr. Joseph Frank and Dr. John Carpinelli of the Electrical and Computer Engineering Department at NJIT for proofreading this manual and making many useful suggestions.

Contents

Acknowledgements	i
Course Organization	iii
1 Introduction to Various Number Systems	1
Objectives	1
The Positional Number Notation	1
Conclusion	6
2 Introduction to Electronics	7
Objectives	7
Basic Electrical Concepts — Ohm’s Law	7
The Light Emitting Diode	10
The Integrated Circuit and the Inverter	12
The AND, the OR and the XOR Gates	14
Conclusion	18
3 Design and Construction of Some Simple Logic Circuits	19
Objectives	19
The Half Adder	19
The Full Adder	20
The Full Adder	22
Conclusion	24
4 Design and Construction of a Decimal Counter	25
Objectives	25
The 7-segment Display and Display Driver	25
The BCD to 7-segment Decoder Chip	26
The dual BCD Counter Chip	28
Conclusion	29

Course Organization

In this short course the student will be presented with some fundamental concepts relating to computer architecture. These concepts will be explored at an introductory level with the idea of broadening the students' background so they have a conception of what computer engineering entails. This should help them decide if this is a major they would wish to pursue as a career option.

All the theoretical material which the students are required to understand will be presented in the classroom along with practice exercises. The students will be familiarized with some engineering applications of the theory. This will lead to the design of some simple digital circuits. The efficacy of the designs will be verified through experiments in the laboratory. This interactive learning technique should help the students assimilate the material rapidly and without any great difficulty.

The material covered will be:

1. A review of the decimal positional number notation and its meaning in terms of powers of ten.
2. Introduction of the octal number system.
3. Introduction of the binary number system.
4. Introduction of the hexadecimal number system.
5. Conversion of numbers between the various number bases.
6. Review of the addition of decimal numbers and how this can be extended to binary numbers.
7. An introduction to the concepts of voltage, current, resistance and Ohm's law. Some experiments will be performed to help the students assimilate these concepts.
8. Introduction to the idea of using 0 and 5 volts as representations of TRUE and FALSE as well as the binary system's 0 and 1. Explanation of the light emitting diode (LED) and its application as a logic indicator.

9. Introduction to the concepts of Boolean logic: inversion (complements), AND, OR and exclusive or (XOR).
10. Introduction to digital logic integrated circuits (ICs) starting with the inverter and continuing to 2-input AND gates, 2-input OR gates and 2-input XOR gates. Some time will be spent performing some experimentation to become familiar with the use of the ICs.
11. Creation of the truth tables for the binary half adder and also for the full adder.
12. The students will design a half adder. This will be constructed and verified.
13. The students will design a full adder, then proceed to construct and verify it.
14. The students will now turn their attention to a 7-segment display.
15. Which they will learn to interface to a binary coded decimal (BCD) to 7-segment display driver.
16. This will in turn be interfaced with a 2-digit BCD counter.
17. The above concepts will be related to the idea of counting goods on a production line.

The laboratory will be provided with the equipment necessary to conduct all the experiments.

Chapter 1

Introduction to Various Number Systems

Objectives

- Review the decimal positional number notation and its meaning in terms of powers of ten.
- Introduce the octal number system.
- Introduce the binary number system.
- Introduce the hexadecimal number system.
- Learn to convert numbers between the various number bases.
- Learn to add two binary numbers.

The Positional Number Notation

The decimal number system uses the digits from 0 to 9 to represent numbers of any desired magnitude. The position of the digits from right to left determines their contribution to the overall number. (To facilitate our task our attention will be confined to positive integers only.) It should come as no great surprise that the interpretation of the decimal number 98234 is

$$98234 = 9 \times 10000 + 8 \times 1000 + 2 \times 100 + 3 \times 10 + 4 \times 1 \quad (1.1)$$

Using powers of 10 we get the more useful interpretation

$$98234 = 9 \times 10^4 + 8 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 \quad (1.2)$$

The Octal Number System

The only reason the decimal number system is so popular is that humanity has ten fingers. Had we possessed 8 fingers then we would no doubt be using the octal system which uses 8 as its base. Had we been born with 16 fingers cluttering our hands, we might have been using a hexadecimal number system which uses 16 as its base. So as not to confuse numbers in different systems we will append a decimal number with a D, an octal number with a Q, a binary number with a B and a hexadecimal number with an H.

Observe that decimal numbers are represented by the digits 0 to 9. The power base used for decimal numbers is 10. In the octal number system the numbers are represented by the digits 0 to 7 and the power base used is 8. For example, to find the decimal value of the octal number 3476 Q, we write

$$3476 \text{ Q} = 3 \times 8^3 + 4 \times 8^2 + 7 \times 8^1 + 6 \times 8^0 = 1854 \text{ D} \quad (1.3)$$

Exercise 1.1 — Convert the following octal numbers to decimal form by following the calculations demonstrated in (1.3).

- (a) 246 Q (b) 751 Q (c) 45063 Q (d) 32754 Q (e) 452207 Q ■

The Binary Number System

Numbers are stored in computers in a binary (base 2) system. This is because all data in computers is stored as strings of zeros and ones. In a base 2 system the numbers are represented by the digits 0 and 1 and the power base used is 2. For example, to find the decimal number value of the binary number 110111010101 B, we write

$$\begin{aligned} 110111010101 \text{ B} &= 1 \times 2^{11} + 1 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 \\ &\quad + 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 \\ &\quad + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 3541 \text{ D} \end{aligned} \quad (1.4)$$

It is helpful to have a tabulation of powers of 2, such as the one shown in table 1.1, when doing the binary to decimal conversions.

Exercise 1.2 — Convert the binary numbers below to decimal form by following the calculations demonstrated in (1.4) and taking advantage of table 1.1.

- (a) 1101110 B (b) 11101 B (c) 1010101010 B (d) 101111101 B (e) 11110111 B ■

Table 1.1: Powers of the number 2.

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536

The Hexadecimal Number System

As mentioned, the binary number system is used in digital computers. But it is readily apparent that the notation for binary numbers is awkward in that even small numbers are represented by long strings of zeros and ones. Over the years two closely related number systems were used to shorten the representation of the numbers. One was the octal system, discussed previously, which is now largely out of favor. The other system, which is alive and well, is the base 16 or hexadecimal number system.

Unfortunately the digits 0 to 9 do not suffice to represent hexadecimal numbers. We need single symbols to represent digits from 0 to 15. Table 1.2 gives the representational system in current use.

Now that we have the notation for the hexadecimal digits fixed in table 1.2, we can convert numbers from hexadecimal to decimal format without any difficulty. For example, to find the decimal number value of the hexadecimal number 4AF8H, we write

$$4AF8H = 4 \times 16^3 + 10 \times 16^2 + 15 \times 16^1 + 8 \times 16^0 = 19192D \quad (1.5)$$

Exercise 1.3 — To get some practice with the above procedure, convert the following numbers from hexadecimal to decimal notation.

- (a) 642H (b) 75BH (c) 5E0H (d) 1FFFH (e) 1000H ■

Table 1.2: The digits used in a hexadecimal system and their decimal and binary equivalents.

Decimal	Hex	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Conversion between Hexadecimal and Binary Numbers

Conversion from hexadecimal to binary notation is so easy that the two systems are almost interchangeable. To convert a number from hexadecimal to binary notation all you need to do is consult table 1.2 to find the binary equivalent that replaces each hexadecimal digit. Below is the number from (1.5) in binary form.

$$4AF8H = \overset{4}{\underbrace{0100}} \overset{A}{\underbrace{1010}} \overset{F}{\underbrace{1111}} \overset{8}{\underbrace{1000}} = 010010101111000B \quad (1.6)$$

Exercise 1.4 — Try your hand at the above procedure by converting the following numbers from hexadecimal to binary notation.

- (a) 642H (b) 75BH (c) 5E0H (d) 1FFFH (e) 1000H ■

Converting from binary to hexadecimal is a repetition of the above process in reverse order. Simply arrange the binary digits (also referred to as bits) into groups of four (called nibbles), starting on the right. Then consult table 1.2 to find the hexadecimal equivalent of each nibble. Below is an illustration of this method.

$$10111011000111B = \overset{5}{\underbrace{101}} \overset{E}{\underbrace{1110}} \overset{C}{\underbrace{1100}} \overset{7}{\underbrace{0111}} = 5EC7H \quad (1.7)$$

Exercise 1.5 — Convert the binary numbers below to hexadecimal form by using the procedure demonstrated in (1.7) and taking advantage of table 1.2.

(a) 1101110B (b) 11101B (c) 1010101010B (d) 101111101B (e) 11110111B ■

Conversion from Decimal to Hexadecimal Numbers

We have seen that once numbers are in hexadecimal form it is no great feat to convert them to binary form and vice versa. So the problem is to convert numbers from decimal to hexadecimal form since the hexadecimal notation is much more compact than the binary. The easiest and most straightforward system is to use successive division as shown below. We have chosen for our illustration the number 102973D.

$$\begin{aligned}
 102973 \div 16 &= 6435 \text{ rem } 13 \\
 6435 \div 16 &= 402 \text{ rem } 3 \\
 402 \div 16 &= 25 \text{ rem } 2 \\
 25 \div 16 &= 1 \text{ rem } 9 \\
 1 \div 16 &= 0 \text{ rem } 1
 \end{aligned} \tag{1.8}$$

The remainders on the right, read from bottom to top in hexadecimal form, are the desired result. We conclude that

$$102973 \text{ D} = 1923 \text{ D H} \tag{1.9}$$

Exercise 1.6 — Convert the decimal numbers below to hexadecimal form by using the procedure illustrated in (1.8) and (1.9).

(a) 512 D (b) 4702 D (c) 65933 D (d) 97255 D (e) 109678 D ■

Addition of Binary Numbers

One last item that we need to discuss before moving on to other topics is the addition of two binary numbers. The method is actually no different than the familiar addition of two decimal numbers. Something can be learned from the example below.

$$\begin{array}{r}
 \text{carry} \quad 11 \ 1 \\
 \quad \quad \quad 357346 \\
 + \quad \quad \underline{475637} \\
 \text{result} = \quad 832983
 \end{array} \tag{1.10}$$

When we perform the addition for each digit, we observe that if the sum of the addition exceeds 9, the largest digit in the decimal system, we carry a one

(1) over into the next digit position. We observe additionally that the carry can never be greater than unity.

The same principles are applied when we add two binary numbers. It is just a question of knowing that for binary numbers $0 + 0 = 0$, $0 + 1 = 1 + 0 = 1$, $1 + 1 = 10$ and $1 + 1 + 1 = 11$. Since the last two results contain two digits then a carry is generated. As an example

$$\begin{array}{r}
 \text{carry} \quad 111111 \\
 \quad \quad 10111101 \\
 + \quad \underline{11110110} \\
 \text{result} = 1101101011 \qquad (1.11)
 \end{array}$$

Exercise 1.7 — Perform the binary additions indicated below and verify by converting all the numbers to decimal form.

$$\begin{array}{r}
 \text{(a)} \quad 1011 \quad \text{(b)} \quad 1101101 \quad \text{(c)} \quad 11110011 \quad \text{(d)} \quad 10110111 \\
 + \underline{111} \quad \quad + \underline{1011001} \quad \quad + \underline{10111011} \quad \quad + \underline{11101110}
 \end{array}$$

■

Conclusion

We have used the material in this section to become familiar with the hexadecimal and binary number systems used in digital computers. In addition we have learned how to convert numbers between the various number systems. Addition of binary numbers was discussed for the specific reason that one of the design objectives in this course will be the design of a binary adder.

Chapter 2

Introduction to Electronics

Objectives

- Get an intuitive understanding of the concepts of voltage, current and Ohm's law.
- Become familiar with the light emitting diode (LED).
- Familiarization with the inverter gate and the 7404 integrated circuit.
- To learn the logical concepts of AND, OR and XOR.
- To become familiar with the integrated circuits associated with the above logic functions.

Basic Electrical Concepts — Ohm's Law

There are many levels of understanding of electrical phenomena. We will simply try to get a rudimentary understanding, leaving deeper studies for later courses.

The idea of *current* is quite easy to grasp. It represents the amount of positive electrical charge per unit time passing some point of observation. Current is measured in units of amperes, the letter A being used to designate that unit. If the glove compartment lamp in a car is a type 194 then, when it is on, it *draws* a current of 0.27 A. This means that there is a charge of 0.27 Coulombs per second entering one terminal and leaving the other. Current passes *through* electrical devices. The question then is: "What is it that makes the current pass through the devices?" The response to that question is — *voltage*.

To make current pass through devices we need to have an electrical tension, namely voltage, across its terminals. Voltage is measured in units of volts, the letter V being used to designate that unit. Voltage is somewhat analogous to

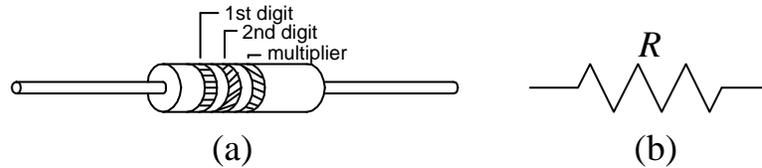


Figure 2.1: A resistor (a), and its schematic symbol (b).

Table 2.1: Color code for resistors. The mnemonic, *Better Be Ready Or Your Great Big Venture Goes West*, can be helpful.

color	number
black	0
brown	1
red	2
orange	3
yellow	4
green	5
blue	6
violet	7
gray	8
white	9

pressure in a hydraulic system. For a given size pipe, it is the pressure difference across the pipe, in pounds per square inch, that determines the flow, in gallons per minute, in the pipe. The larger the pressure difference, the greater the flow in the pipe. The same can be said of electrical devices. For a given device, the current that passes through the device is related to the *voltage difference* across the terminals of the device.

What determines the current, in direct-current electrical devices, is the *resistance* measured in units of Ohms. Ohms are designated by the Greek letter Ω . There are electrical devices available called resistors. They are usually cylindrical in shape with wire conductors (leads) emerging from each end, as shown in figure 2.1. Their resistance value is determined from the color stripes on their round bodies. The stripe nearest one end of the resistor is the first digit. The second stripe represents the second digit. The third stripe is the multiplier which represents the number of zeros that follow the first two digits.

The color code used is given in table 2.1. If, for example, a resistor is marked with yellow-violet-red then its resistance is 4700Ω . This is also referred to as $4.7\text{ k}\Omega$, the 'k' standing for kilo, meaning 1000. As another example, a resistor marked orange-brown-green would have a value of 3100000Ω , or $3.1\text{ M}\Omega$. In this case the 'M' stands for Meg or one million.

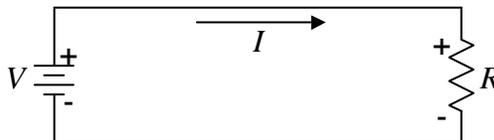


Figure 2.2: A resistor R connected to a power supply V . The voltage polarity and current direction are indicated.

Relating voltage to current in a resistor is a very simple matter. *Ohm's law* states that the current, designated by the letter I , is related to the voltage, designated by the letter V , by the proportional relation

$$I = \frac{V}{R} \quad (2.1)$$

The letter R in the above equation stands for the resistance of the device.

When the engine of a car is running, the voltage of the electrical system is approximately 14 V. A type 194 glove compartment lamp draws a current of 0.27 A. According to Ohm's law of (2.1), we determine the resistance of the type 194 lamp to be

$$R = \frac{V}{I} = \frac{14}{0.27} = 52 \Omega \quad (2.2)$$

That is the lamp's resistance when it is hot. If the lamp is allowed to cool down its resistance will change.

In the simple circuit shown in figure 2.2 we have a battery (or power supply) on the left connected to a resistor on the right. The power supply has a positive and negative terminal as marked. The current leaves the positive terminal of the power source (the battery) and reenters the negative terminal. The situation is the opposite in the resistor where the current enters the positive terminal and leaves the negative terminal as indicated by the arrow marked I .

Assume that in the circuit shown in figure 2.2 the resistor has a value of 2.2 k Ω connected across a 5 V power source. We simply use Ohm's law of (2.1) to determine that the current drawn by the resistor is

$$I = \frac{5}{2200} = 0.0023 \text{ A} = 2.3 \text{ mA}$$

In the above mA indicates milliamps or 1/1000 of an Ampere.

Exercise 2.1 — Use ohm's law to solve the following problems for the circuit of figure 2.2.

- (a) A 150 Ω resistor has a voltage of 12 V across it. Find the current through the resistor.

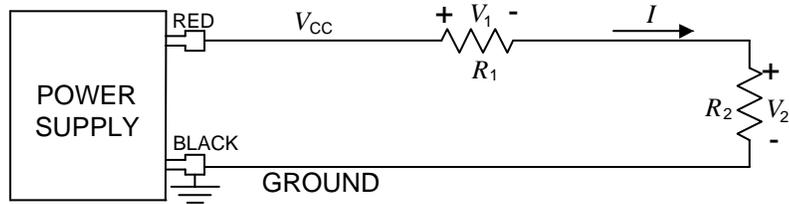


Figure 2.3: A power supply voltage connected to two resistors in series.

- (b) A $470\ \Omega$ resistor is drawing a current of $10\ \text{mA}$. Find the voltage across the resistor.
- (c) The voltage across a resistor is $5\ \text{V}$, while the current through the resistor is $15\ \text{mA}$. What is its resistance? ■

Experiment 2.1 — To get some familiarity with the above concepts connect a $1000\ \Omega$, or bigger, resistor to a power supply in the manner shown in figure 2.2. Adjust the power supply to $2.5\ \text{V}$ and verify the voltage with a voltmeter. Measure the current in the resistor. Repeat the measurement at $5\ \text{V}$, $7.5\ \text{V}$ and $10\ \text{V}$. Do the points lie on a straight line? Use Ohm's law to verify that the measurements are reasonable. ■

Experiment 2.2 — To get a further understanding of how to deal with resistive connections, connect the circuit shown in figure 2.3. Use $1000\ \Omega$ for R_1 and $270\ \Omega$ for R_2 .

Before performing any measurements, use ohm's law to determine the following:

- (a) The current I through the two series resistors by using Ohm's law.
- (b) Using the current I determined above, calculate the voltage across R_1 by using Ohm's law.
- (c) Repeat to find the voltage across R_2 .

Verify experimentally the results of the above calculations. ■

The Light Emitting Diode

A light emitting diode (LED) is shown in figure 2.4a. The diode has a flat on one side. The lead closest to the flat is the cathode. In most LEDs the cathode is also the shorter lead. The LED emits light when a current of approximately

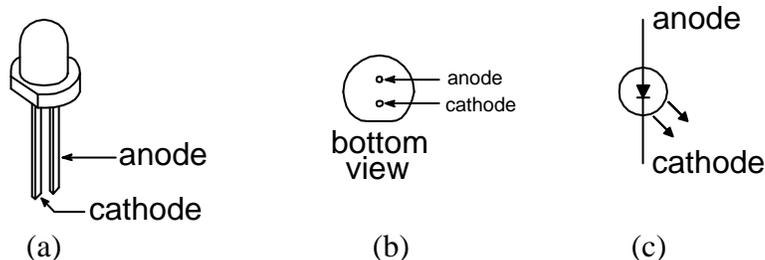


Figure 2.4: A light emitting diode (a), bottom view (b) and schematic symbol (c).

6 to 20 mA passes from anode to cathode in the direction of the arrow in figure 2.4c.

Unlike a resistor, when the commonly available red LED is conducting, it has a fixed voltage across it of approximately 1.6 V. (This may be different for different color diodes.) To control the current in an LED, a series resistor is required as shown in figure 2.5. The value of the resistor depends on the applied voltage and on the LED current that is desired.

In the circuit shown in figure 2.5 we have an LED in series with a current control resistor R_1 connected to a power supply of voltage V_{CC} . When the LED is conducting it has V_L volts across it. This leaves $V_{CC} - V_L$ volts across the resistor R_1 . Using Ohm's law we find the current I_L in the series connection to be

$$I_L = \frac{V_{CC} - V_L}{R_1} \quad (2.3)$$

Example 2.1 — In a typical case, $V_{cc} = 5\text{ V}$, $V_L = 1.6\text{ V}$ and $R_1 = 270\ \Omega$, then the LED would have a current of 12.6 mA. The LED would be much brighter if it were drawing a current of 20 mA.

Exercise 2.2 — Use (2.3) to find answers to the following:

- For example 2.1, find the resistor needed for an LED current of 20 mA.
- The LED has a series resistor of $470\ \Omega$ and the power supply voltage is 12 V. Find the current in the LED.
- The power supply voltage is 5 V and we wish to have an LED current of 10 mA. Find the value of the resistor needed in series with the LED. What will be the current if the resistor value were $270\ \Omega$?
- We have an LED in series with $270\ \Omega$. This is a suitable arrangement for a power supply voltage of 5 V. Is this the case for $V_{cc} = 12\text{ V}$?
Note: It is desired to keep the current in the LED below 20 mA. ■

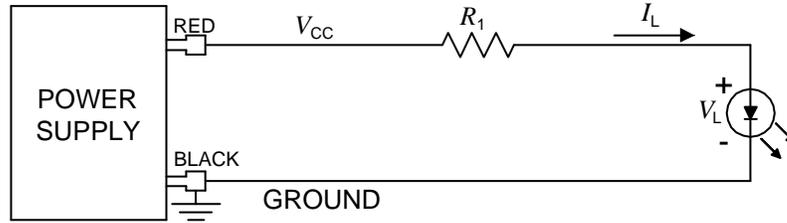


Figure 2.5: A light emitting diode in series with a current control resistor.

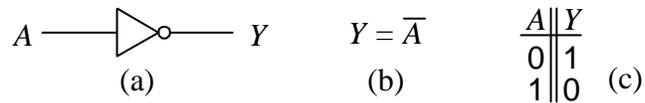


Figure 2.6: Schematic symbol for an inverter gate (a), the input-output relation (b) and the truth table (c).

Experiment 2.3 — Connect a circuit as shown in figure 2.5. Put a milliammeter in series with the LED. Vary the power supply voltage and observe how the brightness of the LED varies with the current through it. ■

LEDs have many uses. Because of their reliability they are used in rows in third brake lights in some cars. They are found as indicators in computer keyboards as well as activity lights for floppy and hard disk drives in computers. They can be used as inexpensive probes to determine the state of digital circuits.

In digital circuits, which as previously mentioned use binary arithmetic, 5 V usually represents a 1 and 0 V a 0. A simple LED in series with a resistor can be used to show if a circuit is in the 1 or the 0 state.

The Integrated Circuit and the Inverter

The device that will take up our attention at this time will be the inverter gate. A schematic form of one is shown in figure 2.6a. This device has an input A and an output Y . If we allow only two voltages at the input, 0 and 5 V, then the output will produce the complements of these voltages. That means that an input of $A = 0\text{ V}$ produces an output of $Y = 5\text{ V}$ and an input of $A = 5\text{ V}$ produces an output $Y = 0\text{ V}$. In digital computer terms a voltage of 0 V represents a binary 0 whereas a voltage of 5 V represents a binary 1.

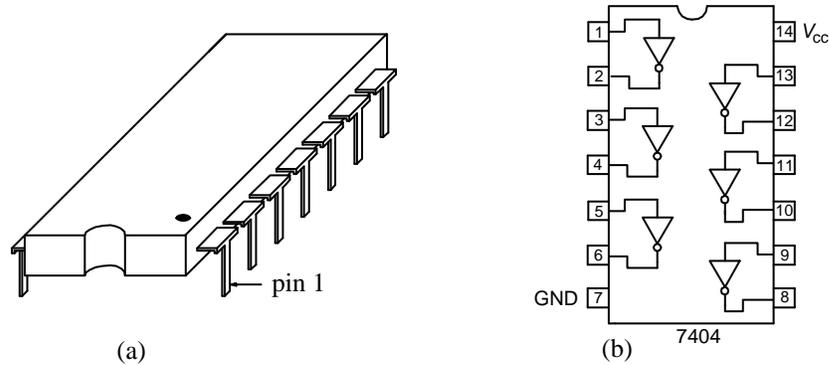


Figure 2.7: A pictorial of an IC (a) and a schematic of a hex-inverter chip (b).

The behavior of this gate can be summarized in the equation (also shown in figure 2.6b)

$$Y = \bar{A} \quad (2.4)$$

In the above, the \bar{A} indicates the logical complement of A . In Boolean logic, 0 represents a FALSE condition whereas 1 represents a TRUE condition. The complement of FALSE is TRUE and the complement of TRUE is FALSE. In binary logic terms, $\bar{0} = 1$ and $\bar{1} = 0$. The table in figure 2.6c summarizes the behavior of the inverter gate. This is known as a *truth table*.

In figure 2.7a we have a pictorial representation of an integrated circuit (IC) also known as a chip. Such a chip can contain many different kinds of electronics. Since we are interested in inverter gates, we have shown in figure 2.7b the schematic of a 7404 hex-inverter chip. This IC contains 6 inverter gates. Thus, the output at pin 2 is the complement of the input at pin 1. The other inverters behave similarly. To make this IC function properly we have to connect +5 V to pin 14, and ground to pin 7.

Experiment 2.4 — Connect an LED in series with a current control resistor from +5 V to the output of one of the gates. For this to function properly the cathode of the LED should be connected to the gate output. Connect the gate input alternately to +5 V and then to ground and observe the LED. Does the LED reflect the conditions at the input to the inverter gate? ■

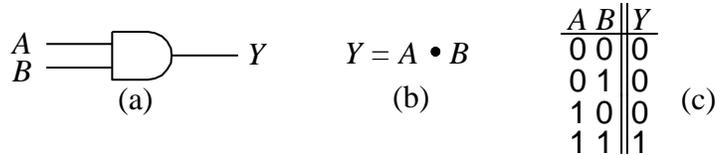


Figure 2.8: The schematic symbol for an AND gate (a), the logical AND statement (b) and the truth table (c).

The AND, the OR and the XOR Gates

At this point it is useful to become acquainted with three types of logic chips. These chips contain AND gates, OR gates and exclusive OR (namely XOR) gates.

The AND Gate

The device that will be considered next is the logical AND gate. Suppose someone makes the (simple but unromantic) statement, “I am in search of a mate that is rich *and* good-looking.” Let us assign, as mentioned earlier, a logical 1 when anything is true and a logical 0 when anything is false. We assign letters to this criterion as follows:

$$A = \text{rich} \quad B = \text{good-looking} \quad Y = \text{search} \quad (2.5)$$

The logical AND statement is written in Boolean notation (see also figure 2.8b) as

$$Y = A \cdot B \quad (2.6)$$

where the \cdot does not represent a product but a logical AND. Its effects can be explained by analyzing the original statement.

If the individual that is looking for a mate comes across someone that is not rich, then $A = 0$. If at the same time this person comes across someone that is also not good-looking then $B = 0$. The criterion is not met so the search has failed and $Y = 0$. Both criteria must be met, so that only if $A = 1$ and $B = 1$ will the search be successful and then Y will be 1. This logical ANDing of two criteria A and B is summarized in the truth table shown in figure 2.8c.

A schematic form of an AND gate is shown in figure 2.8a. This device has inputs A and B and output Y . If either one of the two inputs is low then, as seen in the truth table, the output Y will be low. The only time the output Y is high is when both inputs are high. Chips are available that have 4 AND gates with two inputs each, like the one shown in figure 2.9a. The 7408 chip is referred to as a quad 2-input AND gate IC. Again, a reminder, that to make this IC function properly we have to connect +5 V to pin 14, and ground to pin 7.

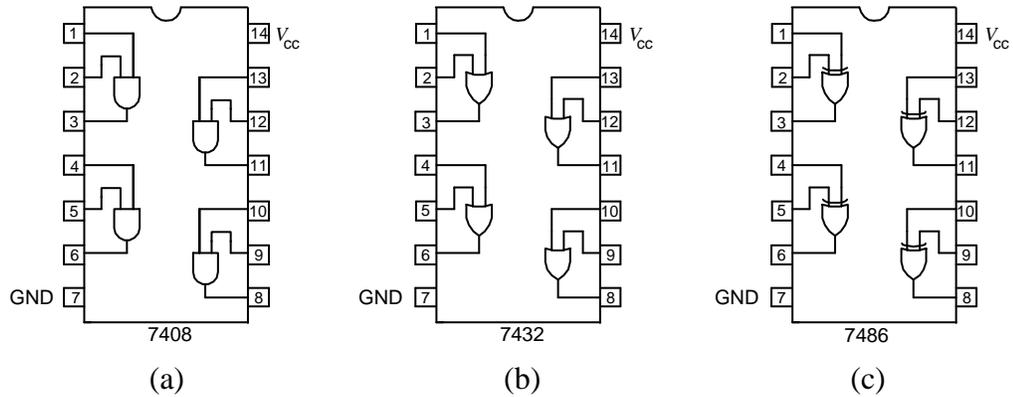


Figure 2.9: A 7408 quad 2-input AND gate (a), a 7432 quad 2-input OR gate (b), and a 7486 quad 2-input XOR gate (c).

Exercise 2.3 — Create truth tables for the Boolean statements

(a) $P = C \cdot \overline{D}$

(b) $Q = \overline{E} \cdot \overline{F}$

Experiment 2.5 — Connect +5 V and ground from a power supply to a 7408 chip. Connect an LED in series with a current control resistor from +5 V to the output of one of the gates. Verify the truth table by connecting the gate inputs to the proper voltage and observing the LED. How can this method be used to verify the proper functioning of the entire chip? ■

The OR Gate

The OR gate is another logic device that is of interest to us. We will assume that this time someone makes the far less demanding statement, “I am in search of a mate that is rich *or* good-looking.” We again assign letters to this criterion according to (2.5). The logical OR statement is written in Boolean notation (see also figure 2.10b) as

$$Y = A + B \tag{2.7}$$

where the + does not represent a sum but a logical OR.

It is clear from the new search statement that the *or* implies that the only mate that won’t be suitable will be one that is neither rich nor good-looking. In logical terms Y will be false only if A and B are both false. This explains the first line in the truth table in figure 2.10c. On the other hand if the candidate

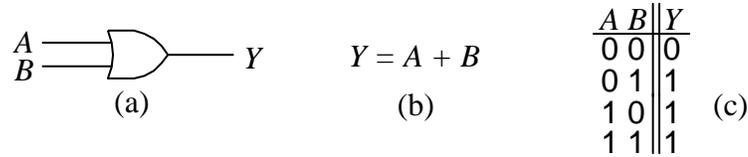


Figure 2.10: The schematic symbol for an OR gate (a), the logical OR statement (b) and the truth table (c).

for marriage is either rich or good-looking or both then the candidate is suitable and Y is then true.

A schematic form of an OR gate is shown in figure 2.10a. This device has two inputs A and B and an output Y . If both inputs are low then as seen in the truth table the output Y will be low. If, however, either input is high then the output Y is high. Chips are available possessing 4 OR gates with two inputs each, like the one shown in figure 2.9b. The 7432 chip is referred to as a quad 2-input OR gate IC. It is worth remembering that for this IC to function properly we have to connect +5V to pin 14, and ground to pin 7.

Exercise 2.4 — Create truth tables for the Boolean statements

- (a) $P = C + \bar{D}$
- (b) $Q = \bar{E} + \bar{F}$

Experiment 2.6 — Connect +5V and ground from a power supply to a 7432 chip. Connect an LED in series with a current control resistor from +5V to the output of one of the OR gates. Verify the truth table by connecting the gate inputs to the proper voltage and observing the LED. Using gates, implement and test the wiring needed to obtain the logic functions

- (a) $Y = A + B + C$
- (b) $Y = A \cdot (B + C)$

Truth tables will have to be created for the testing. ■

The XOR Gate

Another very useful device is the exclusive OR gate, or XOR gate. This time we will assume that someone makes the somewhat confusing statement, “I am in search of a mate that is rich *or* good-looking, but not both.” We again assign

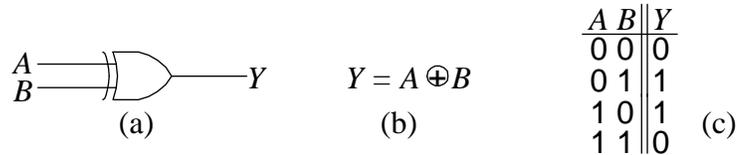


Figure 2.11: The schematic symbol for an XOR gate (a), the logical XOR statement (b) and the truth table (c).

letters to this criterion according to (2.5). The logical XOR statement is written in Boolean notation (see also figure 2.11b) as

$$Y = A \oplus B \quad (2.8)$$

where the \oplus represents the logical exclusive OR (XOR) operation.

It is clear from the new search statement that the XOR implies that the only mates that will be suitable will be ones that are either rich or good-looking. In logical terms Y will be true if A is true and B is false or vice versa. This explains the second and third lines in the truth table in figure 2.11c. If, however, the candidate for marriage is both rich and good-looking or neither rich nor good-looking then the candidate is unsuitable and Y is then false.

A schematic form of an XOR gate is shown in figure 2.11a. This device has two inputs A and B and an output Y . If both inputs are low or both are high then we see in the truth table the output Y will be low. If, however, one input is high and the other low then the output Y is high. Chips with 4 XOR gates with two inputs each are readily available, like the one shown in figure 2.9c. The 7486 chip is referred to as a quad 2-input XOR gate IC. As was cautioned for all aforementioned chips, we have to connect +5 V to pin 14, and ground to pin 7 for the IC to function properly.

Exercise 2.5 — Create truth tables for the Boolean statements

(a) $P = C \oplus \overline{D}$

(b) $Q = \overline{E} \oplus \overline{F}$

Experiment 2.7 — Connect +5 V and ground from a power supply to a 7486 chip. Connect an LED in series with a current control resistor from +5 V to the output of one of the XOR gates. Verify the truth table by connecting the gate inputs to the proper voltage and observing the LED. Using gates implement and test the wiring needed to obtain the logic function $Y = A + (B \oplus C)$. A truth table will have to be created for testing purposes. ■

Conclusion

At this point the student should have an intuitive understanding of the concepts of voltage, current and Ohm's law as well as a reasonable familiarity with light emitting diodes. The student should also be acquainted with the Boolean logic concepts of complements (inversion) as well as AND, OR and XOR. In addition the student should have a good familiarity of some of the integrated circuits that implement those functions.

Chapter 3

Design and Construction of Some Simple Logic Circuits

Objectives

A general principle in engineering is to simplify the design of large projects by basing them on smaller design modules. In carrying out the tasks enumerated below, the student will become familiar with the modularity that is used in electrical and computer engineering design. The objectives will be:

- To design, construct and test a half adder.
- To design, construct and test a full adder using half adders as building blocks.
- To design a 4-bit full adder using 4 full adders as building blocks.
- Become familiar with a commercial 4-bit full adder integrated circuit.

The Half Adder

Calculators have circuitry for performing 4-bit addition. The central processing units (CPUs) of computers contain, at the very least, 8-bit adders. Some CPUs possess adders that can handle 64-bit numbers. What those adders have in common is that they are constructed of 1-bit adders. This simplifies the design problem.

A 1-bit half adder is a device that can add two one bit numbers and produce a sum and a carry. As shown in figure 3.1, the half adder has two inputs, the bits X and Y , and produces two outputs, a sum bit S and carry bit C . The truth table for a half adder is shown in table 3.1.

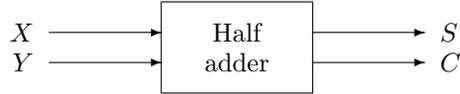


Figure 3.1: A block diagram of a half adder.

Table 3.1: Truth table for a half adder.

Inputs		Outputs	
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

To create the truth table we simply list all possible combinations of the inputs X and Y on the left, and all the resultant sums on the right. The output C is 1 only when both inputs are 1. The output S represents the least significant bit of the sum of the two inputs.

Once the truth table has been created it is no great problem to design the half adder.

Experiment 3.1 — Inspect truth table 3.1 for the half adder and compare it with the truth tables for the AND, OR and XOR gates found in chapter 2. Draw a diagram of the half adder, then construct and test it to see if it functions correctly. ■

The Full Adder

The half adder is very simple to design and build. Alas, it does not meet all the requirements for a general purpose adder. When adding binary numbers we have to consider the possibility of a *carry-in* from a less significant position. Although binary addition with carries was covered in chapter 1, we review this subject in the example below.

$$\begin{array}{r}
 \text{carry} \quad 111 \\
 \quad \quad 1011 \\
 + \quad \underline{1110} \\
 \text{result} = 11001
 \end{array} \tag{3.1}$$

We see in the above example that there is never a carry-in to the least

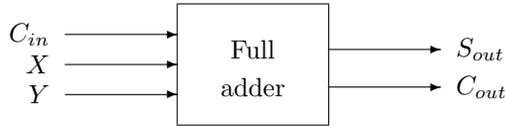


Figure 3.2: A block diagram of a full adder.

Table 3.2: Truth table for a full adder.

Inputs			Outputs	
C_{in}	X	Y	C_{out}	S_{out}
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

significant position on the right. But carry-ins are a distinct possibility for all the other positions. So a half adder doesn't quite fill the bill. We need a full adder as shown in the block diagram of figure 3.2

As shown in figure 3.2, the full adder has three inputs, the bits X and Y , as well as a carry-in bit C_{in} . It produces a sum bit S_{out} and *carry-out* bit C_{out} . The result is the sum of three numbers, which is found by first adding two numbers and then adding that sum to the third number.

The truth table for the full adder is shown in table 3.2. As we did for the half adder, we again listed all possible combinations of the inputs C_{in} , X and Y on the left, and all the resultant sums on the right. In this case, however, we cannot proceed to the design of the full adder by simply inspecting truth table 3.2. Some advanced forms of logic reductions are needed, and those are beyond the scope of this course. Accordingly the design is simply presented in figure 3.3. To convince yourself that this design indeed fulfills the requirements, simply fill in the missing elements in each line of table 3.3 to see that the full adder of figure 3.3 will produce the required results shown on the right.

Exercise 3.1 — Consult the logic diagram which was used to construct the half adder of experiment 3.1 to draw the logic diagram for a full adder. ■

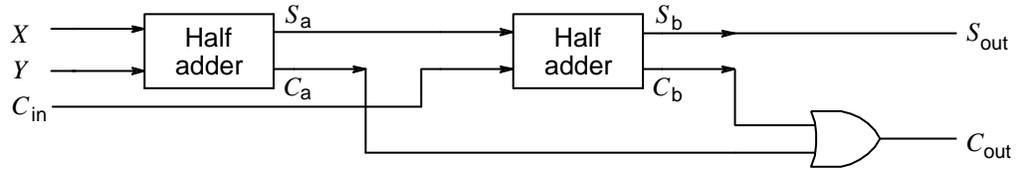


Figure 3.3: A full adder constructed from 2 half adders and an OR gate.

Table 3.3: Expanded truth table for the full adder shown in the block diagram.

Inputs			X plus Y		S_a plus C_{in}		C_a OR C_b	
C_{in}	X	Y	C_a	S_a	C_b	S_b	C_{out}	S_{out}
0	0	0					0	0
0	0	1					0	1
0	1	0					0	1
0	1	1					1	0
1	0	0					0	1
1	0	1					1	0
1	1	0					1	0
1	1	1					1	1

Experiment 3.2 — Construct the full adder by following the logic diagram of exercise 3.1. Test the full adder to verify that it meets the specifications found in table 3.2. ■

The 4-bit Adder

Complicated circuits are usually constructed from simpler ones. We have already seen how a half adder was constructed using logic gates. Then in turn we saw how to construct a full adder utilizing half adders. Those are two examples of modular design. This modular hierarchy continues in the 4-bit adder shown in figure 3.4. It is built by interconnecting 4 full adders. This hierarchical design can be continued indefinitely.

The 4-bit adder shown in figure 3.4 takes a 4-bit *augend* consisting of bits A_3, A_2, A_1, A_0 and adds it to a 4-bit *addend* consisting of bits B_3, B_2, B_1, B_0 . The bit C_1 , which is the carry-out of the rightmost full adder, is used as the carry-in to the next adder. This process is repeated for the higher order bits. We need not construct such an adder as it is commercially available in the 7483 16-pin IC, whose pin assignments are shown in figure 3.5.

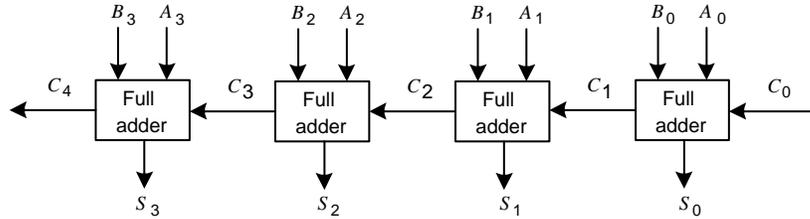


Figure 3.4: Modular design demonstrated by the construction of a 4-bit adder using 4 full adders.

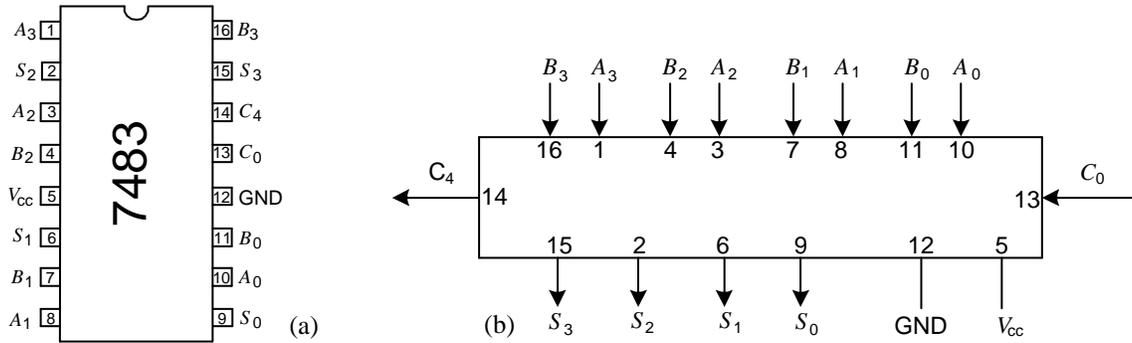


Figure 3.5: The 7483 4-bit full binary adder chip (a). The block diagram of the same IC with pin assignments (b).

Experiment 3.3 — Connect the full adder as shown in figure 3.6. Perform some tests to convince yourself that it does indeed add pairs of binary numbers and that it produces a correct carry-out. Keep this circuit long enough to answer the questions asked below. ■

Exercise 3.2 — This question relates to the experiment on the 4-bit full adder which was just performed.

- How did you connect the carry-in signal C_0 so as to be sure that your additions would produce correct results? How would this input be used if we wanted to construct an 8-bit full adder using two 4-bit full adders? Draw the block diagram based on figure 3.5.
- Suppose we need a 3-bit adder for some special purpose. The only item at our disposal is the 7483 4-bit full adder used in experiment 3.3. In this chip, the only carry signals which are accessible are the

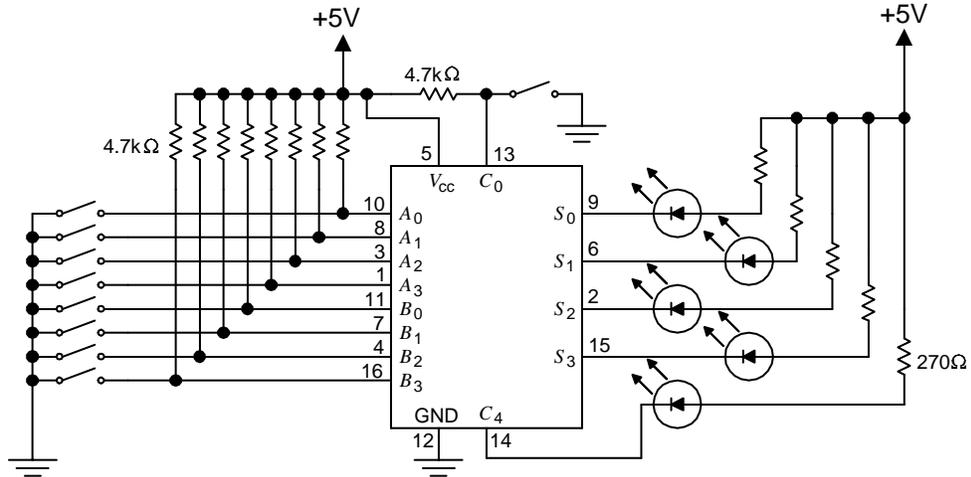


Figure 3.6: Circuit for testing a 4-bit full adder.

carry-in C_0 and the carry-out C_4 . The intermediate carry signals C_1 , C_2 and C_3 are not accessible externally.

How can the above mentioned chip be used as a 3-bit binary full adder? It is important that C_3 be visible externally. ■

Conclusion

At this point the student should have a good understanding of the modularity in the design of digital circuitry. This was acquired by first designing and testing a half adder. This background was put to good use in the design of a full adder using half adders as building blocks. The design phase was concluded when four full adders were used in the design of a 4-bit full adder. This background was further enhanced through the testing of a commercial 4-bit full adder integrated circuit.

Chapter 4

Design and Construction of a Decimal Counter

Objectives

Now that the student is familiar with basic electrical concepts and has learned the fundamentals of digital circuits it is time to apply this knowledge to the design of some interesting circuits. In this segment the student will

- Become familiar with a 7-segment display.
- Become familiar with a binary coded decimal (BCD) to 7-segment display driver.
- Become familiar with 2-digit BCD counter.
- Relate these concepts to the idea of counting goods on a production line.

The 7-segment Display and Display Driver

The 7-segment LED-display chip shown in figure 4.1a is capable of displaying numerals from 0 to 9. It can be used to create multiple digit displays of decimal numbers. Behind the seven segments labeled a . . . g are located LEDs. There is also a decimal point LED which we will ignore. All the LEDs in the HP7760 device have their cathodes connected to the common-cathode terminal CC. It is very easy to test this chip by using the circuit of figure 4.1b.

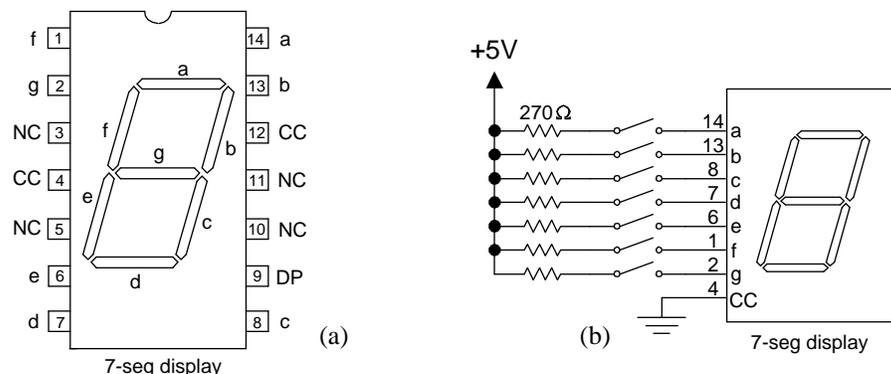


Figure 4.1: A 7-segment display chip (a) and circuit for testing it (b).

Experiment 4.1 — In the circuit shown in figure 4.1b we see that the +5 V power supply is connected to each segment through a $270\ \Omega$ current limiting resistor and switch. Wire up the circuit and using the switches observe the correspondence of the pins on the chip to the appropriate segments. ■

Clearly this is not the way the display is meant to be used. Interfacing the 7-segment display with a special driver chip will facilitate the task immensely.

The BCD to 7-segment Decoder Chip

As we have seen in table 1.2, each decimal digit can be represented by a string of 4 bits. The portion of the table of interest to us here is reproduced in table 4.1. In the table we see that each decimal digit has a binary coded decimal (BCD) representation. Thus the BCD representation of the number 2 is 0010 whereas that for the number 9 is 1001.

There are a number of ICs available for driving 7-segment displays. We will concentrate on the CD4511 BCD to 7-segment decoder chip shown in figure 4.2a. The output terminals which appear on the right side of the chip are connected directly to the corresponding terminals on the 7-segment display. On the left side of the chip we have the BCD input terminals ABCD where we can input the binary code of the digit we wish to display. Additional terminals take care of some additional functions.

When a terminal is marked with a symbol with an overbar it indicates that that terminal has to be connected to ground to make it perform its function. Such a terminal is said to be *active low*. A terminal without an overbar has to be connected to +5 V to make it perform its function. Such a terminal is said to be *active high*. The terminals \overline{LT} , \overline{BL} and \overline{LE} are active low.

Table 4.1: The binary coded decimal (BCD) encoding system.

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

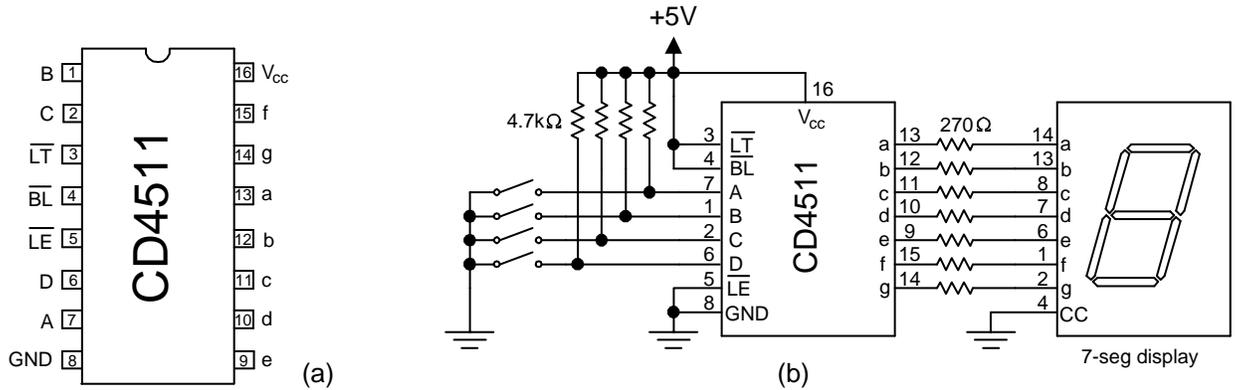


Figure 4.2: A 7-segment display chip (a) and circuit for testing it (b).

The $\overline{\text{LE}}$ is a latch-enable signal and must be low for the chip to accept, and store (latch), an input code at its ABCD terminals. If this terminal is made high then the chip retains indefinitely whatever value was latched previously.

The $\overline{\text{LT}}$ is a light-test pin which, when active, causes all the segments of the 7-segment display to light for testing purposes. This terminal must be kept high during normal operation.

The $\overline{\text{BL}}$ pin is used to blank the display. It may be used to make a flashing display by connecting it to a pulse generator. This terminal must be kept high during normal operation.

Experiment 4.2 — In the circuit shown in figure 4.2b we see that pins $\overline{\text{LT}}$ and $\overline{\text{BL}}$ are connected to +5V to make sure they will be inactive. We need to keep the latch-enable pin $\overline{\text{LE}}$ active so it is grounded. Wire up the circuit

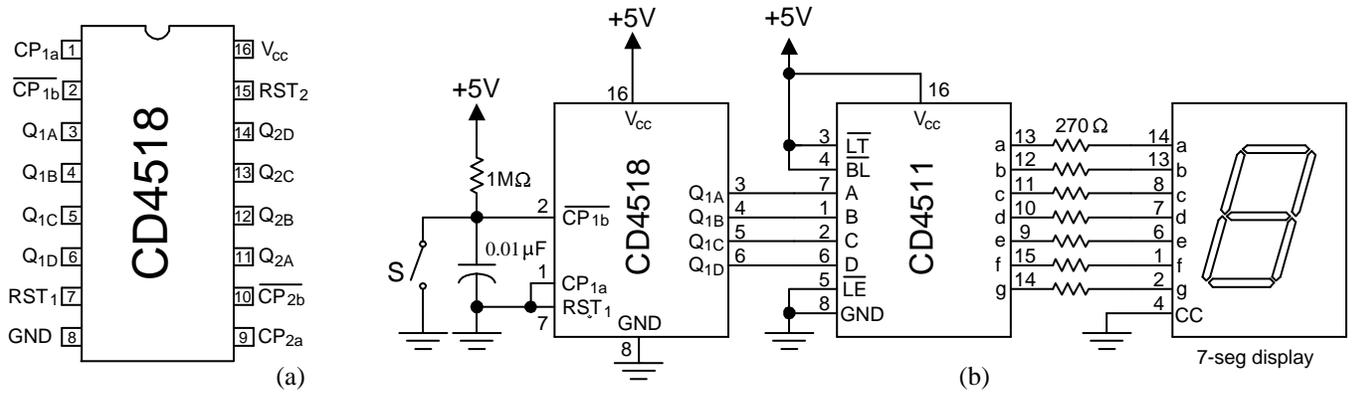


Figure 4.3: A dual BCD counter chip (a) and the complete counter circuit (b).

shown and test it to see if signals on the ABCD pins cause the proper display of numbers. ■

The dual BCD Counter Chip

It remains for us to put the chips discussed above to good use. Counting items on an assembly line might be just the thing we have in mind. For that purpose we need a BCD counter chip, such as the one shown in figure 4.3a. This chip actually contains two BCD counters, counter 1 on the left and counter 2 on the right.

The counter output consists of the bits appearing on terminals Q_{1A}, Q_{1B}, Q_{1C} and Q_{1D}. The incrementing of the counter can be accomplished by either pulsing CP_{1a} from low to high, or by pulsing CP_{1b} from high to low. If it is desired to use CP_{1b} to increment the counter, then the CP_{1a} pin must be connected to ground. Counter 1 can be reset by momentarily applying +5V to the RST₁. This terminal should be connected to ground if no resetting is desired.

Experiment 4.3 — In the circuit shown in figure 4.3b we see that the action of terminals CP_{1a} and RST₁ has been inhibited. The counter is incremented by momentarily closing switch S to pull down terminal CP_{1b}. Wire up the circuit shown and determine if the counting is proceeding properly

Exercise 4.1 — Carefully work out a scheme for a 2 decimal digit counter that is capable of counting from 0 to 99. The question really revolves around the problem of how to trigger the count on counter 2 when counter 1 goes back from 9 to 0. ■

It should be fairly clear from the answer to exercise 4.1 that the design can be extended to create a display of as many digits as are needed for any particular application.

Conclusion

The design performed in the last section of this chapter can be very easily interfaced to a detection circuit that is used to count items on a production line that pass in front of it. Such a detection circuit can consist of a light emitting diode, on one side of the items to be counted, coupled to a light detecting transistor on the other side. The fact that we only dealt with a 2 decimal digit counter is not a limitation. The design can be readily expanded to as many decimal digits as are required.