



Digital Design: Combinational Logic

An Online Continuing Education Course for Engineers

Course Number: E-5012

Credit: 5 Hours / 5 PDH / 5 CPD

Digital Design: Combinational Logic

Mark A. Strain, P.E.

Introduction

You will encounter digital circuits multiple times in any given day. Digital circuits have infiltrated society in ways unheard of only a few decades ago. They are everywhere and seem to be in everything. Without them, we would have no microprocessors. Without microprocessors, we would have no computers, smartphones, sophisticated fifth-generation fighter jets, or even something as simple and convenient as a coffee maker that brews the coffee before we wake up that shuts off automatically when we forget to turn it off. Maybe we could still design a coffee maker with an analog clock with a mechanical switch that will shut off the hot plate when the clock advances forward past a mechanical set point, but the point is that these little devices (digital circuits) are commonplace and here to stay.

Digital circuits are comprised of tiny little on/off switches called transistors. The transistor is the building block of all digital circuits. This revolutionary little switching device was invented in 1947 and its creators were awarded the Nobel Prize in Physics a few years later, and rightly so. Only a few other inventions have impacted and affected our lives in so many ways.

The transistor is the fundamental building block of digital circuits. It has been miniaturized by many orders of magnitude since its inception. This has allowed for the explosive growth in the complexity of digital circuits and microprocessors. Before the transistor, computers were built with mechanical switches called relays. There were bulky, slow and highly prone to failure. If the computing industry was limited to using mechanical relays for processor cores, then the progression in computing technology would have come to a halt decades ago. Simple "hand" calculators would still be the size of a room and we would have no mobile phones not to mention a smartphone that understands the spoken language, "Siri, send a text".

Transistors can be organized into logic gates. The most basic gates are AND, OR and NOT. With these fundamental gates, all other gates can be built. Boolean algebra describes logic gates in symbolic form which gives a designer the ability to design a complicated logic circuit using math by forming equations. These equations are directly transformed into logic symbols and into a logic circuit. Connecting several logic gates together forms something called combinational logic. With this combinational logic, adders can be fabricated as well as encoders, decoders, multiplexers and demultiplexers. A multiplexer is a device that allows one input to be selected from several inputs. An arithmetic logic unit (ALU) is a multiplexer which is at the heart of a microprocessor's core.

Binary Numbering System

In everyday life we are used to the numbering system known as the decimal numeral system. Our common numbering system is based on Arabic numerals (or symbols). Our numbering system is also base-10 which means there are ten symbols (0, 1, 2, 3, 4, 5, 6, 7, 8 and 9) used to represent every possible combination of numbers. Our numeral system is based on the number ten probably because long ago we discovered that we each have ten fingers which are useful tools to count on when doing simple math.

Computer systems and other digital systems use a numbering system based on a number other than ten. Digital systems (such as a computer central processing unit) use a numbering system based on the number two. This base-2 numbering system is called the binary system. The binary system uses two symbols (0 and 1) to represent every possible combination of numbers.

Decimal Notation

When we write decimal (base-10) numbers, we use positional notation. This means that each digit in a number is multiplied by a specific power of ten. The powers of ten (exponents) are positive on the left side of the decimal point and the powers of ten (exponents) are negative on the right side of the decimal point.

For example, consider the decimal number 3854:

$$\begin{aligned} &= 3(10^3) + 8(10^2) + 5(10^1) + 4(10^0) \\ &= 3000 + 800 + 50 + 4 \\ &= 3854 \end{aligned}$$

Now consider the decimal number 1256.79:

$$\begin{aligned} &= 1(10^3) + 2(10^2) + 5(10^1) + 6(10^0) + 7(10^{-1}) + 9(10^{-2}) \\ &= 1000 + 200 + 50 + 6 + 0.7 + 0.09 \\ &= 1256.79 \end{aligned}$$

...	10^5	10^4	10^3	10^2	10^1	10^0	.	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	...
			1	2	5	6	.	7	9				

Binary Notation

The binary system also uses positional notation. Each digit in the base-2 numbering system is multiplied by a specific power of two. Just as in the base-10 system, the powers of two are positive on the left side of the binary point and the powers of two are negative on the right side of the binary point.

Binary to Decimal

Each digit is a multiple of a power of 2. All digits to the left of the decimal point are positive powers of 2 and all digits to the right of the decimal point are negative powers of 2.

Consider the binary number 1011:

$$\begin{aligned} &= 1(2^3) + 0(2^2) + 1(2^1) + 1(2^0) \\ &= 8 + 0 + 2 + 1 \\ &= 11 \text{ (base-10)} \end{aligned}$$

Now consider the binary number 10101101.101:

$$\begin{aligned} &= 1(2^7) + 0(2^6) + 1(2^5) + 0(2^4) + 1(2^3) + 1(2^2) + 0(2^1) + 1(2^0) + 1(2^{-1}) + 0(2^{-2}) + 1(2^{-3}) \\ &= 128 + 0 + 32 + 0 + 8 + 4 + 0 + 1 + 1/2 + 0 + 1/8 \\ &= 173.625 \text{ (base-10)} \end{aligned}$$

...	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	...
		1	0	1	0	1	1	0	1	.	1	0	1			

Decimal to Binary

It is often necessary to represent a decimal fraction in binary. The integer part (to the left of the decimal point) is converted to binary by continually dividing by 2 until you get to 1. The fractional part is converted to binary by continually multiplying by 2 until you get 0 or a repeating sequence or you get tired.

Consider the decimal number 142.378:

First, convert the integer part:

Divide by 2	Remainder
142 / 2	0
71 / 2	1
35 / 2	1
17 / 2	1
8 / 2	0
4 / 2	0
2 / 2	0
1	1

Now, read the remainder part backwards and that is the binary representation of the integer part of the number: 10001110

Next, convert the fractional part. Start with the number to the right of the decimal point (0.378). Multiply the number times 2 and record what is to the left of the decimal place after this operation. Then take this number and discard whatever is to the left of the decimal place and continue.

$$0.378 * 2 = 0.756 \rightarrow 0$$

$$0.756 * 2 = 1.512 \rightarrow 1$$

$$0.512 * 2 = 1.024 \rightarrow 1$$

$$0.024 * 2 = 0.048 \rightarrow 0$$

$$0.048 * 2 = 0.096 \rightarrow 0$$

$$0.096 * 2 = 0.192 \rightarrow 0$$

$$0.192 * 2 = 0.384 \rightarrow 0$$

$$0.384 * 2 = 0.768 \rightarrow 0$$

$$0.768 * 2 = 1.536 \rightarrow 1$$

$$0.536 * 2 = 1.072 \rightarrow 1$$

$$0.072 * 2 = 0.144 \rightarrow 0$$

$$0.144 * 2 = 0.288 \rightarrow 0$$

$$0.288 * 2 = 0.576 \rightarrow 0$$

$$0.576 * 2 = 1.152 \rightarrow 1$$

$$0.152 * 2 = 0.304 \rightarrow 0$$

...

Now, read the number forwards and that is the binary representation of the fractional part of the number: 0.011000001100010

Therefore, $142.378 = 10001110.011000001100010\dots$

Negative numbers in Binary

In mathematics, negative decimal numbers can be represented by using a minus “-” sign. In digital circuits numbers are represented only by a sequence of bits, either a 0 or a 1 and no plus or minus sign. The most popular methods of representing signed numbers in binary are the ones' complement and twos' complement methods. These methods allow subtraction to be performed by adding the complement of a number instead of subtracting the number. The utilization of ones' complement and twos' complement greatly simplifies digital circuits since addition is a fundamental operation.

Ones' Complement

The ones' complement of a binary number is obtained by negating the number by inverting all of the bits. This is accomplished by changing all of the 0s into 1s and all of the 1s into 0s. The ones' complement of a number behaves as the negative of the original number. An addition operation is a fundamental operation in digital systems. There is no fundamental subtraction operation.

Subtraction of two numbers is equivalent to adding one number to the negative of the other number. Therefore, any subtraction operation is equivalent to inverting one of the numbers and adding it to the other number.

Consider taking the ones' complement of the number 0000 1100 (base-2):

0000 1100

Invert all of the bits

1111 0011

Ones' complement is seldom used in digital systems because when a ones' complement number is added to another number, the result is offset by -1 . In other words, the result of a subtraction operation (using ones' complement) is off by -1 .

Consider subtracting 8 from 12 using ones' complement:

ones' complement of 8:

0000 1000

Invert all of the bits

1111 0111

Now subtract 8 from 12:

$$12 - 8 = 4$$

Add the ones' complement of 8 to 12: $(12 - 8)$

0000 1100

1111 0111

0000 0011

= 3 (base-10)

Note that the result is off by one. This problem is resolved by performing a twos' complement operation instead of a ones' complement.

Twos' Complement

Twos' complement representation of a binary number has widespread use in digital systems. It solves the problem of the -1 offset that a ones' complement produces.

Consider the same subtraction operation as above, but this time using twos' complement of 8:

0000 1000

Invert all of the bits and add one

$$\begin{array}{r}
 1111\ 0111 \\
 0000\ 0001 \\
 \hline
 1111\ 1000
 \end{array}$$

(Note: Binary digits are often grouped in groups of four for readability.)

Now subtract 8 from 12:

$$12 - 8 = 4$$

Add the twos' complement of 8 to 12

$$\begin{array}{r}
 0000\ 1100 \\
 1111\ 1000 \\
 \hline
 0000\ 0100 \\
 = 4 \text{ (base-10)}
 \end{array}$$

Now consider subtracting 23 from 17: (17 - 23)

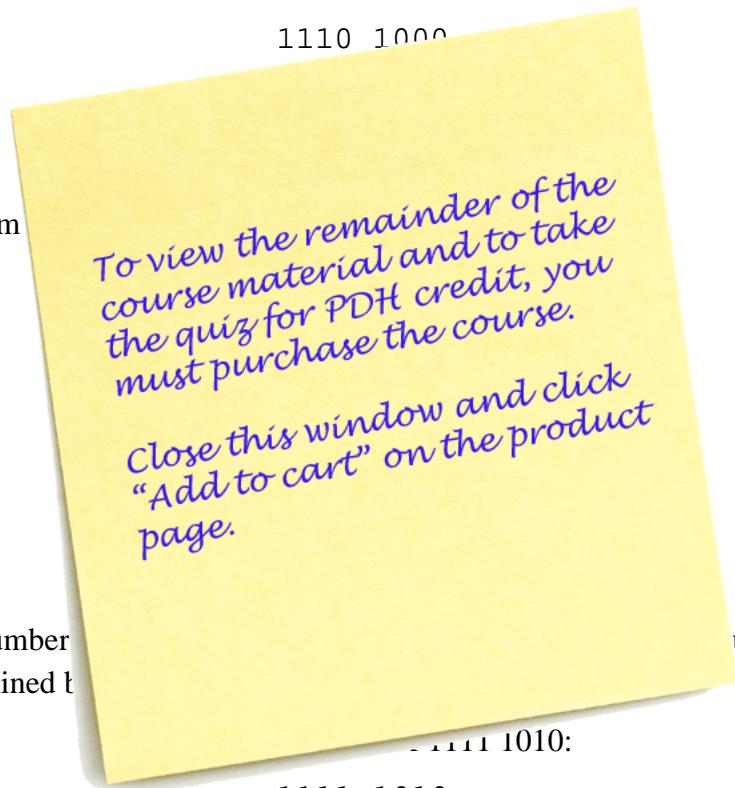
twos' complement of 23:

$$0001\ 0111$$

Invert all of the bits and add one

$$1110\ 1000$$

Now subtract 23 from



This is a negative number (sign) may be determined b

number (without the

$$\begin{array}{r}
 \dots 1111\ 1010: \\
 1111\ 1010
 \end{array}$$