Digital Circuits Made Easy

An Online Continuing Education Course for Engineers

Course Number: E-1030

Credit: 1 Hour / 1 PDH / 1 CPD

Digital Circuits Made Easy

Robert J. Scoff, P.E.

Table of Contents

1. Introduction

When learning about digital circuits, it's hard to know where to begin. So, let's define a "bit." A bit is one digit that is either a Zero(0) or a One(1). It can't be ½ or any other number or fraction. Even though digital circuits are all about handling zeros and ones, we can't say Zero plus One divided by 2 (the number of digits or bits) is ½ in the digital world. Even though it makes perfect sense, it just doesn't work that way. Digital circuits only work with the two states of a bit, zero or one. The concept of a bit with a value of ½ is simply invalid. So, let's start with a digital bit that can only have a value of zero or one.

2. Zeros and Ones

Digital circuits all start with the idea that many physical ideas have two states. They can be zero and one, off and on, pass or don't pass, and some others that are possible. The list could actually go On and On. Or is that On and Off or Off and Off? Other examples include day and night, hot and cold, here and there. Some of these ideas are very subjective. Day passing to night goes through twilight. Cold passing to hot goes through lukewarm. Digital circuits are not like that at all. They are very precise and dependable. There are no ½ s in a digital bit.

In our work with digital circuits, we will be using zero (0) for an off state and One (1) for an on state. This is to maintain consistency. There are electrical and electronic circuits where the Zero and One conditions can occur. A simple example is an ordinary light switch. In one position, a light will turn on. In the other position, the light is off. Figure 2.1 shows this.

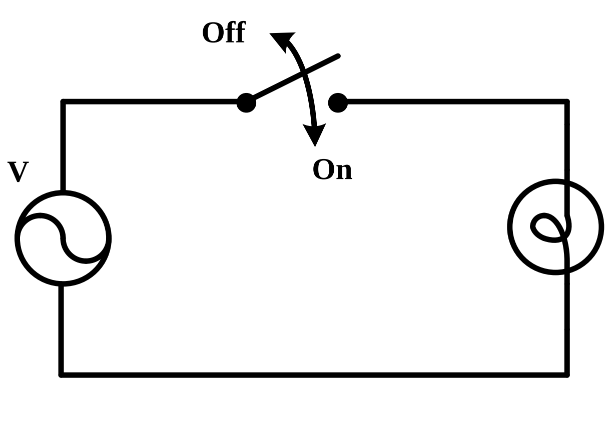

Figure 2.1 Simple Example of an Off and On Circuit

We are going to look at two ways to represent digital quantities. They are Truth Tables and Minterm Maps.

3. Truth Tables

If there is only one bit in our digital number, the truth table will look like Figure 3.1.

A	**X**	**Dec Value**
0	**?**	**0**
1	**?**	**1**

Figure 3.1 A One Bit Truth Table

We call our digital variable 'A.' The Unknown 'X' can take on a digital value of zero or one. The decimal value is as shown, and for this simple example, it can only have the value of 0 or 1. The unknown, 'X,' will have a value of 0 for an 'A' of 0 and a value of '1' for an 'A' of 1. This quickly gets more complicated if there are 2 digital variables, A and B. This is shown in Figure 3.2.

B	**A**	**X**	**Dec Value**
0	**0**	**?**	**0**
0	**1**	**?**	**1**
1	**0**	**?**	**2**
1	**1**	**?**	**3**

Figure 3.2 A Two-Bit Truth Table

Now, the X variable, which can only take on the values of Zero or One, will depend upon what A and B are doing. The decimal value is defined as A times 2^0 plus B times 2^1. This is how a digital number is converted to a decimal number. Incidentally, 2^0 is equal to 1, and 2^1 is equal to 2. We have actually counted from decimal 0 to decimal 3, and that there are 4 possible states. X is still an unknown that can only have two possible states, 0 and 1. This is about digital circuits, but decimal numbers can be represented by digital numbers. Notice that ½ doesn't enter into the calculations. We also have to be careful to keep the digital variables in a proper perspective. There may be times when it is easier or even desirable to put A in the left-hand column and B in the column second from the left. For now, we are going to denote the digital variables from A on the right and B, C, D, etc., to the left. Let's look at a three digital variable example. Figure 3.3 shows this. We are still keeping the digital variables labeled from right to left. We are still keeping the digital variables labeled from right to left. So, the decimal value is A times 2^0 plus B times 2^1 plus C times 2^2. A, B, and C can only take on the values of 0 and 1. This makes the math pretty simple.

C	**B**	**A**	**X**	**Dec Value**
0	**0**	**0**	**?**	**0**
0	**0**	**1**	**?**	**1**
0	**1**	**0**	**?**	**2**
0	**1**	**1**	**?**	**3**
1	**0**	**0**	**?**	**4**
1	**0**	**1**	**?**	**5**
1	**1**	**0**	**?**	**6**
1	**1**	**1**	**?**	**7**

Figure 3.3 A Three Digital Variable Truth Table

We could extend to 4 or more digital values, but the tables become exceedingly long. If you like, try doing a truth table for 4 digital variables. Let's next look at a way to see digital variables in Min Term Maps.

4. Min Term Maps

Min Term Maps are another way to look at digital variables. They can be made directly from the Truth Tables. Figure 4.1 shows a Min Term Map from a one digital bit truth table.

Truth Table

A	**X**	**Dec Value**
0	**?**	**0**
1	**?**	**1**

Min Term Map

1	**A**
0	**$\overline{A}$**

Figure 4.1 Truth Table and Min Term Map for a One Variable Digital Circuit

Notice that here we have added a new concept. Not A is commonly denoted by **$\overline{A}$**. Not B is commonly denoted by $\overline{B}$ and so forth for any digital variable. This means that if **A** is 1, then **$\overline{A}$** is 0 and vice versa. The numbers 0 and 1 that show on the one binary digit Min Term Map are actually decimal values. So, A being 0 gives a decimal value of 0, and A being 1 gives a decimal value of 1. This will become easier to see on the next drawing, Figure 4.2, which shows the Truth Table and Min Term Map for a 2 digital variable number that counts from 0 to 3 in decimal. We can also say that the digital variable can be true or false. If the digital variable has a value of 1, it is true. If it has a value of 0, it is false.

B	**A**	**X**	**Dec Value**
0	**0**	**?**	**0**
0	**1**	**?**	**1**
1	**0**	**?**	**2**
1	**1**	**?**	**3**

A	**$\overline{A}$**	
3	**2**	**B**
1	**0**	**$\overline{B}$**

Figure 4.2 Two-Variable Digital Truth Table and Min Term Map

Looking at the Min Term Map, it can be seen that if **A** and **B** are both digital 1, then the decimal value is 3. Let’s go one more step and make a Min Term Map for a 3 digital variable. This is shown in Figure 4.3.

C	B	A	X	Dec Value
0	0	0	?	0
0	0	1	?	1
0	1	0	?	[illegible]
0	1	[illegible]	[illegible]	[illegible]
1	0	[illegible]	[illegible]	[illegible]
1	0	[illegible]	[illegible]	[illegible]
1	1	[illegible]	[illegible]	[illegible]
1	1	[illegible]	[illegible]	[illegible]

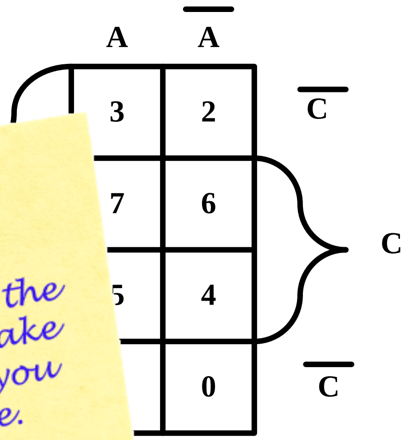

Figure 4.3 Trut[illegible]**igital Variable**

Let’s go one step further [illegible] digit digital variable. This is shown i[illegible]