

## Fundamentals of Digital Quadrature Modulation

*How a formerly analog-only application, the quadrature modulator, is finding a home in modern digital CMOS fabrication techniques*

**By Ken Gentile**

The concept of modulation is not at all new, it has been around since the early days of radio. Modulation, especially in the context of RF applications, refers to the mixing of two sinusoidal signals. One signal is the message signal and contains the information to be modulated. It usually consists of a band-limited spectrum of sinusoids (such as music). The other signal is the carrier signal and is generally a pure tone (a sinusoid of a single frequency). The frequency of the carrier is referred to as the carrier frequency and will be

denoted by the symbol  $f_c$  (or  $\omega_c$  for radian frequency) throughout this text. Typically,  $f_c$  is much higher in frequency than the highest frequency component of the message signal.

### The Basics

The concept of modulation comes from the trigonometric identity:

$$\cos(x)\cos(y) = \frac{1}{2}\cos(x+y) + \frac{1}{2}\cos(x-y)$$

If we assume that the message signal is a pure tone of frequency,  $f_m$ , then the message can be mathematically represented as  $\cos(2\pi f_m t)$ . The same assumption can be made about the carrier signal, thereby expressing it as  $\cos(2\pi f_c t)$ . The “pure tone” assumption makes the mathematics much more tractable. However, it is important to keep in mind that the message signal is rarely a pure tone.

Typically, it is composed of time variations in amplitude, frequency, phase, or any combination thereof. Even the carrier need not necessarily be a pure sinusoid. Applications exist in which the carrier signal is a square wave with a fundamental frequency,  $f_c$ . The harmonics of  $f_c$  inherent in the square wave are dealt with by filtering the modulated signal.

The mixing process mentioned earlier can be thought of as a multiplication operation. Therefore, the trigonometric identity above may be employed to represent the mixing process as follows:

$$\begin{aligned} \cos(2\pi f_c t)\cos(2\pi f_m t) &= \frac{1}{2}\cos[2\pi(f_c + f_m)t] \\ &+ \frac{1}{2}\cos[2\pi(f_c - f_m)t] \end{aligned}$$

Thus, the mixing of the message and carrier results in a transformation of the frequency of the message. The message frequency is translated from its original frequency to two new frequencies — one greater than the carrier ( $f_c + f_m$ ), and one less than the carrier ( $f_c - f_m$ ), the upper and lower side bands, respectively. Furthermore, the translated signal undergoes a 6 dB loss (50 percent reduction) as dictated by the factor of  $\frac{1}{2}$  appearing on the right-hand side of the equality.

The form of modulation just described is referred to as “double sideband modulation,” because the message is translated to a frequency range above and below the carrier frequency. Another form of modulation, known as single sideband modulation, can be used to eliminate either the upper or lower sideband. One method of performing single sideband modulation is to employ a

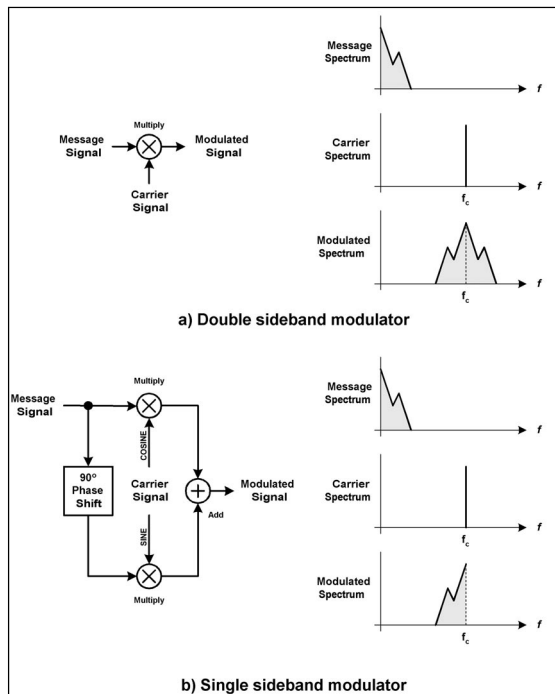


Figure 1. Functional representation of a single and double sideband modulator and associated frequency spectra

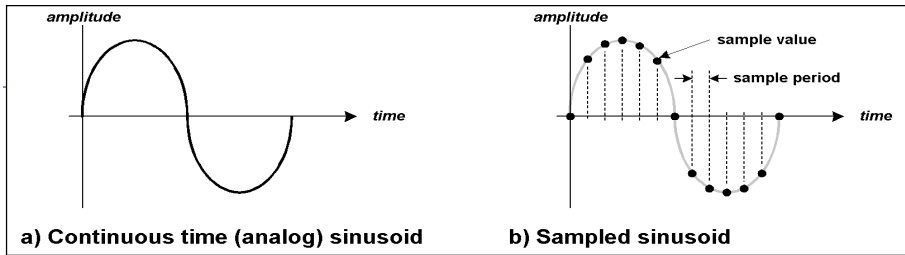


Figure 2. A continuous time representation of a sinusoid

quadrature modulator. A quadrature modulator mixes the message with two carriers. Both carriers operate at the same frequency, but are shifted in phase by 90 degrees relative to one another (hence the “quadrature” term). This simply means that the two carriers can be expressed as  $\cos(2\pi f_c t)$  and  $\sin(2\pi f_c t)$ . The message, too, is modified to consist of two separate signals: the original and a 90 degree phase-shifted version of the original. The original is mixed with the cosine component of the carrier and the phase-shifted version is mixed with the sine component of the carrier. These two modifications result in the implementation of the single sideband function. Trigonometrically, this can be expressed as:

$$\begin{aligned} & [\cos(x)\cos(y)] + [\sin(x)\sin(y)] \\ &= \left[ \frac{1}{2} \cos(x+y) + \frac{1}{2} \cos(x-y) \right] \\ &+ \left[ \frac{1}{2} \cos(x-y) + \frac{1}{2} \cos(x+y) \right] \end{aligned}$$

Note that the right-hand side of the equation reduces to  $\cos(x-y)$ , the lower sideband, only. In the above equation,  $x$  is the carrier and  $y$  is the message. Incidentally, changing the sign of the operator on the left-hand side of the equation results in only the upper sideband appearing on the right hand side.

In figure 1 the functional representation of a single and double sideband modulator are shown along with the associated frequency spectra. However, the message is shown as a band-limited spectrum rather than a pure tone, which better represents a real-world application. Each constituent frequency in the message is translated to one or both sides of the carrier, as shown.

### Digital Number Representation

In addition to an understanding of modulation basics, it is also helpful to have at least an elementary understanding of numeric representation as employed in the digital world. The basic building block of digital numerics is the bit, which can only take on two

values — 0 or 1. Bits may be concatenated to form larger numbers just like decimal digits. For example, a single digit in the decimal system can take on values from 0 to 9, but the number 148 uses three digits to represent the sum:  $8(10^0) + 4(10^1) + 1(10^2) = 8 + 40 + 100 = 148$ . Each digit carries an increasing power of 10 weighting starting with the rightmost digit (ones, tens, hundreds...). Similarly, binary numbers are made up of bits carrying a power of two weighting starting with the rightmost bit. For example,  $10010100 = 0(2^0) + 0(2^1) + 1(2^2) + 0(2^3) + 1(2^4) + 0(2^5) + 0(2^6) + 1(2^7) = 0 + 0 + 4 + 0 + 16 + 0 + 0 + 128 = 148$ .

The benefit of binary number representation is not readily apparent. For instance, binary numbers can become very cumbersome to work with. Consider the number 1 million. It can be expressed with 7 decimal digits, whereas 20 binary digits are required; not very efficient in terms of notation. However, the real beauty of binary numbers comes from the fact that each digit can take on only two values. This is readily modeled by the state of a switch (on or off), which can be electronically implemented with a single transistor. Transistors, in turn, can be physically realized by the millions on a single silicon chip. The ability to place millions of binary switches on a single chip is what gives digital its advantage.

Returning to the concept of modulation, it was pointed out that modulation is applied in the context of sinusoids. Since a sinusoid is represented by a trigonometric function, it can take on positive or negative values. So, digital modulation will require the ability to represent negative binary values. From a notation point of view this is utterly simple — place a negative sign to the left of the leftmost bit. However, from the point of view of a physical implementation a negative sign does not exist.

To tackle the negative number problem, the concept of two's-complement binary representation was developed. In this system, the leftmost bit carries the positive/negative information. The leftmost bit is often referred to as the most significant bit or MSB. The remaining

bits carry the magnitude information. Two's-complement numbers, in which the MSB is 0, are positive numbers, while those for which the MSB is 1 are negative numbers. When the MSB is 0 (positive numbers) the non-MSB bits are taken as an ordinary binary number. For example, the two's-complement number 0101 is +5 in decimal notation. When the MSB is 1 (negative numbers) the non-MSB bits are first inverted (i.e., 0s become 1s and vice versa) and then 1 is added to the result. For example, the two's-complement number 1101 is -3 in decimal notation. The MSB indicates a negative value, while the remaining bits (101) are inverted to yield 010, or 2 in decimal. Then 1 is added to the result to make 3. So, the end result is -3 as indicated by the MSB. Although this may seem complicated it is readily implemented in hardware using fundamental digital building blocks.

It should be noted that the digital implementation of an analog function requires a certain amount of compromise. For instance, an analog signal is not a numeric quantity, but a physical quantity. A digital signal, on the other hand, is a numeric quantity and serves only to model an analog signal.

Digital systems rely on a compromise between absolute numeric accuracy and sufficient numeric accuracy. For example, the number of amplitude values that constitute an analog sinusoidal signal is infinite. That is, we can think of an analog sinusoid as being made up of infinitely small steps in amplitude. If we elect to make the steps larger, we sacrifice accuracy for the luxury of requiring fewer numbers with which to represent the purely analog waveform.

In effect, we can trade an absolutely pure analog sinusoid for one made up of small (but finite) steps plus some noise (the deviation between each step and the ideal analog equivalent). The step size is directly related to digital resolution. Resolution is the number of bits used to represent the full range of amplitude of an analog signal. For example, a 10-bit binary number can represent an analog signal to an accuracy of 1 part in  $2^{10}$ , or 1/1024 (approximately 0.1 percent).

### Sampled Digital Signals

Digital modulation is wholly dependent on the fundamental concepts of sampling theory. The subject of sampling is far too broad to be covered here,

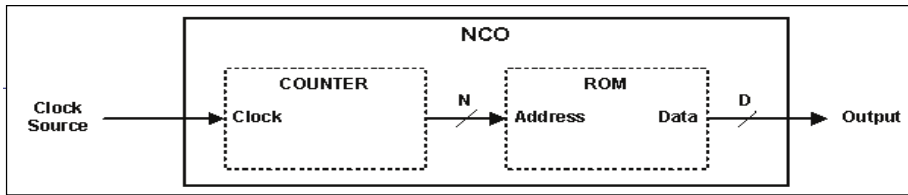


Figure 3. A numerically controlled oscillator

but a brief overview is given for the sake of clarity.

Since the topic is modulation, we will use a sinusoidal signal as a model. A continuous time representation of a sinusoid is shown graphically in figure 2a. At any instant, “*t*,” on the horizontal axis, the amplitude of the sinusoid may be found on the vertical axis. A uniformly sampled version of the sinusoid is shown in figure 2b. Note that the amplitude is only known at certain discrete points in time (the sampling instants), which are uniformly distributed in time. Sampling theory states that as long as at least two sampling instants occur within a complete cycle of the sampled sinusoid, then the sinusoid can be completely reconstructed from the two sample points.

What makes sampling attractive is that the amplitude of the sample values can be encoded as twos-complement binary numbers. So, from a digital perspective, if we can generate a proper sequence of numeric values, we can generate a digital sinusoid. But why would we want to generate a digital sinusoid in the first place? Recall that modulation requires a carrier signal; i.e., a sinusoid. In the analog world this is implemented with an oscillator circuit that operates at the specified frequency. In the digital world, some form of a digital oscillator must be realized. It turns out that this can be readily accomplished with a numerically controlled oscillator.

### Numerically Controlled Oscillator

In its simplest form, a NCO consists of a lookup table made up of sinusoidal sample values, (usually implemented as a read only memory – ROM), a binary counter for addressing the ROM, and a clock signal to drive the counter (see figure 3). Successive address locations in the ROM contain the successive sample values of the desired sinusoid. As the counter is clocked, each new count addresses the next ROM location causing the appropriate digital number to appear at the ROM output. The rate at which the counter is clocked is the sample rate of the system. If we examine the output of the ROM over time, we would observe a series of numbers that update at the sample rate. The numbers would span a numeric range that

depends on the bit-width of the ROM output. Thus, the number of bits in the ROM’s output word determines the resolution of the desired digital sinusoid. As an example, if the ROM output were 10 bits, then using twos-complement representation would yield a numeric range of –512 to +511 for the amplitude values of our sampled sinusoid.

The particular NCO example just described would only be capable of generating a digital sinusoid of one specific frequency; namely, the sample rate divided by the number of samples stored in the ROM (assuming that the samples stored in ROM span a single cycle of the sinusoid). A more flexible NCO would use a fairly large ROM (perhaps containing 4,096 samples, or more) and a counter that can count by some input modulus; that is, count by 1, 2, 3, 4, 5, etc. as determined by a “frequency control number”. For example, if the sample rate is 10 MHz, the ROM is 4,096 words in length, and the frequency control number is 1, then the output sinusoid would have a frequency of: 10 MHz/4096 or 2.44 kHz. If the frequency control number is 5, then the counter jumps 5 steps for each input clock period. This, in turn, causes every 5<sup>th</sup> ROM address to be accessed. The net result is a sinusoid with coarser amplitude steps, but of greater frequency. Specifically, the frequency of the new sinusoid would be: 10 MHz/(4096/5) or 12.21 kHz. In general, this can be expressed as:  $f_s(N/M)$  where  $f_s$  is the system sample rate,  $N$  is the frequency control number, and  $M$  is the length of the ROM.

Many variations on this theme can be found in the literature. The main point is that a NCO serves as a means for generating a digital sinusoid at a specific sample rate but with programmable frequency. The frequency is restricted, however, to an integer multiple of the sample rate divided by the ROM word length up to a maximum of  $\frac{1}{2}$  of the sample rate (the Nyquist constraint). The larger the ROM length (address range), the finer the frequency resolution. The more bits in the ROM output word, the finer the amplitude resolution.

### The Digital Quadrature Modulator

The foregoing sections provided a foundation for understanding modula-

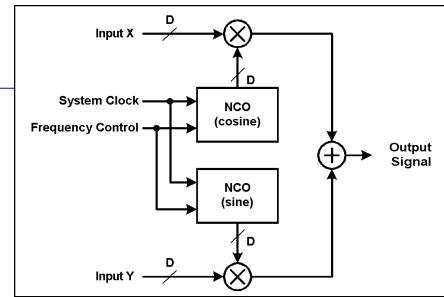


Figure 4. The digital quadrature modulator

tion, number representation in the digital domain, and a method for generating a sampled sinusoid. These three concepts are required for a firm understanding of digital modulation. The fundamental building blocks of a digital quadrature modulator are essentially the same as those for the analog single-sideband modulator shown in figure 1b.

The digital version is shown in figure 4 — the main difference being that the two multipliers, the adder, and the carrier signals are all made of digital building blocks.

The model shown in figure 4 can be readily implemented using digital hardware. The digital makeup of the NCOs was discussed earlier. Multipliers and adders, too, are readily designed from elemental digital building blocks (AND, OR and NOT units). The only real constraints are the maximum possible sampling frequency (which is mostly dependent on the semiconductor process) and power dissipation.

There is one fundamental rule that cannot be overlooked in digital modulation. Both the digital carrier signal and the digital message signal must be sampled at the same rate. In some instances, the message signal consists of a digital signal sampled at a rate less than the carrier. Such situations require that the message signal be digitally up-sampled to match the carrier sample rate. However, this is another topic altogether and is well beyond the scope of this article. Sample rate conversion techniques are rigorously covered in the existing literature.

Returning to figure 4, the digital quadrature modulator has two message signal inputs: *X* and *Y*. In addition, two NCOs produce the quadrature carrier signals. The same system clock and frequency control number are provided to both NCOs. However, one NCO has a cosine wave stored in its ROM while the other NCO has a sine wave stored in its ROM. The carrier frequency ( $f_c$ ) is determined by the frequency control number.

Typically, the *X* and *Y* input signals are intended to be quadrature compo-

## Digital Quadrature Modulator Model

Ken Gentile, Analog Devices, Inc., Greensboro, NC 27409

### ----- System Parameters -----

Units:    kHz =  $10^3$     MHz =  $\text{kHz}^2$

NCO ROM length:    **N := 10**    ...number of address bits

NCO ROM width:    **D := 6**    ...number of data bits

System sample rate:    **F<sub>s</sub> := 100 MHz**

Desired carrier frequency:    **F<sub>carrier</sub> := 5.5 MHz**

Frequency resolution:     $dF := F_s \cdot 2^{-N}$     **dF = 0.097656 MHz**    ...lowest frequency for a N-bit NCO sampling at F<sub>s</sub>

Frequency Control Number:     $FCN := \text{round}\left(\frac{F_{\text{carrier}}}{dF}\right)$     **FCN = 56**

Actual carrier frequency:     $F_c := dF \cdot FCN$     **F<sub>c</sub> = 5.469 MHz**    ...the nearest multiple of dF to F<sub>carrier</sub>

### ----- Create a sine & cosine NCO -----

Cosine NCO Function

Sine NCO Function

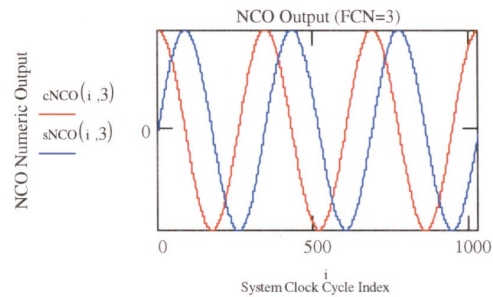
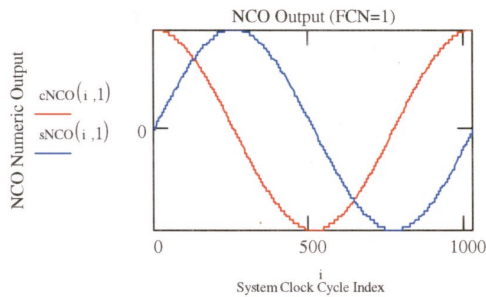
$$cNCO(n, FCN) := \text{round}\left(\frac{2^D - 1}{2} \cos\left(2\pi \frac{\text{mod}(n \cdot FCN, 2^N)}{2^N}\right) - 0.5\right)$$

$$sNCO(n, FCN) := \text{round}\left(\frac{2^D - 1}{2} \sin\left(2\pi \frac{\text{mod}(n \cdot FCN, 2^N)}{2^N}\right) - 0.5\right)$$

The cNCO & sNCO function each have two arguments -- n: number of elapsed clock cycles | FCN: frequency control number

### ----- Demonstrate NCO Operation -----

$i := 0..2^N - 1$     ...system clock cycle index (allow for at least  $2^N$  clock cycles)



### ----- Create X & Y Input Signals -----

Let X and Y be pure sinuoids both operating at frequency, F<sub>in</sub>, and offset by  $\pi/2$  radians. The amplitude of each is of P-bit resolution.

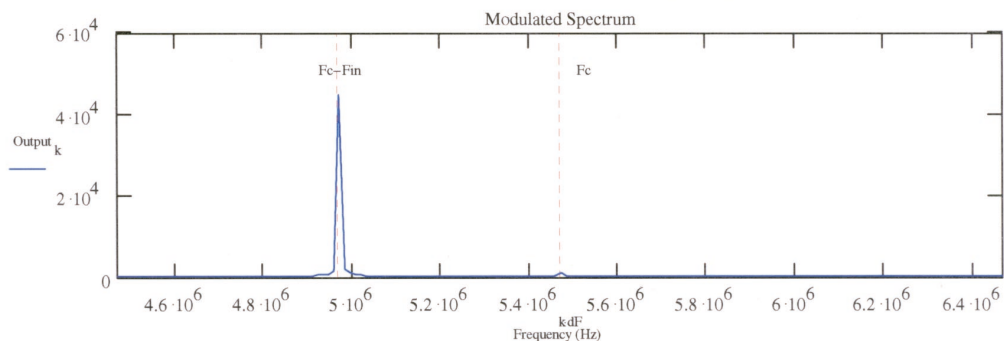
**F<sub>in</sub> := 0.5 MHz**    **P := 6**     $Xin(n) := \text{round}\left(\frac{2^P - 1}{2} \cos\left(2\pi \frac{F_{in}}{F_s} \cdot n\right) - 0.5\right)$      $Yin(n) := \text{round}\left(\frac{2^P - 1}{2} \sin\left(2\pi \frac{F_{in}}{F_s} \cdot n\right) - 0.5\right)$

### ----- Quadrature Modulate X & Y with F<sub>c</sub> and Display Output Spectrum -----

**B := 13**     $i := 0..2^B - 1$     ...system clock cycle index     $dF := F_s \cdot 2^{-B}$     ...fourier frequency

$QuadMod_i := Xin(i) \cdot cNCO(i, FCN) + Yin(i) \cdot sNCO(i, FCN)$     ...time domain output

Output :=  $\left| \text{fft}(QuadMod) \right|$     ...output spectrum     $k := 0.. \text{length}(\text{Output})$



nents. For example, if  $X$  were a digital cosine wave of frequency  $f_m$ , and  $Y$  were a digital sine wave of the same frequency, then the output of the quadrature modulator would be a single-sideband tone of frequency  $f_c - f_m$ . The Appendix contains a Mathcad program that precisely models this scenario. For other scenarios, simple modifications can be made to the program. For example, an upper sideband can be generated by simply changing the sign of the operator from + to - in the right-hand portion of the "QuadMod<sub>i</sub>" statement.

If a double-sideband signal is desired simply replace  $Y_{in}(t)$  in the right-hand portion of the "QuadMod<sub>i</sub>" statement with  $X_{in}(t)$ . The NCO ROM parameters,  $N$  and  $D$ , can also be changed to see the effects of both frequency and amplitude resolution. The system sample rate ( $F_s$ ) and carrier frequency ( $F_{carrier}$ ) can both be changed, as well. Note, however, that the actual carrier frequency ( $F_c$ ) may not exactly match the value entered for  $F_{carrier}$ . This is due to the fact that the frequency control number (FCN) for the NCO must be an integer value because it controls the modulus for a binary counter. This restriction means that only finite frequency resolution is possible. Specifically, only those frequencies that correspond to the FCN values are possible.

## The Results

This article has demonstrated the key elements of digital quadrature modulation. The high speeds available from today's semiconductor processes make it ever more practical to implement modulation functions in digital rather than analog technologies.

This trend will likely continue as digital semiconductor technology pushes operating speeds ever higher. It should be kept in mind that the end result of implementing analog functions in the digital domain is a time sequence of digital numbers, a natural consequence of the sampling process. However, this number stream must ultimately be converted to an analog waveform to be of any practical use. Thus, a digital-to-analog converter (DAC) must be employed to transform the digital signal into the analog domain. As such, there is still a significant role for analog circuitry, especially DACs. As both sample rate and resolution increase, the requirement for high speed, high resolution DACs becomes ever more apparent.

**RF**

## About the author

Ken Gentile graduated with honors from North Carolina State University where he received a B.S.E.E. degree in 1996. Currently, he is a system design engineer at Analog Devices Inc. ([www.analog.com](http://www.analog.com)) where he is responsible for the system level design and analysis of signal synthesis products. His specialties are the application of digital signal processing techniques in communications systems and analog filter design. Ken has been published in several industry trade journals and was a primary contributing author to ADI's "A Technical Tutorial on Digital Signal Synthesis". Gentile can be reached at [ken.gentile@analog.com](mailto:ken.gentile@analog.com).