ANALOG DEVICES

AN-953
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

Direct Digital Synthesis (DDS) with a Programmable Modulus

by Ken Gentile

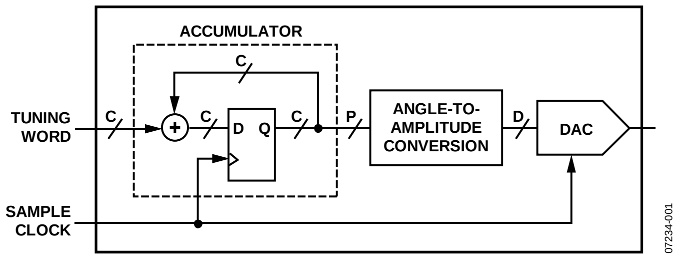

Figure 1. Typical Accumulator-Based DDS Architecture

OVERVIEW

The programmable modulus is a modification of the typical accumulator-based DDS architecture. It extends the use of DDS to applications that require exact rational frequency synthesis (any-rate applications, for example). The AD9913 is the first accumulator-based DDS product by Analog Devices, Inc., to offer the programmable modulus architecture.

TYPICAL ACCUMULATOR-BASED DDS

A typical accumulator-based DDS relies on an accumulator to recursively sum the digital input tuning word at the rate of the sample clock (see Figure 1). This produces a time series of digital words at the output of the accumulator that increases linearly until the accumulator rolls over at its maximum value of 2^C. Hence, the accumulator output has a fixed modulus of 2^C.

Usually the accumulator output is truncated to P bits (using only the MSBs) to reduce the size and complexity of the angle-to-amplitude conversion block that immediately follows the accumulator. This causes the time series of digital words produced by the accumulator to appear at the input to the angle-to-amplitude converter as P-bit word(s) ranging in value from zero to $2^P - 1$.

The angle-to-amplitude converter maps the P-bit word(s) to one revolution on the unit circle; that is, it linearly maps binary values from 0 to 2^P to radian angles from 0 to 2π. This mapping arrangement allows the angle-to-amplitude converter to translate the P-bit word(s) to D-bit amplitude value(s) (A) in a very efficient manner.

The translation process relies on the trigonometric relationship

$$x = \sin(2\pi k/2^P)$$

where:
P is the number of bits taken from the accumulator.
k is the binary value of those bits at any given instant.

The value of A for each k is the value of x scaled, offset, and rounded such that A takes on integer values between zero and $2^D - 1$.

It is possible to readily extend the angle-to-amplitude conversion function to perform both sine and cosine conversions concurrently. Therefore, DDSs that offer both conversion schemes are commonly found.

Following the angle-to-amplitude converter is a D-bit DAC. It converts the D-bit digital amplitude values produced by the angle-to-amplitude converter to analog levels. The result is a sinusoidal waveform at the output of the DAC with a frequency determined by the average rollover rate of the accumulator.

The following equation expresses the frequency of the sinusoid that appears at the DAC output for a typical accumulator-based DDS:

$$f_O = \frac{M}{2^C} f_S \qquad (1)$$

where:
f_O is the synthesized frequency.
f_S is the sampling frequency.
$M/2^C$ is the fractional scale factor (*M* and *C* are positive integers).

Normally, M is constrained to be less than 2^{C-1}. Otherwise, a Nyquist image frequency is synthesized. The quantity, 2^C, is the modulus of the accumulator, where C is the accumulator width in bits. The integer, M, is often referred to as the frequency tuning word. Since M, by definition, is an integer, then f_O is constrained to the following set of frequencies:

$$f_O \in \left\{0, \frac{f_S}{2^C}, \frac{2f_S}{2^C}, \frac{3f_S}{2^C}, \ldots, \frac{\left(2^{C-1}-1\right)f_S}{2^C}\right\} \qquad (2)$$

Thus, the set of frequencies from dc to almost ½f_S, is at equally spaced intervals of $f_S/2^C$. This spacing, $f_S/2^C$, is referred to as the frequency resolution of the DDS. Inspection of the f_O frequency set indicates that the modulus of the DDS accumulator (2^C)

determines both the frequency resolution of the DDS and the number of possible output frequencies. For example, the output frequency set of a DDS with a 32-bit accumulator has 2,147,483,648 (that is, 2^{31}) possible output frequencies for a given sample rate (f_S).

Even with such a large set of available frequencies, the typical accumulator-based DDS is not capable of generating some useful frequencies (like precisely $f_S/10$) because the accumulator modulus is fixed and the tuning word (M) must be an integer. Rearranging Equation 1 reveals that the ratio of the output frequency to the sampling frequency must satisfy

$$\frac{f_O}{f_S} = \frac{M}{2^C} \tag{3}$$

Now for output frequencies that are an integer submultiple of the sample rate (for example, $f_S/10$), f_O can be expressed as $f_O = f_S/Q$ (Q is an integer). Substituting f_S/Q for f_O in Equation 3 leads to

$$\frac{1}{Q} = \frac{M}{2^C} \tag{4}$$

Solving for M yields $M = 2^C/Q$. Because M and Q must both be integers, the only values of Q that satisfy Equation 4 are those that take the form 2^K, where K is an integer. That is, $M = 2^C/2^K = 2^{C-K}$ (both C and K are integers). To demonstrate, let $f_O = f_S/8$ and assume that C is 32. Then $Q = 8 = 2^3$ and $M = 2^{32-8} = 2^{24} = 536{,}870{,}912$ (an integer). On the other hand, if $f_O = f_S/10$, then Q is 10 and $M = 2^{32}/10 = 429{,}496{,}729.6$ (not an integer), which means that the typical DDS is not capable of generating the exact frequency $f_S/10$. The typical DDS can synthesize an output frequency extremely close to $f_S/10$ (based on the frequency resolution) but not exactly $f_S/10$. In most cases, extremely close is acceptable, but in certain applications (network clocks, for example), the exact frequency ratio is a necessity.

Inspection of Equation 4 indicates that the ratio on the right-hand side is established by the modulus of the accumulator (2^C), which is a fixed quantity governed by the accumulator size. Clearly, the fixed modulus of the accumulator limits the set of available output frequencies. However, it seems reasonable to assume that an adjustable accumulator modulus would make it possible to accommodate values of Q other than just those that take the form 2^K. Such is the foundation of the programmable modulus DDS architecture.

PROGRAMMABLE MODULUS DDS

Programmable modulus is a modification of the typical accumulator-based DDS architecture (see Figure 2). Although the underlying function of the programmable modulus technique is to alter the accumulator modulus, the actual implementation is more complicated. The complication is twofold. The first complication is that the angle-to-amplitude converter maps the entire P-bit input range (0 to 2^P) to 0 to 2π radians. It is this power-of-two mapping arrangement that allows the angle-to-amplitude converter to operate efficiently. Arbitrarily changing the accumulator modulus to something other than a power-of-two necessarily violates the mapping arrangement required by the angle-to-amplitude converter. To overcome this difficulty, the programmable modulus architecture uses a secondary accumulator to make the primary accumulator appear to have an altered modulus while still maintaining the power-of-two mapping required for the angle-to-amplitude converter.

The second complication involves spurious performance. If not done properly, modification of the accumulator modulus can introduce spectral artifacts. The programmable modulus implementation requires great care to minimize any negative impact on spurious performance.

The programmable modulus DDS architecture results in a variation of Equation 3 as follows:

$$\frac{f_O}{f_S} = \frac{M}{N} \tag{5}$$

where *M* and *N* are integers and $1 \le M < N/2$.

Notice that the frequency ratio for the programmable modulus DDS (Equation 5) is very similar to that of the typical accumulator-based DDS (Equation 3). The only difference is that N is not required to be a power-of-two for the programmable modulus. It is simply an integer that can be chosen at will. The only constraint placed on N is that when the fraction M/N is reduced to its lowest terms, N must satisfy $0 < N < 2^{32}$. This constraint indirectly ensures the required power-of-two scaling for the angle-to-amplitude converter.

The programmable modulus function of the AD9913 is implemented by programming three 32-bit registers with values for N, Y, and X. These appear as inputs to the modulus control logic in Figure 2. The numeric programming range for N, X, and Y is from 0 to $2^{32} - 1$, inclusive, but there is a lower bound imposed on N by M, namely N > 2M.

The relationship between f_S, f_O, M, N, Y, and X is summarized in Equation 6. The values of f_O and f_S establish M and N with the fraction M/N reduced to its lowest terms. M and N, along with the modulus of the accumulator (2^C), establish X and Y by means of long division.

$$\frac{f_O}{f_S} = \frac{M}{N} \longrightarrow N\overset{\quad X}{\overline{\big)\,2^C M}} \qquad \begin{array}{r} -\underline{XN} \\ Y \longrightarrow Y = 2^C M - XN \end{array} \tag{6}$$

where:
X is the integer quotient of the division process.
Y is the integer remainder.

Interestingly, M establishes the periodicity of the accumulator output sequence. That is, the accumulator is required to cycle through M overflows to complete one unique output sequence and return to its original starting point. This unique output sequence repeats indefinitely. It is worth noting that if Y is 0, then the programmable modulus is superfluous. That is, f_O can be synthesized directly from f_S using a standard DDS.

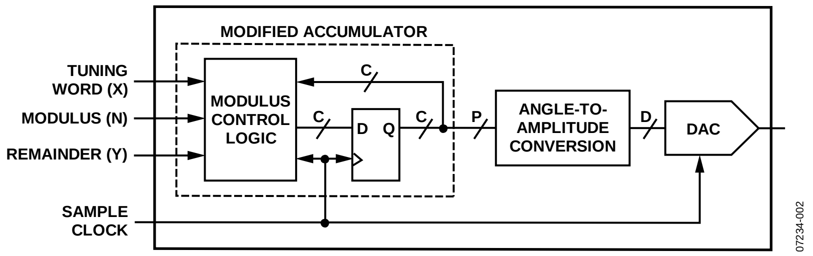

Figure 2. Programmable Modulus DDS Architecture

PROGRAMMABLE MODULUS EXAMPLE

Consider the case in which f_S = 250 MHz and the desired value of f_O is 25 MHz. This scenario synthesizes an output frequency that is an integer submultiple of the sample rate, namely $f_O = f_S/10$. The frequency ratio, f_O/f_S, leads directly to M and N, which are determined by reducing the fraction (25,000,000/250,000,000) to its lowest terms. That is:

M/N = 25,000,000/250,000,000 = 1/10

Therefore, M = 1 and N = 10. M and N, in turn, allow X and Y to be calculated as X = 429,496,729 and Y = 6.

Programming these values into the AD9913 causes it to produce an output frequency of exactly 25 MHz given a 250 MHz sampling clock.

A standard DDS is not capable of synthesizing exactly 25 MHz given that f_S = 250 MHz. The closest it can get is 25.00000002328306436538696289062 5 MHz. In a network application, this close value is not acceptable because the requirement is 25 MHz with absolutely no fractional error.

CONSIDERATIONS

It is important to point out that most computational programs (for example, Excel®, Mathcad®, and MATLAB®) can introduce numerical errors when calculating X and Y. These errors are due to internal rounding or truncation, which shows up when large numbers are multiplied or divided. Interestingly, the calculator included as part of the Windows® XP operating system provides sufficient numeric precision to perform these computations without the aforementioned errors (at least in the context of the 32-bit numbers associated with programming the AD9913). In fact, having prior knowledge of the numeric precision issue led developers of the AD9913 evaluation board software to mitigate any round-off or truncation errors.

To demonstrate the effects of round-off and truncation, once again consider the case in which the desired value of f_O is 25 MHz, but f_S is 249,999,999.5 MHz (instead of 250 MHz). The frequency ratio, f_O/f_S, leads directly to M and N by reducing the fraction (25,000,000/249,999,999.5) to its lowest terms. That is

M/N = 25,000,000/249,999,999.5 = 50,000,000/499,999,999

Therefore, M = 50,000,000 and N = 499,999,999; M and N lead to X = 429,496,730 and Y = 229,496,730.

Using Excel yields the same value for X, but Y calculates as 229,496,736. Notice that the least significant digit is 6 instead of 0 (the correct value). Although this error is extremely small (only 0.026 ppm), it still results in an output frequency that is not exactly 25 MHz but rather 25.000000000000000698492 MHz. Applications that require a precise f_O/f_S ratio would find this to be an unacceptable error.

The point is that the computing engine used for calculating N, X, and Y must have sufficient numerical precision to avoid truncation or rounding errors.

NOTES

©2008 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
AN07234-0-1/08(0)