

Advanced Phase-Lock Applications: Frequency Synthesis

James A. Crawford

©2010-2011 James A Crawford

AMI LLC

Chapter 5

Direct Digital Frequency Synthesis

Direct digital frequency synthesis first came to the forefront as a viable frequency synthesis method in circa 1971 [1] and was further popularized by a later IEEE publication [2] in 1975. The most recent textbooks on this subject were published circa 2000 (e.g., [3]) when advanced techniques like noise shaping were still in their infancy, however. The contents of this chapter bring this important synthesis technique more up to date.

The mathematics behind direct digital frequency synthesis are deeply embedded in virtually every modern communication system that employs digital signal processing. The distinction between direct digital synthesis and numerically-controlled oscillators (NCOs) has become markedly vague. In the dedicated frequency synthesis arena, however, fractional-N frequency synthesis based on Δ - Σ techniques has taken substantial market share away from direct digital synthesis except in all but the most demanding applications.

Direct digital synthesis in the context of this chapter is limited to systems that physically utilize a D-to-A converter (DAC) to convert numerical values into a physical analog voltage or current output, followed by an anti-aliasing lowpass or bandpass filter.

5.1 Strengths and Weaknesses

The strengths and weaknesses of the direct digital synthesis method are primarily dictated by the high-speed DAC used to convert the precise sine wave numerical quantities into a physical voltage or current. Aside from this key block, it is possible to make the supporting computational blocks nearly *perfect*. Ultimately, the DAC determines what spurious performance is possible and its power consumption is the dominant factor in the total power dissipation for a given DDS.

Direct digital synthesis is unsurpassed when it comes to

- High-speed frequency or phase switching
- Wideband phase and or frequency modulation capability
- Phase linearity and continuity
- Linear frequency sweeping

With the availability of very high performance 12- and 14-bit DACs now, the range of applications where direct digital synthesis is a viable alternative has increased substantially. The technique is, however, still suboptimal compared to other available methods when it comes to

- Low spurious performance
- Lowest power consumption
- Wideband frequency coverage

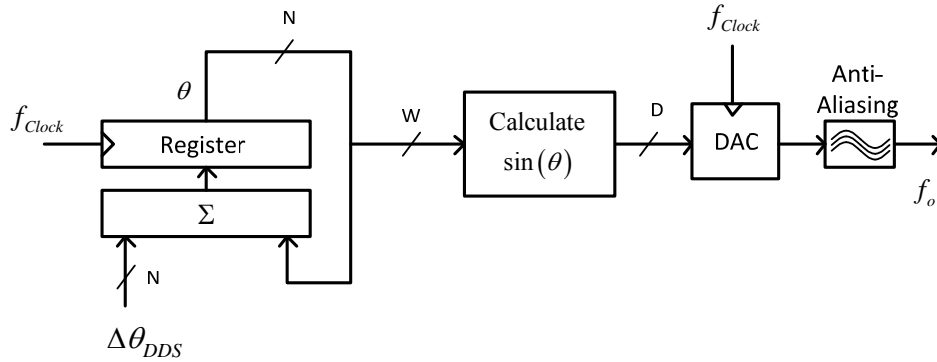


Figure 5-1 Basic direct digital synthesizer using a sampling rate of f_{Clock} Hz and input frequency word value $\Delta\theta_{DDS}$ used to create an output sine wave of f_o Hz

5.3 Direct Digital Synthesis Fundamentals

Direct digital synthesis fundamentally involves time-sampled signals as suggested in Figure 5-2.² The idealized DDS output can be thought of as creating the signal

$$d(t) = \sum_{n=-\infty}^{+\infty} v(nT_{Clock})h(t-nT_{Clock}) \quad (5.2)$$

where $h(t)$ represents the ideal zero-order sample-and-hold function,³ T_{Clock} is the time between samples, and $v(t)$ is an ideal sine wave given by

$$v(t) = \sin(2\pi f_o t) \quad (5.3)$$

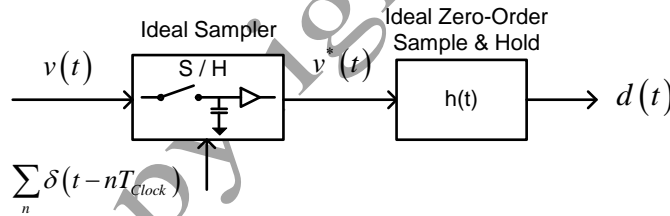


Figure 5-2 Idealized mathematical perspective for the DDS

The Laplace transform of $d(t)$ is given by⁴

$$D(f) = \exp(-j\pi f T_{Clock}) \frac{\sin(\pi f T_{Clock})}{\pi f T_{Clock}} \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} c_m \delta(f - n f_{Clock} - m f_o) \quad (5.4)$$

where

² [6], Appendix 4D and [7], Section 7.3.

³ $h(t)$ is unity for $0 \leq t < 1$ and is otherwise equal to zero.

⁴ [6], Appendix 4D and [7], Section 7.3.

$$x = \left\{ a f_{Clock}, \frac{1-a}{4} f_{Clock}, \frac{1+a}{4} f_{Clock}, \frac{1-a}{3} f_{Clock}, \frac{1+a}{3} f_{Clock}, \frac{1-a}{2} f_{Clock} \right\} \quad (5.11)$$

The form for the adjacent terms is clearly given by $(1-a)f_{Clock}/k$ and $(1+a)f_{Clock}/k$.

Referring to Figure 5-8, frequency regions that are sufficiently void of 2nd and 3rd harmonic signals are given by three regions: $[x_1, x_2]$, $[x_3, x_4]$, $[x_5, x_6]$. So long as the DAC does not exhibit appreciable harmonic distortion above 3rd order, these frequency regions can be used to create virtually spurious-free output signals.

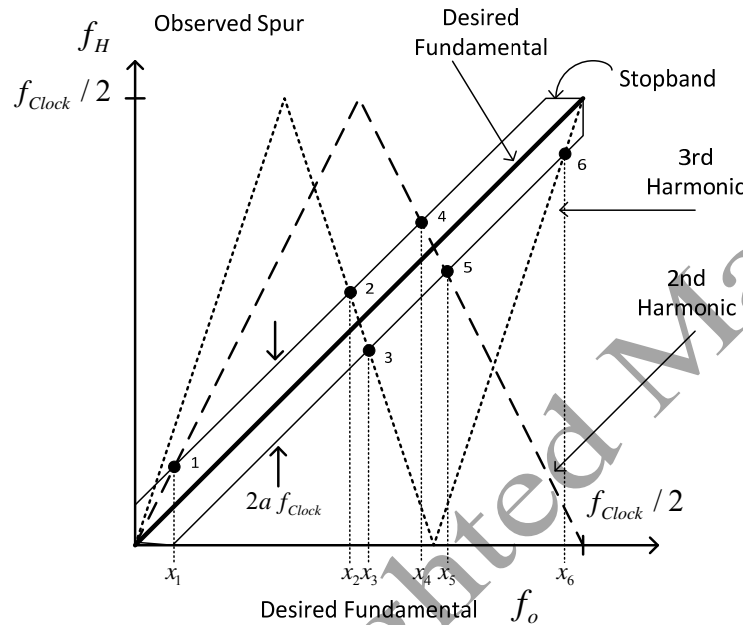


Figure 5-8 Augmented diagram with tracking stopband region shown. The x -axis values are given by (5.11).

In the special case where 3rd-order distortion is negligible and only 2nd-order harmonic distortion is significant, the two *clean* frequency regions are given by $[x_1, x_4]$ and $[x_5, f_{Clock}/2]$. Similarly, if the 2nd-order distortion is negligible and only 3rd-harmonic distortion is significant, the two clean frequency regions are given by $[a f_{Clock}/2, x_2]$ and $[x_3, x_6]$.

These concepts can be extended to higher-order harmonic distortion cases of course, but not without substantially greater complexity. In the case where the DDS is followed with an arbitrarily small bandwidth filter compared to f_{Clock} , there are still frequencies at which harmonic distortion will be potentially problematic (e.g., $f_o = f_{Clock}/k$ for k an arbitrary integer).

Second- and third-order output spurious frequencies can be avoided by operating in the three *clean* regions shown in Figure 5-8 and denoted by $[x_1, x_2]$, $[x_3, x_4]$, $[x_5, x_6]$.

In general, harmonic distortion is best avoided using the methodology described in this section or in Section 5.9. Otherwise, appreciable harmonic distortion is very difficult to combat unless a substantial amount (e.g., $1/4$ of full-scale) of dithering is used. Normally, this dithering is applied outside the usage-bandwidth and subsequently suppressed with additional filtering. This method and others are discussed later in Section 5.10.

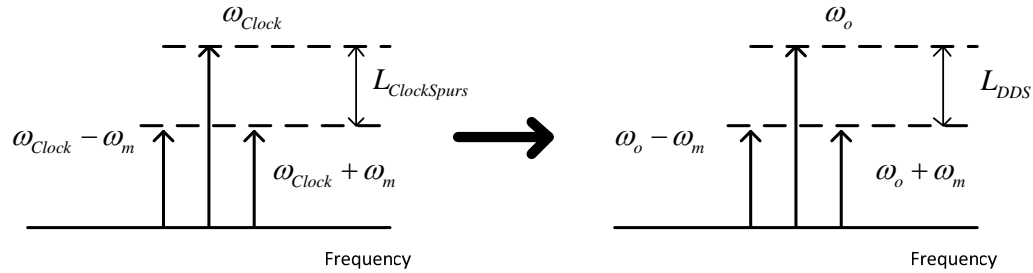


Figure 5-9 Input DDS clock PM spurs are translated directly to the DDS output as PM spurs

5.5.1.2 Random Clock Jitter

The effect of random clock jitter on DAC performance with random data is treated in Section 5.9 of [6]. The analysis for a DAC sine wave output begins in much the same way except the data sequence is a deterministic sine wave rather than random data. An idealized DAC output is shown in Figure 5-10 where grossly exaggerated clock-jittered boundaries are also shown. The early or late DAC output transition points result in error pulses compared to the ideal output waveform like those shown in the lower portion of Figure 5-10.

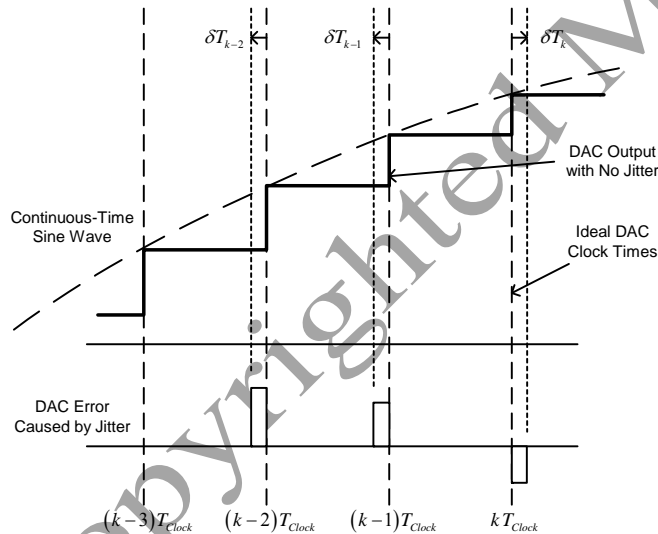


Figure 5-10 Idealized DAC output with time-jittered clock-times

The time-jittered DAC output can be represented by

$$V_M(t) = \sum_{k=-M}^M d_k \text{rect}(t - kT_{Clock} - \delta T_k, t - kT_{Clock} - T_{Clock} - \delta T_{k+1}) \quad (5.21)$$

where $\text{rect}(t_1, t_2) = 1$ for $t_1 \leq t < t_2$ but otherwise zero, and the d_k represent the ideal DAC sinusoidal sample values. For the sinusoidal output, the d_k are given by

$$d_k = \sin(2\pi f_o k T_{Clock}) \quad (5.22)$$

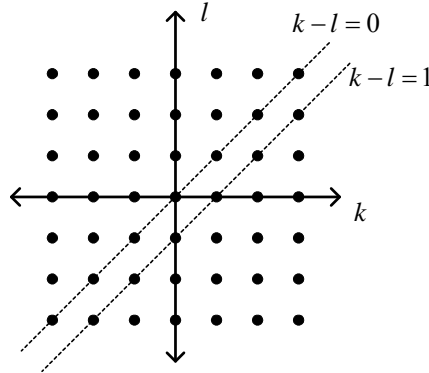


Figure 5-12 Diagram showing that the number of indices there $k - l = n$ is given by $2M + 1 - |n|$. This diagram corresponds to $M = 3$.

The autocorrelation function $R_v(n)$ is defined as

$$R_v(n) = E[v_k v_{k-n}]$$

$$= E \left\{ \begin{array}{l} 4 \sin^2(\pi f_o T_{Clock}) \cos \left[2\pi f_o \left(k - \frac{1}{2} \right) T_{Clock} \right] \times \\ \cos \left[2\pi f_o \left(k - n - \frac{1}{2} \right) T_{Clock} - \pi f_o T_{Clock} \right] \times \delta T_k \delta T_{k-n} \end{array} \right\} \quad (5.32)$$

After bringing the expectation operator inside the braces, applying some basic trigonometry, and making use of (5.25),

$$R_v(n) = 2 \left[\frac{T_{Clock} \sin(\pi f_o T_{Clock})}{2\pi} \right]^2 \cos(2\pi f_o T_{Clock}) R_{\delta\theta}(n) \quad (5.33)$$

In summary then, when $R_{\delta\theta}(n)$ is known, the power spectral density of the clock jitter related DAC output noise can be computed directly from (5.31). Since the phase noise (and clock jitter) random processes are assumed to be wide-sense stationary, $R_{\delta\theta}(n)$ can be obtained from the associated power spectral density of the clock phase noise using a Fourier transform.

An Example Calculation

The Lorentzian power spectral density frequently occurs for the phase noise of a DAC clock¹¹ and it is given by

¹¹ See Section 5.2 of [6].

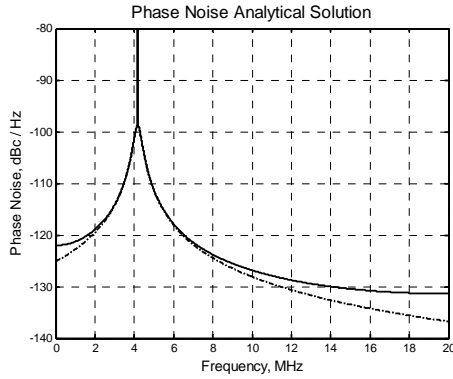


Figure 5-13 Calculated¹² sinusoidal-output DDS phase noise spectrum with $f_o = 4.18$ MHz

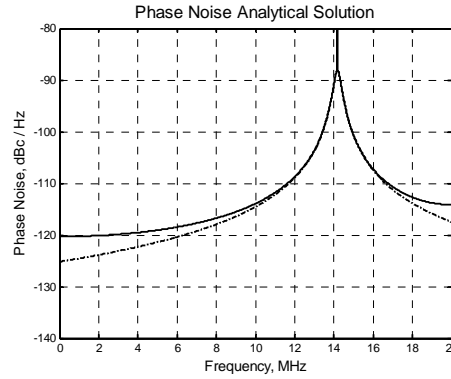


Figure 5-14 Calculated¹³ sinusoidal-output DDS phase noise spectrum with $f_o = 14.18$ MHz

5.5.2 Noise and Spurious Performance Related to Phase Truncation

Spectrum degradation due to phase-truncation related spurious outputs was a serious issue in early DDS designs due to finite-precision constraints. Deep sub-micron gate densities have largely made the finite-precision issue a thing of the past except possibly in ultra-high-speed DDS situations or when exceptionally low spurious requirements are in play. Algorithm advancements using dithering (Section 5.10) and noise-shaping techniques (Section 5.11) make it possible to all but eliminate these spurs even when relatively low-precision DACs are used. If these spurs are not properly attended to, however, they will always be present in the DDS output spectrum thereby making a mastery of this topic very worthwhile.

The high gate-count achievable in modern digital CMOS devices has made it possible to apply brute-force techniques to largely quell the phase truncation problem in modern systems. Phase accumulators using precisions of 24-bits or more are fairly typical and 10 to 14 bits are frequently retained for the θ to $\sin(\theta)$ conversion. Phase truncation related spurs are almost *always* present unless dithering or noise-shaping methods are used, however. It is therefore necessary to understand the level at which these may occur.

In the context of Figure 5-15, the DAC output sample sequence may be written as

$$s(n) = \sin\left(2\pi 2^B \left\lfloor \frac{F_r n}{2^{N-B}} \right\rfloor\right) + \varepsilon_n \quad (5.38)$$

where B is the number of phase accumulator bits truncated at the input to the $\sin(\theta)$ calculation (i.e., $B = N - W$), F_r is the (integer) frequency control word given by

$$F_r = \left\lfloor \frac{f_o}{f_{Clock}} 2^N \right\rfloor \quad (5.39)$$

¹² MATLAB script u16470_time_jittered_DAC_clock.m.

¹³ MATLAB script u16470_time_jittered_DAC_clock.m.

$$\begin{aligned} s_1(n) &= 2^u \langle n F_{r1} \rangle_T \\ s_2(\langle n F_{r1} \rangle_T) &= 2^u \langle n F_{r1} \rangle_T \end{aligned} \quad (5.57)$$

thereby making it possible to conclude that

$$s_1(n) = s_2(\langle n F_{r1} \rangle_T) \quad (5.58)$$

Similarly, it is true that

$$s_2(n) = 2^u \langle n \rangle_T \quad (5.59)$$

from (5.55), and

$$\begin{aligned} s_1(\langle n J \rangle_T) &= 2^u \langle \langle n J \rangle_T F_{r1} \rangle_T \\ &= 2^u \langle n J F_{r1} \rangle_T \\ &= 2^u \langle n \langle J F_{r1} \rangle_T \rangle_T \\ &= 2^u \langle n \rangle_T \end{aligned} \quad (5.60)$$

provided that $\langle J F_{r1} \rangle_T = 1$. When J satisfies this condition, s_2 and s_1 are also related by

$$s_2(n) = s_1(\langle n J \rangle_T) \quad (5.61)$$

The relationships given by (5.58) and (5.61) provide the means to rearrange one output sequence in terms of the other. This is especially helpful because the s_2 sequences for different values of u can consequently be viewed as the *basis waveforms* for DDS operation from which all of the other DDS quantities can be derived.

Equations (5.58) and (5.61) are identical with the earlier results given in (5.49) except that they also apply for non-zero values of u . The major value in recognizing the sequence rearrangement property between s_1 and s_2 is that the same is true of their respective discrete Fourier transforms represented by S_1 and S_2 . Consequently, the DFT of every possible frequency control word can be found by computing the DFTs for the N different basis sequences of s_2 (for $u \in [0, \dots, N - 1]$) in (5.54) corresponding to the different values of u in Figure 5-21, and applying the proper frequency-bin rearrangements to these DFT results.

It may be troubling that the discussions thus far have focused entirely on the phase sequence $\theta(n)$ rather than on the DDS output sequence $\sin[\theta(n)]$, but this is a minor issue which will be dealt with shortly.

5.5.2.3 Sequence Rearrangement in the Frequency Domain

Just as the different time sequences s_1 and s_2 can be related to one another through the index rearrangements (5.58) and (5.61), similar relationships exist between their respective DFTs as

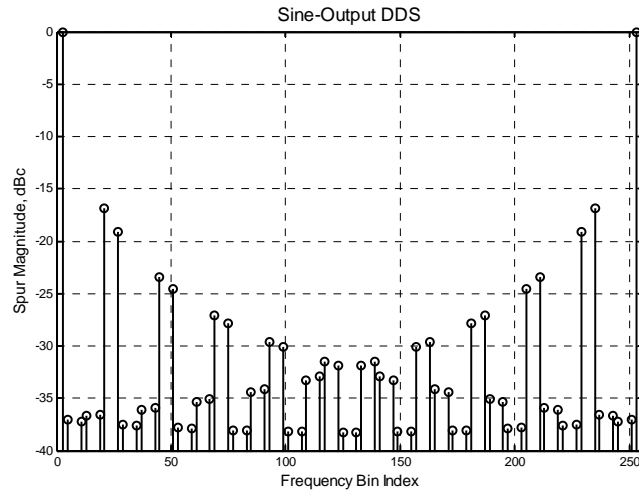


Figure 5-24 DDS output spectrum after the phase θ has been perfectly converted to $\sin(\theta)$. The sine-weighting strongly suppresses many of the spurs that would otherwise be present. The desired signal is present in bin #3 with its alias in bin #253. The worst-case undesired spur level is at about -16.9 dBc which is slightly less than the worst-case bound of -14.1 dBc predicted by (5.40).

The value of J corresponding to this choice for F_{r1} is $J = 171$. The output spectrum for the sinusoidal DDS case is shown in Figure 5-24. The sinusoidal shaping by $g(\cdot)$ in (5.70) suppresses many of the spur levels but the significant phase truncation present in this example still produces significant spurious terms as shown.

Example II: $N = 12$, $W = 8$, $F_{r1} = 723$

The spurious spectrum for this DDS case is shown in Figure 5-25. There are four distinct spurs that are noticeably higher than the others but a substantial number remain at about -60 dBc. These spur levels are identical to those of the basis frequency case $F_{r2} = 1$ shown in Figure 5-26.

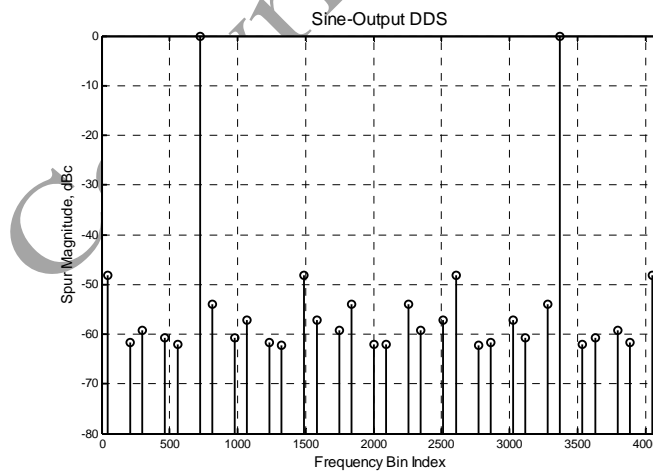


Figure 5-25 DDS output spectrum showing the worst-case spur level at about -48 dBc which is again consistent with the bound of -44.1 dBc predicted by (5.40). $N = 12$, $B = 4$, $F_{r1} = 723$.

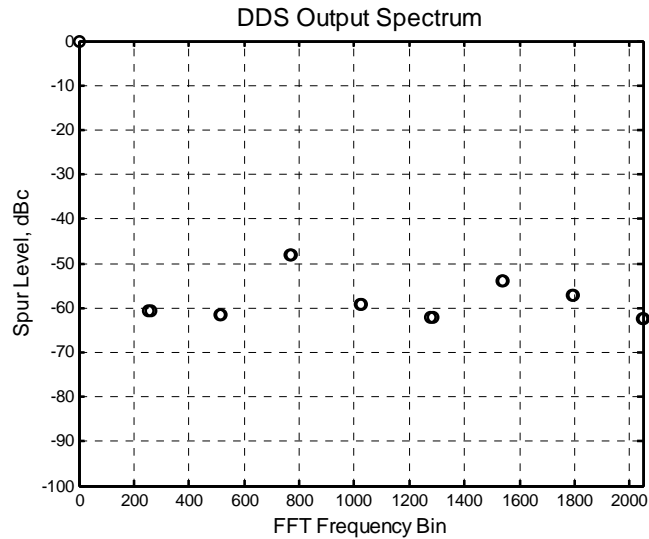


Figure 5-27 DDS output spectrum for $N = 12$, $W = 8$, and $D = 16$.²⁰ The only spurs visible are phase truncation related spurs and the worst-case spur level is about -48 dBc as predicted by (5.40).

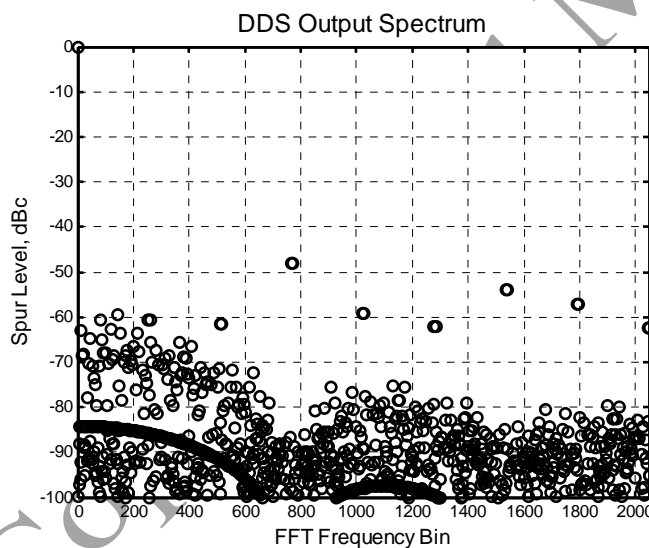


Figure 5-28 DDS output spectrum for $N = 12$, $W = 8$, and $D = 6$.²¹ Many more discrete spurs have appeared due to the smaller number of DAC bits used, but the magnitude and position of the phase truncation related spurs remain unchanged compared to Figure 5-27.

²⁰ From calling `u17612_dac_spur_check(3, 12, 4, 16)`.

²¹ From calling `u17612_dac_spur_check(3, 12, 4, 8)`.

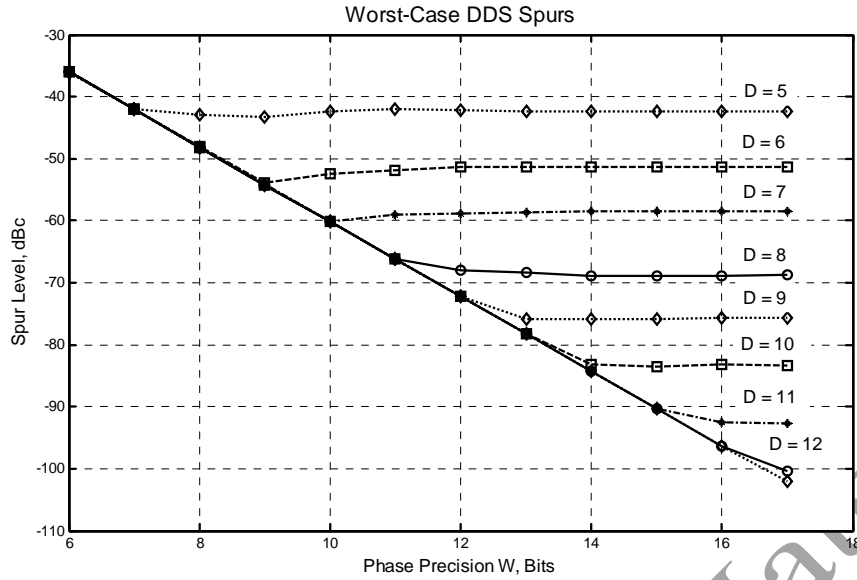


Figure 5-31 Worst-case DDS spur level versus number of phase bits W and number of DAC bits D .²⁴ (DAC input values truncated to D -bits.)

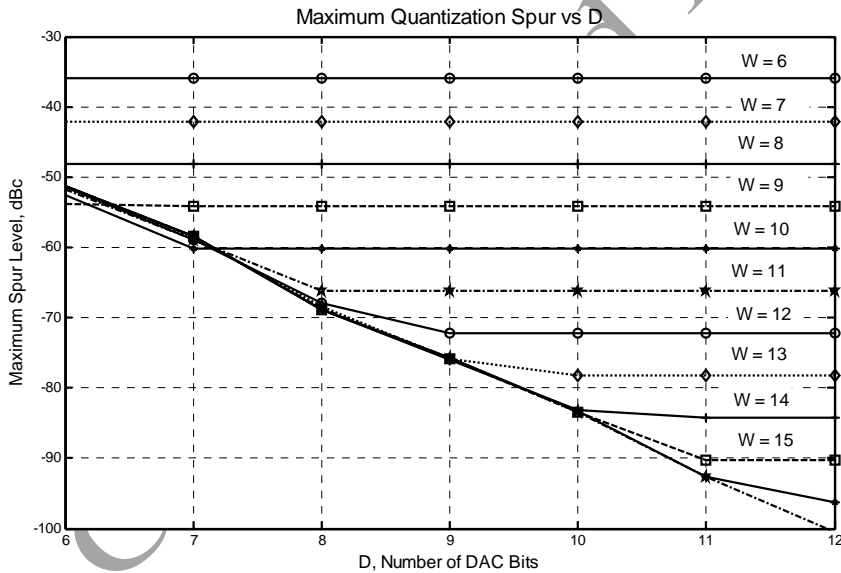


Figure 5-32 Worst-case DDS spur level versus number of phase bits W and number of DAC bits D (as in Figure 5-31) but with D as the independent variable.²⁵ (DAC input values truncated to D -bits.)

²⁴ From u16539_dac_quantization_errors.m.

²⁵ From u16539_dac_quantization_errors.m.

$$f(x) = \sin\left(\frac{\pi}{2}x\right) - \frac{\pi}{2}x \quad (5.79)$$

Once $f(x)$ has been computed using (5.79), it is a simple matter to obtain the sine value by adding the $(\pi/2)x$ quantity back in.

The CORDIC method is a powerful method that can be used to compute a host of transcendental functions including sine and cosine as discussed in the section that follows.

5.6.1 CORDIC Methods

The Coordinate Rotation Digital Computer method published by Volder [19] is commonly referred to as the CORDIC method. The method can be used to directly compute the functions shown in Table 5-5. The CORDIC method can also be extended to compute natural logarithms based upon the identity

$$\log_e(\alpha) = 2 \tanh^{-1}\left(\frac{\alpha-1}{\alpha+1}\right) \quad (5.80)$$

Table 5-5 CORDIC-computable functions

$\sin(\theta)$	$\cos(\theta)$	$\tan^{-1}(x)$	\times	\div
$\sinh(\theta)$	$\cosh(\theta)$	$\tanh^{-1}(x)$		
$\tan^{-1}\left(\frac{y}{x}\right)$	$\sqrt{x^2 + y^2}$	$\sqrt{x^2 - y^2}$	$y+xz$	$e^z = \sinh(z) + \cosh(z)$

The CORDIC method can be derived using the general (Givens) rotation of the coordinate pair (x, y) by the angle ϕ as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (5.81)$$

The rotation is length-preserving since the determinant of the matrix is one. If (x, y) lies on the unit-circle, (5.81) is nothing more than the simple trigonometric identity for the sum of two angles as shown in Figure 5-33. It is convenient to rewrite (5.81) as

$$\begin{aligned} x' &= \cos(\phi)[x - y \tan(\phi)] \\ y' &= \cos(\phi)[y + x \tan(\phi)] \end{aligned} \quad (5.82)$$

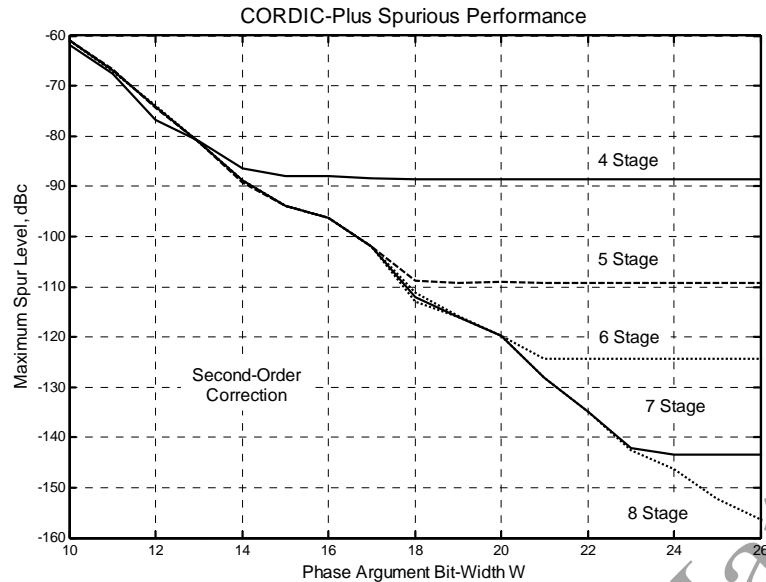


Figure 5-37 Worst-case spur level for CORDIC using the second-order residual phase rotation method described in this section.²⁷ These results are markedly better than the first-order method described in the previous section for a given number of CORDIC iterations.

This discussion about the CORDIC method has now come full-circle in that it has led back to using a fairly small look-up table to deal with the coarse portion of the phase, combined with a Taylor-series based final phase rotation.

5.7 D-to-A Converter Imperfections

DAC imperfections can be grouped into the following categories:

- Codeword-to-voltage (or current) precision issues
- Output sample-and-hold / buffer amplifier imperfections including harmonic distortion, finite slew-rate, glitches, and finite bandwidth
- Circuit-related noise
- Aperture jitter
- Other higher-order degradations

These imperfections manifest themselves quite differently across the many different DAC design architectures possible. Generally speaking, however, DAC performance is most frequently described using the terminology below [6]:

Differential Nonlinearity (DNL) – the normalized error between any two adjacent converter codes with respect to an ideal LSB based on the full-scale range of the converter. The DNL must always be less than unity in order for the converter to be considered monotonic.

²⁷ MATLAB script u16455_cordic2.m with an input parameter of 4.

The unequal rise and fall times consequently lead to a small gain error that is equal for every output pulse thereby averting the nonlinear pulse area behavior present in Figure 5-43. This approach forces the DAC to be operated at a higher output clock rate, but the improvements in the output spectral quality usually outweigh this consideration.

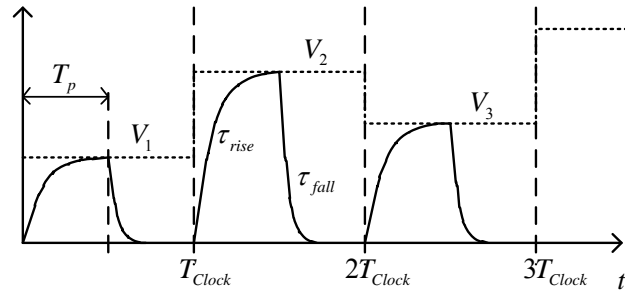


Figure 5-43 Return-to-zero DAC output pulses

5.7.2 Assessing DAC Imperfections

The techniques described in this chapter are very effective in suppressing spurious contributions due to computational issues, but DAC imperfections and harmonic distortion within any subsequent analog circuitry can still mar the DDS output spectral purity.

The most accurate way to assess DAC performance is in the frequency domain using a high-quality spectrum analyzer. Narrowing down DAC performance limitations to specific aspects of a DAC's detailed design (e.g., *INL*, *DNL*, clock jitter, harmonic distortion, etc.) is considerably more challenging and requires genuine detective work, however. Many high-performance data converters integrate self-calibration methods within themselves and the residual performance imperfections that remain are consequently more complicated to unravel. Some of the best diagnostic methods are still the traditional single-tone and two-tone testing methods.

Signal processing has certainly improved DAC performance characterization. A characterization approach that simultaneously estimates ADC and DAC performance using maximum-likelihood estimation is described in [22] for example. In the case of harmonic distortion, an arbitrary waveform generator technique is described in [23] that exploits relationships like those discussed in Section 5.4 to assess the underlying distortion terms and then reduce them substantially. DAC and ADC performance assessment has become a stand-alone technical discipline as performance levels and design sophistication have continued to advance.

5.8 DAC $\text{Sin}(x) / x$ Compensation

An ideal DAC creates aliasing and a $\text{sin}(x) / x$ shaping in the frequency domain as discussed earlier in the context of Figure 5-3. For an arbitrary continuous-time input signal $v(t)$ in Figure 5-2 having a Laplace transform of $V(s)$, the Laplace transform of the output signal is given by²⁹

²⁹ See Section 6.8.3 of [6] for a detailed derivation.

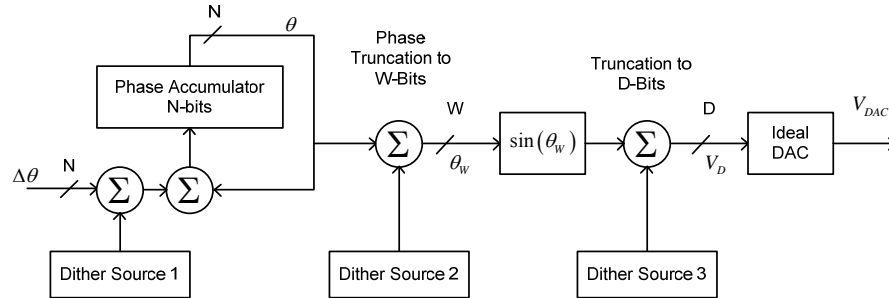


Figure 5-52 Possible dither insertion points within a DDS

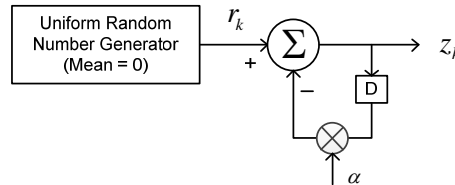


Figure 5-53 Dither source configured with a 1st-order differentiator / highpass filter making it suitable for dithering the frequency word input in Figure 5-52. Parameter α must be unity in order to remove any hint of random center frequency walking at the DDS output.

5.10.2 Phase Dithering

Phase dithering (using *Dither Source 2* in Figure 5-52) is particularly attractive because spurious tone magnitudes due to finite word-length effects in the phase representation are accelerated from the usual -6 dBc per phase bit to -12 dBc per phase bit [24]. This is at the expense of a small increase in output system noise. Phase dithering makes it possible to substantially reduce the size and complexity of the $\sin(\theta)$ computational hardware compared to a non-dithered DDS while retaining equal or better spurious performance.

In the context of Figure 5-52, W represents the number of phase-bits following truncation. The dither value that is added to θ prior to truncation is a uniformly distributed value within the span $[0, \Delta)$ where $\Delta = 2\pi 2^{-W}$ radians (or equivalently $[0, 2^W - 1]$ in binary form). Assuming no other imperfections in the DDS are present when this kind of phase dithering is included, the upper-bound for the output signal to (total) noise ratio is given by [25, 26]

$$SNR_U = 6.02W - 6.93 \text{ dB} \quad (5.104)$$

and a lower-bound is similarly given by

$$SNR_L = 6.02W - 8.18 \text{ dB} \quad (5.105)$$

In actual practice, it is more helpful to know what the C/N_o ratio is on a per-Hz basis. So long as $W < N - 6$ in Figure 5-52 so that the error sequences create more or less a continuum of spurious products, it suffices to use the lower-bound and state that

$$\frac{C}{N_o} = 6.02W - 8.18 + 10 \log_{10} \left(\frac{F_{Clock}}{2} \right) \text{ dBc/Hz} \quad (5.106)$$

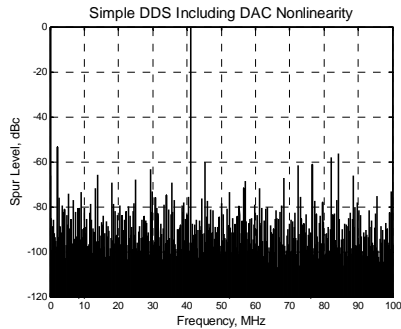


Figure 5-72 Spurious result corresponding to Figure 5-70 when 2nd through 7th harmonic performance is assumed to be -60 dBc for the DAC

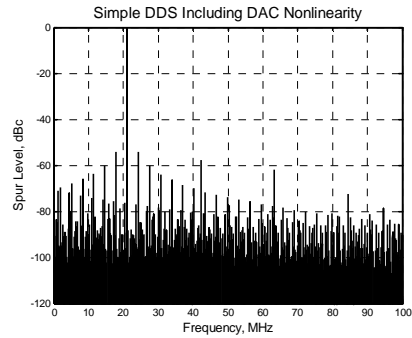


Figure 5-73 Spurious result corresponding to Figure 5-71 when 2nd through 7th harmonic performance is assumed to be -60 dBc for the DAC

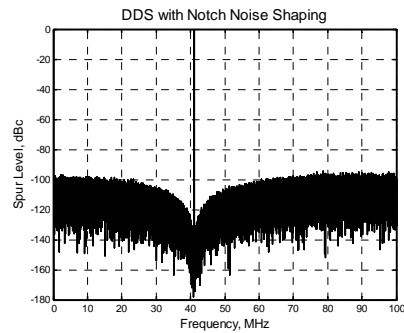


Figure 5-74 Noise shaping based on Figure 5-69 essentially eliminates the discrete spurious tones that are otherwise present in Figure 5-70, other conditions remaining constant (i.e., 9-bit ideal DAC, 200 MHz sampling rate, 41.1 MHz output)

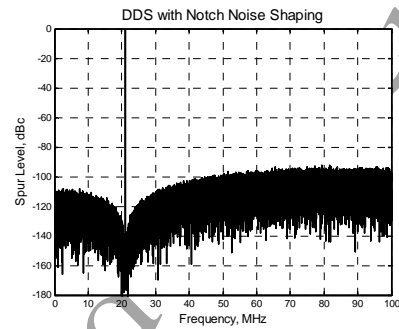


Figure 5-75 Noise shaping based on Figure 5-69 essentially eliminates the discrete spurious tones that are otherwise present in Figure 5-71, other conditions remaining constant (i.e., 9-bit ideal DAC, 200 MHz sampling rate, 41.1 MHz output)

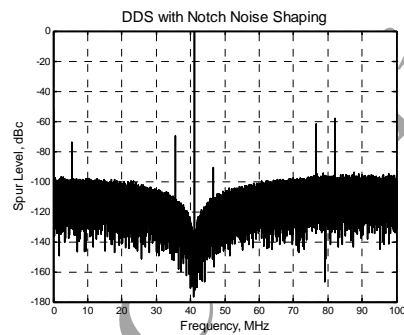


Figure 5-76 DDS output spurious performance⁵² corresponding to Figure 5-74 when -60 dBc harmonic distortion is present for the 2nd through 7th harmonics. The close-in spurious tones are at 35.6 MHz = 200 MHz - 4 F_{DDS} and 46.6 MHz = 6 F_{DDS} - 200 MHz.

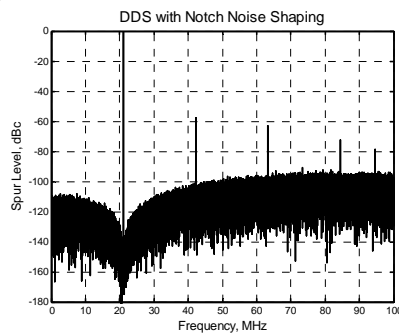


Figure 5-77 DDS output spurious performance⁵³ corresponding to Figure 5-75 when -60 dBc harmonic distortion is present for the 2nd through 7th harmonics. The first four spurs are simply harmonics of the fundamental frequency $F_{DDS} = 21.1$ MHz.

⁵² MATLAB script u16283_dds_noiseshaping.m.

⁵³ Ibid.