# Advanced Phase-Lock Applications: Frequency Synthesis

James A. Crawford

AM1 LLC

www.am1.us

# 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

Direct digital synthesis is attractive in one regard in that it is largely digital in nature aside from the output DAC. Lower frequency applications can incorporate Δ-Σ DACs into the design thereby making the entire DDS digital except for the output analog filtering. In general though, many factors influence the final design decisions. A grade-card to facilitate some of the performance tradeoffs is provided in Table 5-1.

**Table 5-1** Direct Digital Synthesis Compared to Alternative Synthesis Methods

| **Performance** | **DDS** | **PLL + DDS** | **Fractional-N PLL** | **Integer-N PLL** |
|---|---|---|---|---|
| Frequency Step Size | A | A | A | C |
| Frequency Switching Speed | A | B | C | D |
| Phase Linearity | A | A | B | F |
| Phase Modulation | A | B+ | B | F |
| Linear Frequency Sweep Capability | A | B | C | F |
| Close-In Spurious Performance | C | B | B+ | A |
| Far-Off Spurious Performance | D | B | B+ | A |
| Power Consumption | D | B | A– | A |
| Digital Complexity | C | C | B | A |
| RF/Analog Complexity | A | C | B | B |
| Overall Complexity | B | C | C+ | A |

Direct digital synthesis is generally the preferred approach for systems operating in the low frequency range up to several MHz and for high-speed switching or modulation capabilities like those required in high-performance radar or arbitrary waveform generation equipment. Other methods are generally better suited for RF and microwave applications.

## 5.2 Introduction to Direct Digital Frequency Synthesis

A simplified block diagram for a direct digital synthesizer is shown in Figure 5-1. The sampling rate[1] is $f_{Clock}$ Hertz. A digital accumulator is used to integrate the frequency word represented here by $\Delta\theta_{DDS}$ thereby resulting in a time-series of phase values $\theta_k$ that are ultimately converted to an output voltage (or current) sine wave using the sin( $\theta_k$ ) calculation and precision output DAC. If the lowest frequency Nyquist zone is used at the DAC output, the output frequency of the DDS is given by

$$f_o = \frac{\Delta\theta_{DDS}}{2\pi} f_{Clock} \text{ Hz} \tag{5.1}$$

The balance of this chapter looks at a wide range of issues pertaining to direct frequency synthesis.

---

[1] Small-case $f_o$ denotes frequency in Hz throughout this chapter whereas upper-case $F$ denotes an integer representation of frequency that is related to the DDS clock rate $f_{Clock}$ and phase accumulator total bit-width $N$ by $F = \lfloor (f_o / f_{Clock}) 2^N \rfloor$.

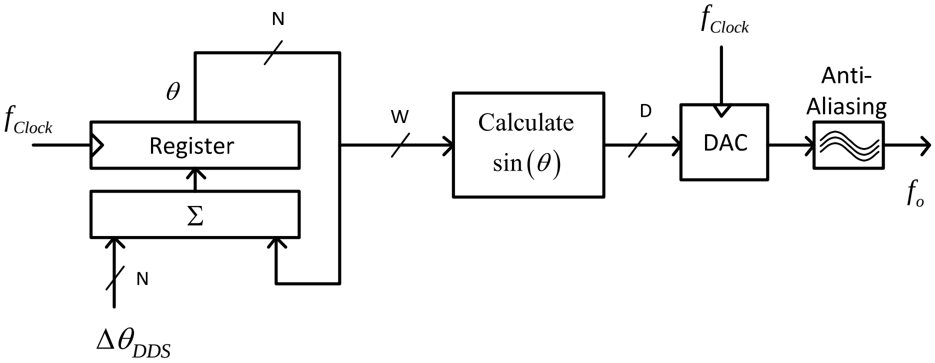


**Figure 5-1** Basic direct digital synthesizer using a sampling rate of $f_{Clock}$ Hz and input frequency word value $\Delta\theta_{DDS}$ used to create an output sine wave of $f_o$ Hz

### 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

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

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

$$v(t) = \sin(2\pi f_o t) \tag{5.3}$$

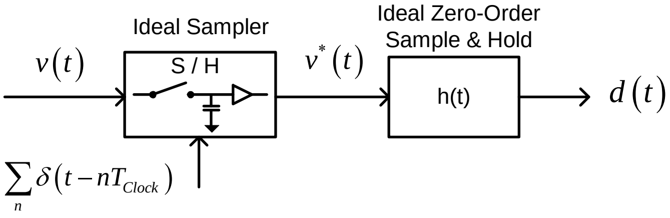


**Figure 5-2** Idealized mathematical perspective for the DDS

The Laplace transform of $d(t)$ is given by[4]

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

where

---

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

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

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

$$c_m = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} v(u) \exp(-j2\pi f_o m u)\, du \tag{5.5}$$

with $T_p = 1 / f_o$ and $f_{Clock} = 1 / T_{Clock}$. The result in (5.4) shows that it is possible to have a spectral component at every frequency given by

$$f = m\, f_o + n\, f_{Clock} \tag{5.6}$$

for arbitrary integers $m$ and $n$. The voltage sine wave $v(t)$ is calculated in a classical DDS by using a structure like that shown in Figure 5-1. If the otherwise ideal output DAC exhibits harmonic distortion, this can be modeled by using a modified input signal instead of (5.3) as

$$v(t) = \sum_{k=1}^{K} a_k \sin(2\pi k\, f_o t) \tag{5.7}$$

where the $a_k$ coefficients represent the harmonic distortion amplitudes. This ultimately changes the $c_m$ values computed from (5.5), but the frequencies at which spectral components can be present (5.6) remain unchanged.

The resultant DDS output spectrum consequently contains many discrete tones at the frequencies given by (5.6) with their corresponding amplitudes given by (5.5) and further weighted by the sin($x$) / $x$ function appearing in (5.4). An example output spectrum (less the output anti-alias lowpass filtering) is shown in Figure 5-3 where the fundamental output frequency is $f_o$ = 112 MHz and the clock frequency is $f_{Clock}$ = 300 MHz.

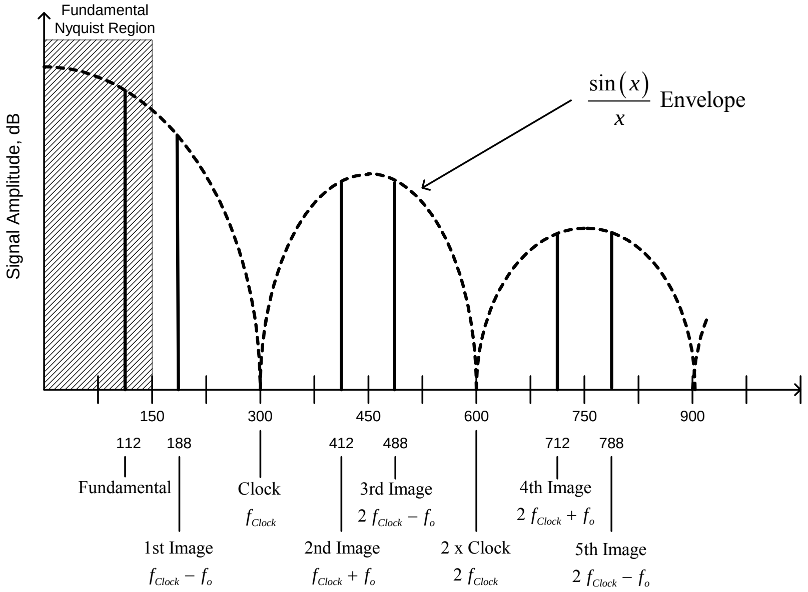


**Figure 5-3** Example output spectrum from an ideal DDS.[5]

---

[5] After Figure 2-1 of [8].

If DAC-related harmonic distortion had also been present as suggested in Figure 5-4, Figure 5-3 would have contained many more discrete tones. Consequently, the amount of harmonic distortion present can be inferred from these additional spectral tones as discussed later in Section 5.7. The most commonly used terminology to characterize DAC linearity performance is also discussed in that section.

Equation (5.6) is foundational for predicting the DDS output spurious frequencies.

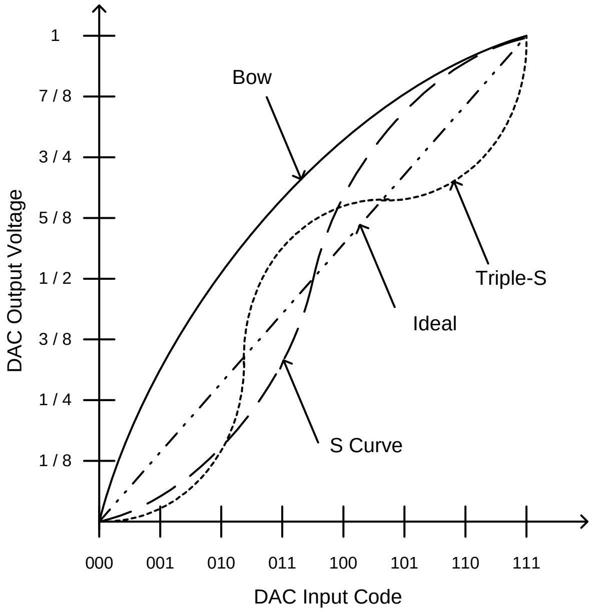


**Figure 5-4** Common DAC nonlinear transfer function characteristics[6]

### 5.4 Systematic Frequency Planning for Design and DDS Usage

Careful frequency planning is always required for high performance frequency synthesis and this statement is equally applicable to direct digital synthesis. DAC-related harmonic distortion is the primary bane for direct digital synthesis. Digital images come into play of course like those shown in Figure 5-3, but these are dictated by the Nyquist theorem and are simple to predict. In the case of harmonic distortion, however, even very high-order harmonic distortion terms can be heterodyned into any Nyquist zone by way of (5.6) thereby leading to unwanted in-band spurious products.

When the fundamental DDS output frequency $f_o$ is much less than $f_{Clock}$ and the maximum-order harmonic is reasonably small, all of the output harmonics appear without any aliasing involved (i.e., $n = 0$ in (5.6)) as shown in Figure 5-5. As $f_o$ is increased, harmonics that would otherwise appear at frequencies higher than $f_{Clock} / 2$ in the absence of aliasing are instead aliased per (5.6) into the fundamental Nyquist region ( $0 \leq f_o \leq f_{Clock} / 2$ ) as shown in Figure 5-6. (Note: Aside from the $\sin(x) / x$ shaping factor, these same spurious terms are also identically repeated in the $k^{th}$ Nyquist zone which is defined as [ $(k-1) f_{Clock} / 2 \leq f_o \leq k f_{Clock} / 2$ ]. )

---

[6] After Figure 8 of [4].

Harmonic distortion up to the $K^{th}$ harmonic is fairly easy to measure in the laboratory for DDS output frequencies $f_o < f_{Clock} / (2K)$ because they have the appearance shown in Figure 5-5. For higher output frequencies, however, the harmonic distortion levels normally change as several second- and third-order imperfections become more significant over frequency. This consequently makes separate harmonic distortion assessment equally important for higher $f_o$ values where aliasing must be factored in.

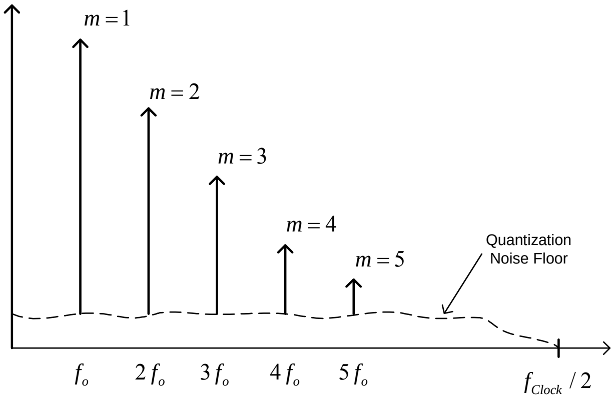


**Figure 5-5** DDS harmonic distortion example where only the first five harmonics of the fundamental output frequency $f_o$ have an appreciable amplitude and where $f_o \ll f_{Clock}$ so that frequency aliasing is less apparent. In the context of (5.6), $n \equiv 0$.

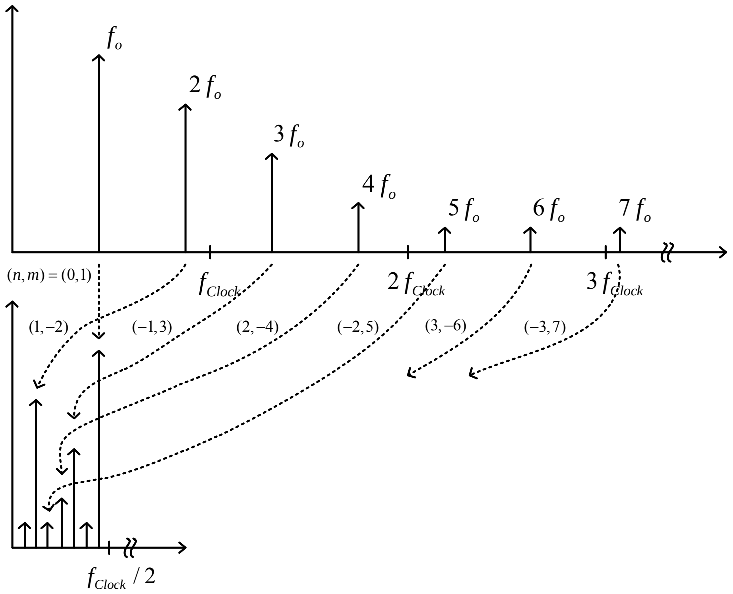


**Figure 5-6** As the DDS fundamental output frequency $f_o$ is increased, harmonics of $f_o$ are aliased per (5.6) into the fundamental Nyquist frequency range spanning from DC to $f_{Clock} / 2$.

A graphical means for frequency planning based upon (5.6) is presented in [5] and a portion of that discussion is reconstructed here. Based upon (5.6) and constraining the frequency region of interest to the fundamental Nyquist zone, the desired DDS output frequency is limited to

$$0 \le f_o \le \frac{f_{Clock}}{2} \tag{5.8}$$

The second-harmonic of $f_o$ falling within the fundamental Nyquist zone is given by

$$f_{H2} = \begin{cases} 2f_o & \text{for } 0 < f_o < f_{Clock}/4 \\ f_{Clock} - 2f_o & \text{for } f_{Clock}/4 < f_o < f_{Clock}/2 \end{cases} \tag{5.9}$$

The third-harmonic of $f_o$ falling within the fundamental Nyquist zone is given by

$$f_{H3} = \begin{cases} 3f_o & \text{for } 0 < f_o < f_{Clock}/6 \\ f_{Clock} - 3f_o & \text{for } f_{Clock}/6 < f_o < f_{Clock}/3 \\ 3f_o - f_{Clock} & \text{for } f_{Clock}/3 < f_o < f_{Clock}/2 \end{cases} \tag{5.10}$$

These results are shown graphically in Figure 5-7. In general, the fundamental Nyquist zone will always contain harmonics of the desired DDS output frequency $f_o$, and if their levels are higher than desired, the DDS must be followed with a suitable bandpass filter and or other suppression techniques pursued as discussed in this chapter.

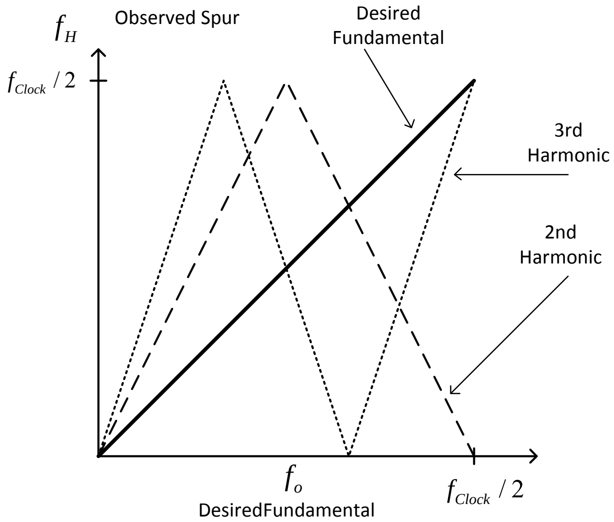


**Figure 5-7** Graphical representation of the 2nd and 3rd harmonic DDS output frequencies based upon (5.9) and (5.10)

In the case where the DDS is followed by a bandpass filter, the filter can be assumed to have a $-A_{STOP}$ dB stopband and a bandwidth of $B_{FIL} = 2\, a\, f_{Clock}$ symmetrically located with respect to the desired DDS output frequency $f_o$. The bandpass filter will attenuate any spurious frequency falling more than $a\, f_{Clock}$ Hertz away from $f_o$ by at least $A_{STOP}$ dB. Figure 5-7 can be augmented with the filter stopband limits as shown in Figure 5-8. As given in [5], the different *x*-axis points in the figure are given by

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

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.

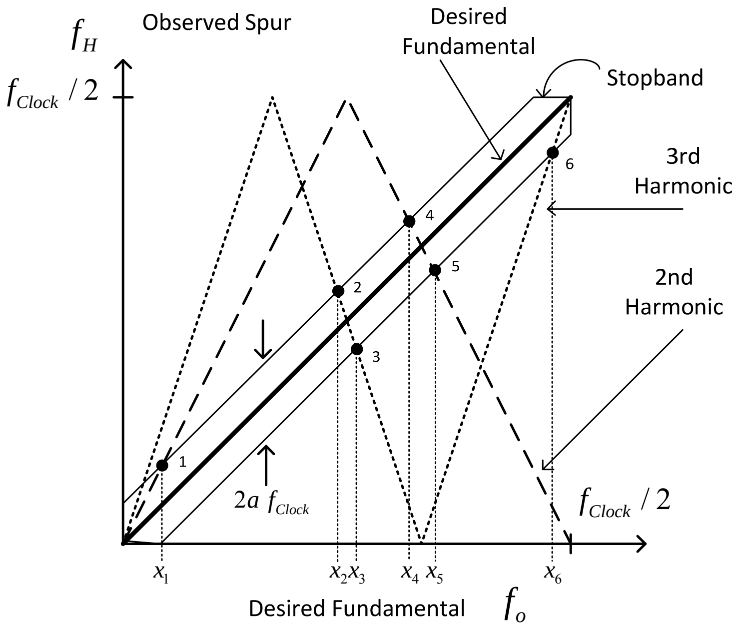


**Figure 5-8** Augmented diagram with tracking stopband region shown. The $x$-axis values are given by (5.11).

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., ¼ 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.

## 5.5 DDS Output Spectrum

The previous section dealt with a frequency planning approach that makes it possible to avoid 2nd- and 3rd-order harmonic distortion related spurious frequencies at the DDS output. Other key high-level factors that directly affect the output DDS spectrum are discussed in this section.

Characterization of the DDS output spectrum falls into two different camps depending upon whether noise-shaping techniques are used or not. No noise-shaping is considered for the discussions in this section. Noise shaping methods are addressed separately in Section 5.11.

### 5.5.1 DDS Input Clock-Jitter Considerations

Input clock spectral imperfections ( i.e., jitter ) will directly affect the spectrum at the DDS output. Two separate cases corresponding to deterministic jitter and random jitter are considered here. It so happens that the results in this section apply whether noise-shaping techniques are used in the DDS or not.

#### 5.5.1.1 Deterministic Clock Jitter (Sinusoidal PM)

The DDS input clock can be represented in this case by

$$v_{Clock}\left(t\right)=\sin\left[\omega_{Clock}t+\Delta\theta\sin\left(\omega_m t\right)\right] \tag{5.12}$$

where the clock input radian frequency is given by $\omega_{Clock}$, the associated peak phase deviation is given by $\Delta\theta$ radians, and the radian modulation frequency is represented by $\omega_m$ ( $<< \omega_{Clock}$ ). The phase modulation causes the otherwise perfect zero-crossings of the clock to be jittered in time. To facilitate this discussion, denote the $k^{th}$ zero-crossing time as

$$t_k = k\,T_{Clock}+\delta T_k \tag{5.13}$$

where the nominal DDS clock frequency is given by $f_{Clock} = 1 / T_{Clock}$. From (5.12), the zero-crossing times must satisfy

$$\omega_{Clock}\left(k\,T_{Clock}+\delta T_k\right)+\Delta\theta\sin\left[\omega_m\left(k\,T_{Clock}+\delta T_k\right)\right]=2k\pi \quad \text{for } \left(k\geq 0\right) \tag{5.14}$$

In the case of no clock-jitter, clearly $\omega_{Clock}\, k\, T_{Clock} = 2k\pi$ and this observation can be used to transform (5.14) into

$$\omega_{Clock}\delta T_k+\Delta\theta\sin\left[\omega_m\left(k\,T_{Clock}+\delta T_k\right)\right]=0 \tag{5.15}$$

In general, $|\omega_m \delta T_k| << 1$ and using the small-angle approximations[7] for sin( ) and cos( ) in (5.15) produces the end-result

$$\delta T_k \cong -\frac{\Delta\theta}{\omega_{Clock}}\sin\left(\omega_m k\, T_{Clock}\right) \tag{5.16}$$

The primary reason for expressing the zero-crossing variations in the time domain via (5.16) is to show that *time* is never scaled in the real world. Consequently, if the zero-crossings of the DDS input clock are being moved by $\delta T_k$, the same must be true for zero-crossings at the DDS output. Ideally, the DDS output sinusoid is given by

$$v_{DDS}\left(t\right) = \sin\left(2\pi f_o t\right)\Big|_{t=kT_{Clock}} \tag{5.17}$$

but in fact, $t$ must be replaced by $t_k = kT_{Clock} + \delta T_k$. This modifies the ideal DDS output represented by (5.17) to

$$\begin{aligned} v_{DDS}\left(t_k\right) &= \sin\left[2\pi f_o\left(k\,T_{Clock} + \delta T_k\right)\right] \\ &= \sin\left(2\pi f_o k\,T_{Clock} + \phi_k\right) \end{aligned} \tag{5.18}$$

where $\phi_k$ represents the phase modulation on the DDS output sine wave given by

$$\phi_k = -\Delta\theta\frac{\omega_o}{\omega_{Clock}}\sin\left(\omega_m k\,T_{Clock}\right) \tag{5.19}$$

Note that (5.19) has the same form as the original phase-modulated clock in (5.12). Therefore, if the DDS input clock at frequency $f_o$ has discrete PM spurs with a frequency-offset of $f_m$ and dBc level of $L_{ClockSpurs}$, these spurs are manifested at the DDS output as PM spurs on the desired sinusoidal output with the same offset frequency $f_m$ but with the modified dBc level given by

$$L_{DDS} = L_{ClockSpurs} + 20\log_{10}\left(\frac{f_o}{f_{Clock}}\right) \text{ dBc} \tag{5.20}$$

This result is shown graphically in Figure 5-9.

In general, the DDS acts like a frequency divider with a divide-ratio of $N_{DDS} = f_o / f_{Clock}$. In this respect, the result given by (5.20) follows the same well-known spurious improvement delivered by a divider, namely 20 $\log_{10}(N)$.

---

[7] This approximation is equivalent to ignoring the higher-order terms in the Bessel series expansion for (5.12) which is $v\left(t\right) = \sum_{n=-\infty}^{+\infty} J_{|n|}\left(\Delta\theta\right)\cos\left[\left(\omega_{Clock} + n\omega_m\right)t\right]$ from equation (3.4) in [6] and the $J_n$ represents an $n^{th}$-order Bessel function.

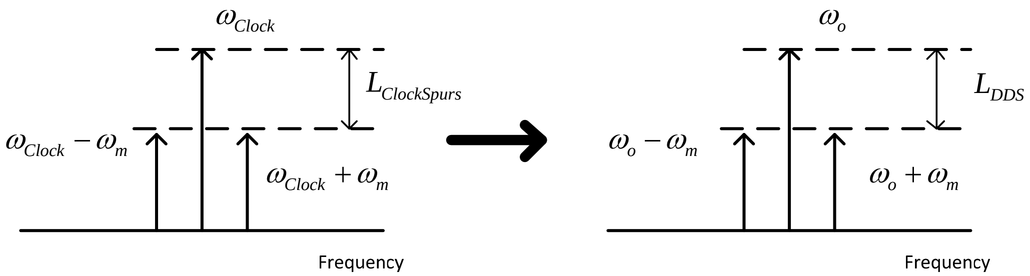


**Figure 5-9** Input DDS clock PM spurs are translated directly to the DDS output as PM spurs

### 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.

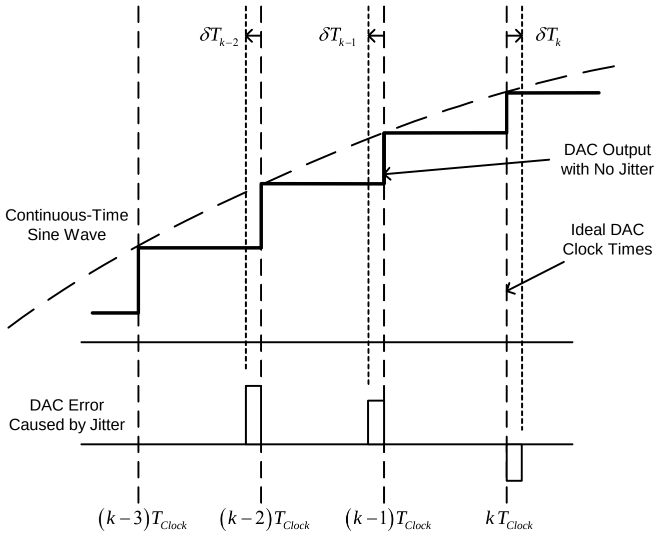


**Figure 5-10** Idealized DAC output with time-jittered clock-times

The time-jittered DAC output can be represented by

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

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

$$d_k = \sin\left(2\pi\, f_o k\, T_{Clock}\right) \tag{5.22}$$

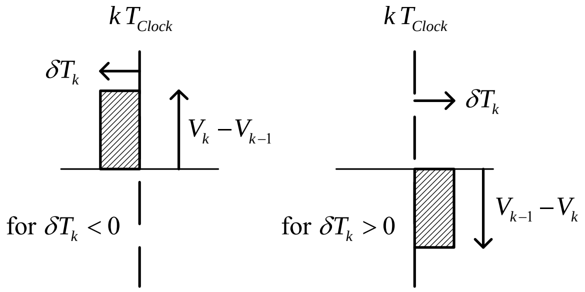


**Figure 5-11** Error pulses corresponding to Figure 5-10

The error pulses shown in Figure 5-11 have extremely small widths in time (i.e., $|\delta T_k| << T_{Clock}$) for any system of genuine interest and they can be modeled as a periodic sequence of ideal delta-functions with impulse areas ( which must be scaled by $T_{Clock}$ ) given by

$$u_k = -\left(d_k - d_{k-1}\right)\frac{\delta T_k}{T_{Clock}} \tag{5.23}$$

For the sinusoidal case (5.22), the difference can be expressed as

$$d_k - d_{k-1} = 2\sin\left(\pi f_o T_{Clock}\right)\cos\left(2\pi f_o k T_{Clock} - \pi f_o T_{Clock}\right) \tag{5.24}$$

If a time domain simulation for computing the $\delta T_k$ quantities is available, computing the associated power spectral density of the noise amounts to computing the squared-magnitude of the resultant fast Fourier transform of (5.23).

### A More Careful Look

A more in-depth look at the mechanics behind (5.23) can provide greater insight into this rather simple result. The autocorrelation function of the time jitter or associated clock phase noise is assumed to be known for this discussion. The two autocorrelation functions are related as

$$R_{\delta T}\left(\tau\right) = \left(\frac{T_{Clock}}{2\pi}\right)^2 R_{\delta\theta}\left(\tau\right) \tag{5.25}$$

When the jitter quantities are wide-sense stationary as assumed here, the autocorrelation functions are related to their respective power spectral densities through a Fourier transform.[8]

The power spectral density for a wide-sense stationary continuous-time random process is given by[9]

$$P\left(f\right) = \lim_{M\to\infty} \mathbf{E}\left\{\frac{1}{\left(2M+1\right)T_{Clock}}\left|L_M\left(j2\pi f\right)\right|^2\right\} \tag{5.26}$$

---

[8] A basic property for wide-sense stationary processes as discussed in Section 4A.1 of [6].

[9] Equation (2.94) from [6].

where **E** denotes statistical expectation, and $L_M$ represents the Fourier transform of the error signal over a time interval of ± $M\ T_{Clock}$. It is important to note this definition produces a two-sided spectral result for $P(f)$.

The Laplace transform for the error pulse-sequence shown in the lower portion of Figure 5-10 can be written as

$$L(s) = \sum_{k=-M}^{M} -\left(d_k - d_{k-1}\right) \frac{1-e^{-s\,\delta T_k}}{s} e^{-skT_{Clock}} \tag{5.27}$$

As noted earlier, | $\delta T_k$ / $T_{Clock}$ | << 1 makes it possible to approximate the fractional portion of (5.27) by

$$\frac{1-e^{-s\,\delta T_k}}{s} \approx \frac{1-\left(1-s\,\delta T_k\right)}{s} = \delta T_k \tag{5.28}$$

(This approximation was the key enabling approximation used in obtaining (5.23).) Not surprisingly, (5.27) can be closely approximated by

$$L(s) \cong \sum_{k=-M}^{M} \left[-\left(d_k - d_{k-1}\right)\delta T_k\right] e^{-skT_{Clock}} \tag{5.29}$$

where the bracketed quantity will be subsequently referred to as $v_k$. (The $v_k$ lack a factor of $1/T_{Clock}$ compared to the $u_k$ in (5.23) ) The squared-magnitude of (5.29) is then

$$\left|L(s)\right|^2 = \sum_{k=-M}^{M} \sum_{l=-M}^{M} v_k v_l e^{-s(k-l)T_{Clock}} \tag{5.30}$$

The double-sum is most easily handled by using contours along which $k - l$ is a constant as shown in Figure 5-12. Invoking this reasoning and using the wide-sense stationarity of the $v_k$ samples, the statistical expectation in (5.26) can be brought through the double-summation in (5.30) thereby resulting in[10]

$$P(f) = \frac{1}{T_{Clock}} \lim_{M\to\infty} \left\{ R_v(0) + 2\sum_{n=1}^{2M} \left(1 - \frac{n}{2M+1}\right) R_v(n) \cos\left(2\pi f\, nT_{Clock}\right) \right\} \tag{5.31}$$

which is again a two-sided power spectral density. The $R_v(n)$ quantity is the autocorrelation function of the $v_k$ samples. In this result, the sin( $x$ ) / $x$ factor which usually occurs with DAC outputs is absent because the pulse-widths in Figure 5-11 are so small compared to $T_{Clock}$.

---

[10] Interestingly, when $R_{\delta\theta}(n)$ is known exactly, retaining the $1-\frac{n}{2M+1}$ factor can lead to a small scaling inaccuracy even near the carrier unless M is extremely large. In the associated MATLAB code script u16470_time_jittered_DAC_clock.m, this term is set to unity for this reason.

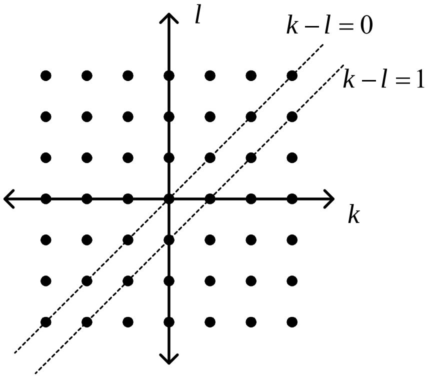


**Figure 5-12** Diagram showing that the number of indices there $k - l = n$ is given by $2M + 1 - |n|$. This diagram corresponds to $M = 3$.

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

$$\begin{aligned} R_v(n) &= \mathrm{E}\left[v_k v_{k-n}\right] \\ &= \mathrm{E}\left\{ \begin{aligned} &4\sin^2\left(\pi f_o T_{Clock}\right)\cos\left[2\pi f_o\left(k-\frac{1}{2}\right)T_{Clock}\right]\times \\ &\cos\left[2\pi f_o\left(k-n-\frac{1}{2}\right)T_{Clock} - \pi f_o T_{Clock}\right]\times \delta T_k\, \delta T_{k-n} \end{aligned} \right\} \end{aligned} \tag{5.32}$$

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

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

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 clock[11] and it is given by

---

[11] See Section 5.2 of [6].

$$\mathscr{L}(f) = \frac{L_o}{1 + \left( \frac{f}{f_c} \right)^2} \tag{5.34}$$

The total integrated phase noise for this two-sided phase noise spectrum is

$$\sigma_\theta^2 = \int_{-\infty}^{+\infty} \frac{L_o}{1 + \left( \frac{f}{f_c} \right)^2} df = \pi L_o f_c \text{ rad}^2 \tag{5.35}$$

The corresponding autocorrelation function is given by

$$R_\theta(\tau) = \pi L_o f_c \exp\left( -2\pi f_c |\tau| \right) \tag{5.36}$$

This autocorrelation function can be used for $R_{\delta\theta}$ in (5.33) and the power spectral density of the DAC output noise due to clock jitter computed using (5.31).

The two cases considered here have the following parameter values:

| | |
|---|---|
| $f_c$ | 200 kHz |
| $R_{\delta\theta}(0)$ | Corresponding to 5° rms integrated phase noise |
| $f_{Clock}$ | 40 MHz |
| $f_o$ | Two cases, 4.18 MHz and 14.18 MHz |

The computed output spectrums are shown in Figure 5-13 and Figure 5-14 where the noise spectrum clearly degrades as the DDS output frequency is increased. The dashed curve in both figures is present for strictly comparative purposes and is given by

$$\mathsf{L}_{Comp}(f) = 10\log_{10}\left[ \frac{L_o}{1 + \left( \frac{f - f_o}{f_c} \right)^2} \right] + 20\log_{10}\left( \frac{f_o}{f_{Clock}} \right) \frac{dBc}{Hz} \tag{5.37}$$

The output noise spectra always tend to flatten out near dc and near $f_{Clock}$ / 2 due to higher-order noise aliasing products as shown in the figures. Since the noise sidebands are not symmetric, the output noise contains both AM and PM noise contributions. In general, the computed results given by (5.31) match the *expected* results given by (5.37) in Figure 5-13 and Figure 5-14 quite well until fairly low noise levels are reached. In a real world application, the total integrated clock phase noise would probably be kept considerably smaller than in these two examples thereby improving the output noise sidebands proportionately.

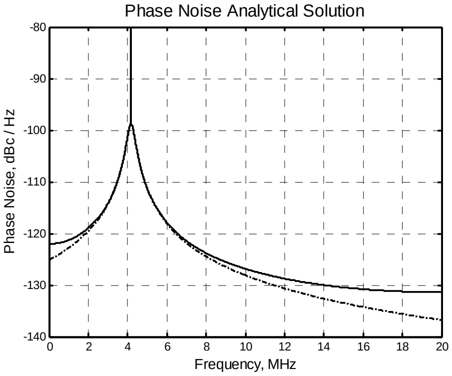


**Figure 5-13** Calculated[12] sinusoidal-output DDS phase noise spectrum with $f_o = 4.18$ MHz

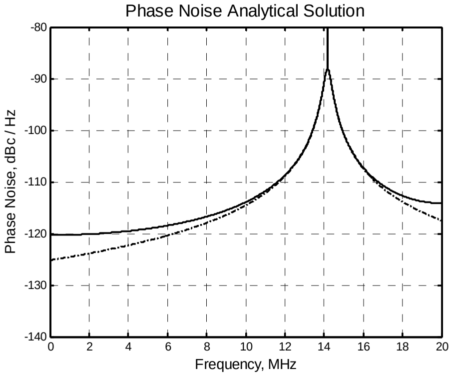


**Figure 5-14** Calculated[13] sinusoidal-output DDS phase noise spectrum with $f_o = 14.18$ MHz

## 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 $\theta$ 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

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

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

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

---

[12] MATLAB script u16470_time_jittered_DAC_clock.m.

[13] MATLAB script u16470_time_jittered_DAC_clock.m.

and $\varepsilon_n$ represents the additional noise due to the limited *D*-bit quantization of the output sine quantity. The notation $\lfloor x \rfloor$ denotes retaining only the integer portion of $x$. The DAC is assumed to be ideal without any quantization noise for this present discussion ( $\varepsilon_k = 0$ ).

The worst-case spur level is given by [7, 9]

$$\begin{aligned} L_{wcp} &= 6.02\left(B-N\right)+3.92 \quad \text{dBc} \\ &= -6.02W+3.92 \quad \text{dBc} \end{aligned} \tag{5.40}$$

None of the results discussed in this section include the sin( $x$ ) / $x$ roll-off of the output frequency terms due to the sample-and-hold inherently present within the DAC function, however, so additional allowances must be made in order to reflect this factor into what would be measured in the laboratory.

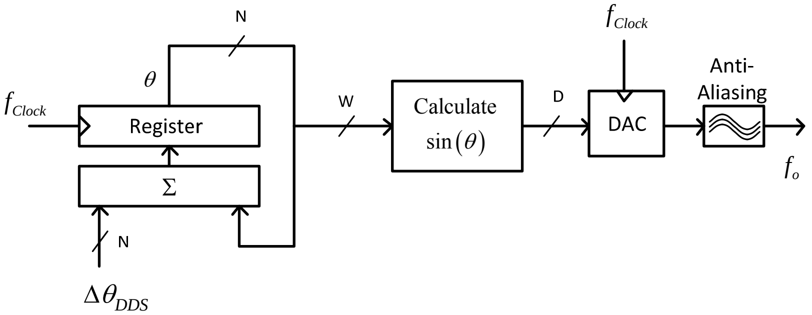


**Figure 5-15** Bit-widths in the general sinusoidal DDS. The number of bits truncated between the phase accumulator ( $N$ bits ) and the phase argument presented to the sin( $\theta$ ) calculation ( $W$ bits ) is represented by $B$ in the text.

The level of the maximum observed spur can be up to 3.92 dB lower than that given by (5.40) depending upon the specific output frequency being synthesized. The maximum observed spur level[14] for a specific output frequency is rigorously given by [7, 9]

$$L_{Spur}\left(F_r\right) = 6.02\left(B-N\right)+20\log_{10}\left\{\frac{\pi \gcd\left(F_r,2^B\right)2^{-B}}{\sin\left[\pi \gcd\left(F_r,2^B\right)2^{-B}\right]}\right\} \quad \text{dBc} \tag{5.41}$$

where $F_r = \lfloor ( f_o / f_{Clock} )\, 2^N \rfloor$. The maximum value for $L_{Spur}$ occurs when the greatest common divisor (gcd) between $F_r$ and $2^B$ is $2^{B-1}$ thereby resulting in (5.40). The 20log$_{10}$( ) portion of (5.41) is shown in Figure 5-16 to illustrate that the additional spur-level increase represented by this term occurs very infrequently for $B > 5$.

Two example output spectrums are shown in Figure 5-17 and Figure 5-18. In both cases, $f_{Clock}$ = 100 MHz, $N$ = 18, and $W$ = 10 are used. The first spectrum corresponds to $F_r$ = 7340 and the second spectrum corresponds to $F_r$ = 45298. As exhibited here, the spectral shape of the spurious patterns can vary dramatically, and the spur levels dwarf the level of any noise-like terms created by the phase truncation. Phase truncation almost always leads to a discrete line spectrum as discussed later in Section 5.5.2.1 because the truncation errors are highly correlated.

---

[14] Some have taken issue with this formula for small values of $B$ ( e.g., 1 ), notably [10] .

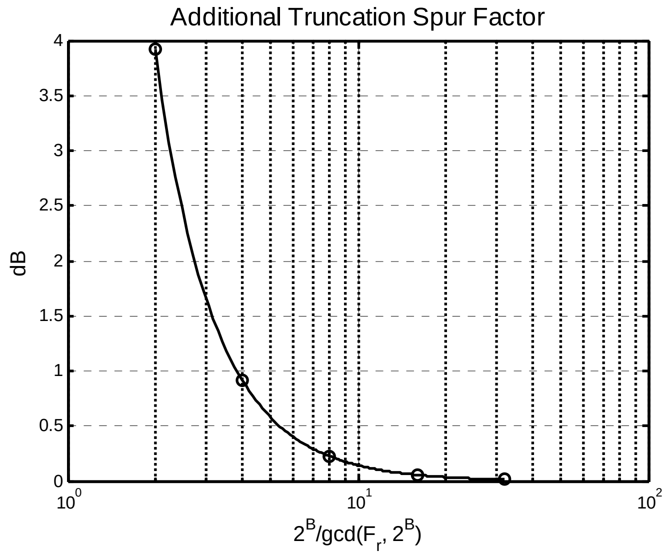


**Figure 5-16** The additional spur factor corresponding to the $20\log_{10}(\ )$ value in (5.41). For $B > 4$, a value much greater than a fraction of one dB is rare.

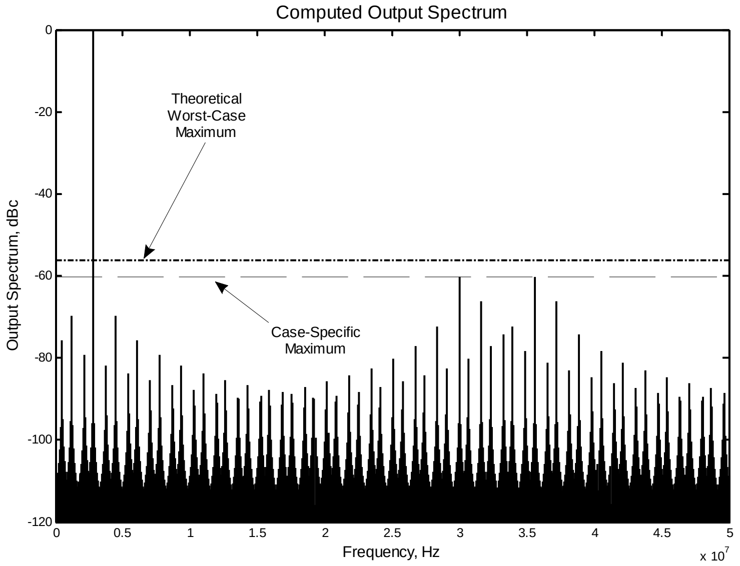


**Figure 5-17** Sinusoidal DDS output spectrum[15] ( not including the DAC $\sin(x)/x$ ) for $F_r$ = 7340, 18-bit phase accumulator, $W = 10$, floating-point precision for the $\sin(\theta)$ values, $f_{Clock}$ = 100 MHz. The theoretical worst-case maximum spur level is computed using (5.40) whereas the case-specific maximum spur level is computed using (5.41).

---

[15] MATLAB script u16484_nco_phase_truncation.m.

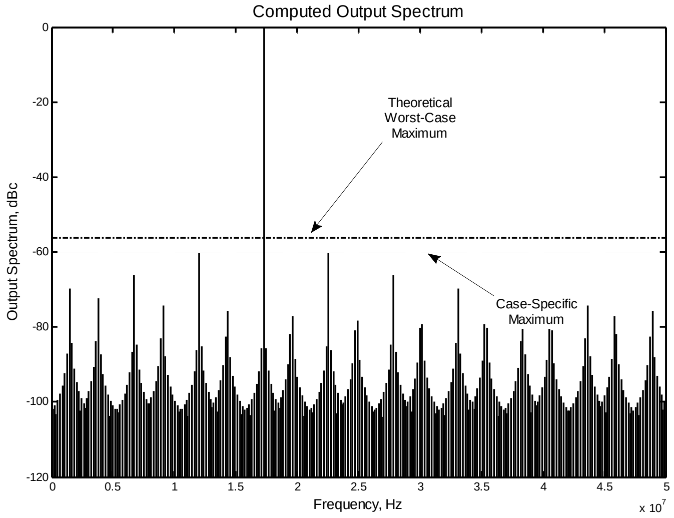


**Figure 5-18** Case identical to Figure 5-17 except $F_r = 45298$

### 5.5.2.1 Phase Truncation Spectrum Details

The complete spectrum associated with phase truncation was first solved by Nicholas and Samueli [9, 11], but has since been revisited by numerous researchers offering additional insights and advancements [12, 13, 14, 15]. The solution given by Nicholas and Samueli is provided below.

- Divide the frequency range from 0 to $f_{Clock} / 2$ into a sequence of $2^{N-1} / ( F_r, 2^N )$ evenly spaced potential spur locations, each sequentially numbered. The sequential frequency number of a spur location is related to the actual analog spur frequency, $f_{Spur}$, by

$$F_n = \frac{f_{Spur}}{f_{Clock}} \frac{2^N}{\left(F_r, 2^N\right)} \tag{5.42}$$

  In this context, $N$ is the number of bits in the phase accumulator and ( $a$, $b$ ) denotes the greatest common divisor of $a$ and $b$.

- Find the spur number, $K$ , from the frequency number, $F_n$, using the following rules:
  a) If both $F_n - \Lambda$ and $-F_n - \Lambda$ are not divisible by 2, then the magnitude of the spur at $F_n$ is zero.
  b) If 2 divides $F_n - \Lambda$, then

$$K = \left\langle \Gamma^{\Lambda-1} \frac{F_n - \Gamma}{2^{N-B}} \right\rangle_{2\Lambda} \tag{5.43}$$

  where

$$\Lambda = \frac{2^{B-1}}{\left(F_r, 2^B\right)} \tag{5.44}$$

and $B$ is the number of phase-bits truncated.

c) Otherwise, 2 must divide $-F_n - \Lambda$ and

$$K = \left\langle -\Gamma^{\Lambda-1} \frac{F_n + \Gamma}{2^{N-B}} \right\rangle_{2\Lambda} \tag{5.45}$$

In these last equations, the large brackets $< a >_b$ represents taking the integer remainder of $a$ modulo $b$ ( e.g., $< 17 >_4 = 1$ ).

- The magnitude of the spurious noise line at $F_n$ is given by

$$\varsigma_K = \frac{\pi 2^{B-N}}{2\Lambda} \operatorname{cosec}\left( \frac{\pi K}{2\Lambda} \right) \tag{5.46}$$

As noted in [9], the number of spurs and their magnitudes only depend on $F_r$ through the greatest common divisor quantity ( $F_r$, $2^B$ ). Values of $F_r$ having the same value for ( $F_r$, $2^B$ ) also have the same number of spurs and same spur amplitudes. Perhaps most interesting of all, the spur frequency locations are simple rearrangements of one another.

### 5.5.2.2 Spurious Due to Phase Quantization Excluding Phase Truncation

Finite precision within a traditional DDS always leads to additional periodicity factors resulting in unwanted spurious products being created. The modulo-P number theoretic perspective developed here makes it possible to fully characterize the spurious performance of an otherwise ideal DDS with very minimal analysis compared to previous methods. These same computational techniques can be extended to address several DAC-related imperfections as well.

The preceding results, while computationally valuable, fail to convey much insight into what is happening under the banner of phase truncation. This can be remedied by building upon the statements in the previous paragraph in the context of [13] which is an overview of the larger work in [16].

In the sinusoidal DDS case represented by (5.38), the phase argument sequence is given by

$$\theta(n) = 2\pi 2^B \left\lfloor \frac{F_r n}{2^{N-B}} \right\rfloor \tag{5.47}$$

For the moment, assume there is no phase truncation taking place ( $B = 0$ and $W = N$ ) and consider two frequency control words $F_{r1} = 0001101$ and $F_{r2} = 0000001$. It is important to note that both frequency words have ones in their LSB bit position. In this present context, both words have $N = 7$ and a DDS output period $P = 2^N = 128$ samples.[16] The contents of the numerical phase accumulator in Figure 5-15 for these two different frequency cases are given by

$$\begin{aligned} s_1(n) &= \langle nF_{r1} \rangle_P \\ s_2(n) &= \langle nF_{r2} \rangle_P \end{aligned} \tag{5.48}$$

where $\langle\ \rangle_P$ denotes modulo base $P$. By selecting $F_{r2}$ equal to the minimum nonzero frequency word possible, every possible phase accumulator state is visited in sequential order before the entire sequence is repeated. The two sample sequences are plotted in Figure 5-19 and appear to be quite different, along with their respective DFT spectrums in Figure 5-20. It turns out, however, that the two sequences can be expressed in terms of each other by a simple index rearrangement as

$$\begin{aligned} s_1(n) &= s_2\left(\langle n\,F_{r1} \rangle_P\right) \\ s_2(n) &= s_1\left(\langle n\,J \rangle_P\right) \end{aligned} \tag{5.49}$$

where $J$ is the smallest integer satisfying the equation

$$\langle F_{r1}\,J \rangle_P = 1 \tag{5.50}$$

Two relationships used later in proving these results for integer values $a$, $b$, and $c$ are

$$\langle a\,b \rangle_c = \langle \langle a \rangle_c \langle b \rangle_c \rangle_c \tag{5.51}$$

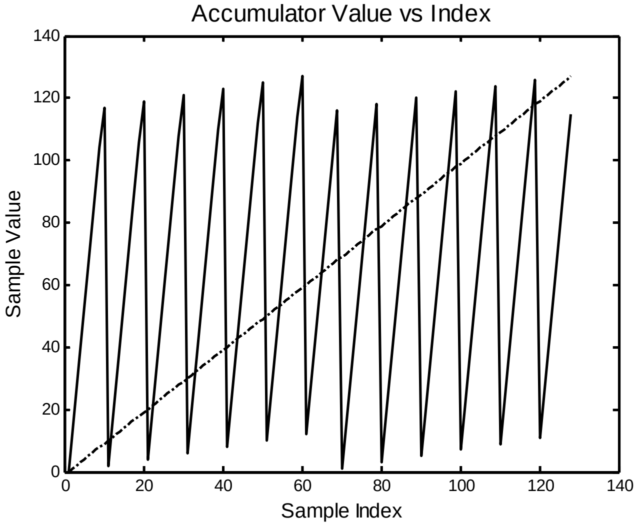


**Figure 5-19** Accumulator value versus sample index[17] for $F_{r1} = 13$ and $F_{r2} = 1$ ( $N = 7$ )

---

[16] The smallest nonzero LSB determines the repetition period of the DDS output.

[17] MATLAB script u16537_torosyan_dds.m.

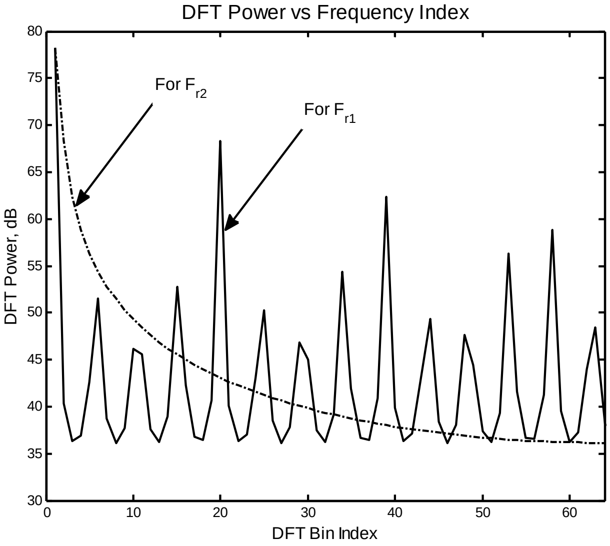


**Figure 5-20** DFT of the sample sequences[18] $s_1$ and $s_2$ shown in Figure 5-19 and given by (5.48)

$$\left\langle 2^d a \right\rangle_{2^N} = 2^d \left\langle a \right\rangle_{2^{N-d}} \tag{5.52}$$

Several of the $J$-values for the $N = 7$ case are listed in Table 5-2. A sample index reassignment example for the case $F_{r1} = 3$ is provided in Table 5-3.

**Table 5-2** J-Values Satisfying (5.50) for $P = 2^N = 128$

| $F_{r1}$ | $J$ |
|---|---|
| 3 | 43 |
| 5 | 77 |
| 7 | 55 |
| 9 | 57 |
| 11 | 35 |
| 13 | 69 |
| 15 | 111 |
| 17 | 113 |
| 19 | 27 |

**Table 5-3** Example Sample Index Rearrangement for $F_{r1} = 3$ ( $J = 43$, $N = 7$ )

| $n \rightarrow$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $\langle n J \rangle_P \rightarrow$ | 43 | 86 | 1 | 44 | 87 | 2 | 45 | 88 |

---

[18] Ibid. Only odd values of $F_{r1}$ are used in order to keep the LSB a one.

The rearrangement relationship given by (5.49) has been recognized by many researchers going back to the original works in [9] and [11], but the proof of this relationship has typically been rather involved. The explanation that follows is based upon basic number theory.

Let the two sample sequences in (5.48) be denoted by

$$S_1(n) = \left\{ \langle MA \rangle_P, \langle 2MA \rangle_P, \langle 3MA \rangle_P, \ldots, \langle RMA \rangle_P \right\} \tag{5.53}$$

$$S_2(n) = \left\{ \langle A \rangle_P, \langle 2A \rangle_P, \ldots, \langle RA \rangle_P \right\} \tag{5.54}$$

where $M = F_{r1} / F_{r2}$, $A = 2^u = F_{r2}$, $P = 2^N$, and $R = 2^{(N-u)}$. The frequency control word situation is shown more clearly in Figure 5-21. In this context, $M$ must always be an integer.

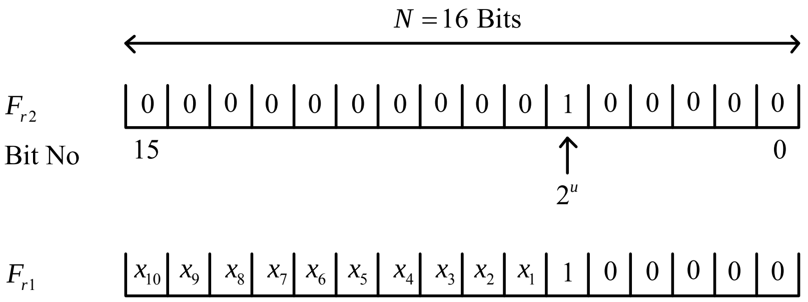


**Figure 5-21** Frequency control word for the more general case. $F_{r2}$ always has a single binary-one in the $u^{th}$ bit position as does $F_{r2}$, but at least one of the $x_k$ values in $F_{r1}$ must also be a binary-one. Each $F_{r2}$ case ( one for each value of $u$ ) serves as the basis case for all of the respective $F_{r2}$ values having the same right-most nonzero LSB set.

Focusing first on (5.54), the value of the nth term in the $s_2$ sequence is given by

$$\begin{aligned} s_2(n) &= \langle nA \rangle_P \\ &= \left\langle n2^u \right\rangle_{2^N} \\ &= 2^u \langle n \rangle_{2^{(N-u)}} \\ &= 2^u \langle n \rangle_T \end{aligned} \tag{5.55}$$

where $T = 2^{(N-u)}$. In the case of (5.53), the nth term of the $s_1$ sequence is given by

$$\begin{aligned} s_1(n) &= \langle n\,MA \rangle_P \\ &= \left\langle n\,M\,2^u \right\rangle_P \\ &= 2^u \langle n\,F_{r1} \rangle_T \end{aligned} \tag{5.56}$$

Using these results, transitive relationships can be written as

$$\begin{aligned} s_1(n) &= 2^u \left\langle nF_{r1} \right\rangle_T \\ s_2\left( \left\langle nF_{r1} \right\rangle_T \right) &= 2^u \left\langle nF_{r1} \right\rangle_T \end{aligned} \tag{5.57}$$

thereby making it possible to conclude that

$$s_1(n) = s_2\left( \left\langle n\,F_{r1} \right\rangle_T \right) \tag{5.58}$$

Similarly, it is true that

$$s_2(n) = 2^u \left\langle n \right\rangle_T \tag{5.59}$$

from (5.55), and

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

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

$$s_2(n) = s_1\left( \left\langle n\,J \right\rangle_T \right) \tag{5.61}$$

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 $u \in [0, \ldots, N-1]$ ) 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

$$\begin{aligned} S_1(k) &= S_2\left(\langle k\,J\rangle_T\right) \\ S_2(k) &= S_1\left(\langle k\,F_{r1}\rangle_T\right) \end{aligned} \tag{5.62}$$

This means that the same rearrangements used in (5.49) for the time sequences apply to the frequency domain as well. The proof for this statement is based upon the symmetry inherent within the DFT. The DFT for each time sequence can be written as

$$\begin{aligned} S_1 &= \boldsymbol{W} s_1 \\ S_2 &= \boldsymbol{W} s_2 \end{aligned} \tag{5.63}$$

where $\boldsymbol{W}(k, n) = \exp(-j\,2\pi\,k\,n\,/\,T)$. If the $n^{\text{th}}$ column of $\boldsymbol{W}$ is replaced with the $< n\,F_{r1} >_T$ column to create a new matrix ***H,*** each element is given by

$$\begin{aligned} \boldsymbol{H}(k,n) &= \boldsymbol{W}\left(k,\langle n\,F_{r1}\rangle_T\right) \\ &= \exp\left(-j\frac{2\pi}{T}k\langle n\,F_{r1}\rangle_T\right) \end{aligned} \tag{5.64}$$

Since the integer portion of the $\langle\ \rangle_T$ quantity in (5.64) contributes nothing to the final result, (5.64) can be rewritten as

$$\begin{aligned} \boldsymbol{H}(k,n) &= \exp\left(-j\frac{2\pi}{T}\left\langle k\langle n\,F_{r1}\rangle_T\right\rangle_T\right) \\ &= \exp\left(-j\frac{2\pi}{T}\langle k\,n\,F_{r1}\rangle_T\right) \\ &= \exp\left(-j\frac{2\pi}{T}n\langle k\,F_{r1}\rangle_T\right) \end{aligned} \tag{5.65}$$

which is the same value for the $(k, n)$ element that would have been obtained had a row rearrangement of $\boldsymbol{W}$ been done instead. Similarly, replacing the $n^{\text{th}}$ column of $\boldsymbol{H}$ with the $< n\,J >_T$ column or replacing the $k^{\text{th}}$ row of $\boldsymbol{H}$ with the $< k\,J >_T$ row reproduces the original DFT matrix $\boldsymbol{W}$.

Based upon this result and starting with the lower equation in (5.63), rearrangement of the sequence elements in $s_2$ by $F_{r1}$ along with a column rearrangement of $\boldsymbol{W}$ by $F_{r1}$ ( thereby creating $\boldsymbol{H}$ ) leaves the left-hand side of the equation unchanged as

$$\begin{aligned} S_2 &= \boldsymbol{W} s_2 \\ &= \boldsymbol{H} s_2\left(\langle k\,F_{r1}\rangle_T\right) \\ &= \boldsymbol{H} s_1 \end{aligned} \tag{5.66}$$

Similarly starting with the upper equation in (5.63), rearrangement of the sequence elements in $s_1$ by $J$ along with a column rearrangement of $\boldsymbol{W}$ by $J$ ( thereby creating $\boldsymbol{H}$ ) leaves the left-hand side of the equation unchanged as

$$\begin{aligned} S_1 &= \boldsymbol{W} s_1 \\ &= \boldsymbol{H} s_1\left(\left\langle k\,J\right\rangle_T\right) \\ &= \boldsymbol{H} s_2 \end{aligned} \tag{5.67}$$

In conclusion, the set of spur magnitudes created by $F_{r1}$ is identical to the set created by $F_{r2}$ except for the spurs rearrangement in the frequency domain. It is therefore possible to predict the frequency domain behavior for every possible $F_{r1}$ value using only the $N$ spectrum results for the $F_{r2}$ basis values along with the appropriate frequency bin rearrangements.

### 5.5.2.4 Including Phase Truncation

The preceding results for the non-truncation case can also be applied directly when phase truncation is present. When phase truncation is present, the $B = N - W$ LSBs in Figure 5-21 are not used to compute the sin( $\theta$ ) value as shown in Figure 5-15. For the $u = 0$ frequency control word case, $F_{r2}$ results in a ramp function that has each step repeated $2^B$ times as shown in Figure 5-22. The analysis is simplified if the LSB of $F_{r1}$ is always unity because only one basis value for $F_{r2}$ is needed ( $u = 0$ ) to compute the DDS behavior for all possible cases of $F_{r1}$. This assumption is made throughout the following discussion.

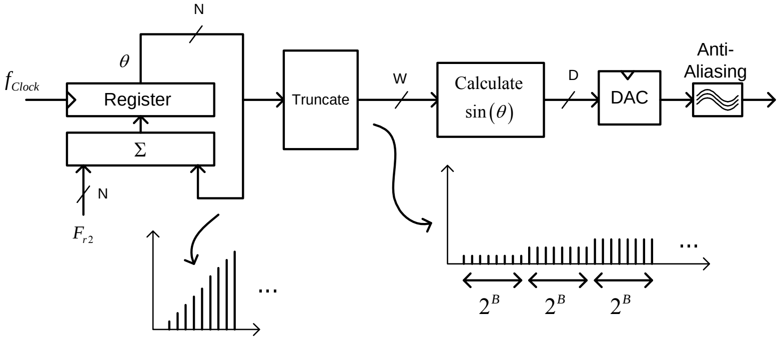


**Figure 5-22** Phase truncation causes the phase argument presented to the sin( $\theta$ ) calculation to be truncated in value and repeated for $2^B$ steps in a row for the $F_{r2}$ case with $u = 0$

The DFT of the bit-truncated $s_2$ sample sequence can be written for the nontrivial case ( $q > 1$ ) as

$$S_2'(k) = \left(1 + e^{-j2\pi k/P} + \ldots + e^{-j2\pi k(q-1)/P}\right)\sum_{n=0}^{T-1} s_2(qn)\, e^{-j2\pi nk/T} \tag{5.68}$$

where $q = 2^B$, $P = 2^N$, and $T = 2^W$ for $k = 0, 1, \ldots, 2^N - 1$. The prime denotes that this is the DFT for the bit-quantized $s_2$ sequence. This result may be further simplified to

$$S_2'(k) = \exp\left[-j\frac{\pi k(q-1)}{P}\right] \times \left[\frac{\sin\left(\pi\dfrac{kq}{P}\right)}{\sin\left(\dfrac{\pi k}{P}\right)}\right]\sum_{n=0}^{T} s_2(qn)\, e^{-j2\pi nk/T} \tag{5.69}$$

The DFT for all of the bit-truncated $s_1$ sequences directly follows by using the frequency bin rearrangement which was described earlier in connection with (5.67) and applying it to (5.69).

### 5.5.2.5 Including the Ideal Mapping $\theta$ to sin( $\theta$ )

Thus far, all of the attention has been given to the behavior of the phase argument presented to the $\theta$-to-sin( $\theta$ ) computation shown in Figure 5-22. In actuality, most of the attention has been given to the indices used with the time and frequency domain sequences rather than the actual phase values. Since the $\theta$ to sin( $\theta$ ) mapping is a one-to-one mapping, it is a simple matter to include this functionality so that the DFT of the basic $s_2$ sequence at the output of the mapping function is given by

$$S_2''(k) = \exp\left[-j\frac{\pi k(q-1)}{P}\right] \times \left[\frac{\sin\left(\pi\frac{kq}{P}\right)}{\sin\left(\frac{\pi k}{P}\right)}\right] \sum_{n=0}^{T} g\left(2\pi\frac{qn}{T}\right) e^{-j2\pi nk/T} \qquad (5.70)$$

In the ideal case, $g(\theta) = \sin(\theta)$, but this formulation makes it easy to include nonlinear DAC distortions of any kind including quantization so long as $g$ is a straight-forward mapping. Nonlinear DAC behavior is discussed separately in Section 5.7.

### 5.5.2.6 Phase Truncation Examples

**Example I: $N$ = 8, $W$ = 3, $F_{r1}$ = 3**

In this example, $W$ = 3 and five bits of the phase argument are truncated. The small frequency control word has been purposely chosen in order to make the different sequences easier to identify. The phase accumulator output and truncated phase accumulator output versus time index are shown in Figure 5-23 where the expected stair-step behavior due to truncation is clearly apparent.

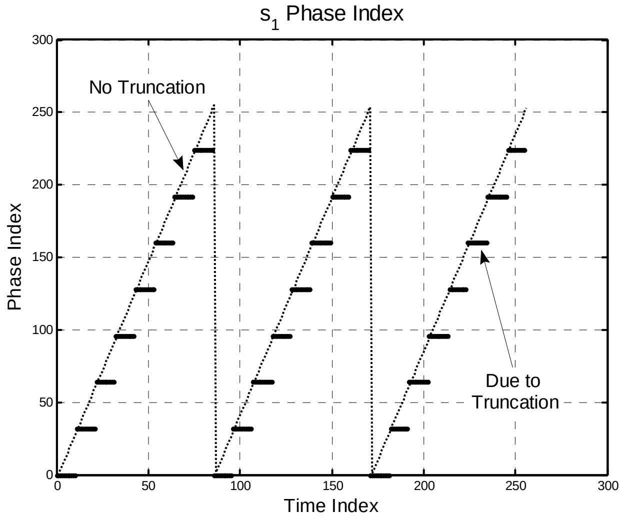


**Figure 5-23** Phase argument index versus time index[19] for $N$ = 8, $B$ = 5, $F_{r1}$ = 3. Stair-step behavior due to phase truncation is clearly apparent.

---

[19] MATLAB script u16538_torosyan_spurs.m.

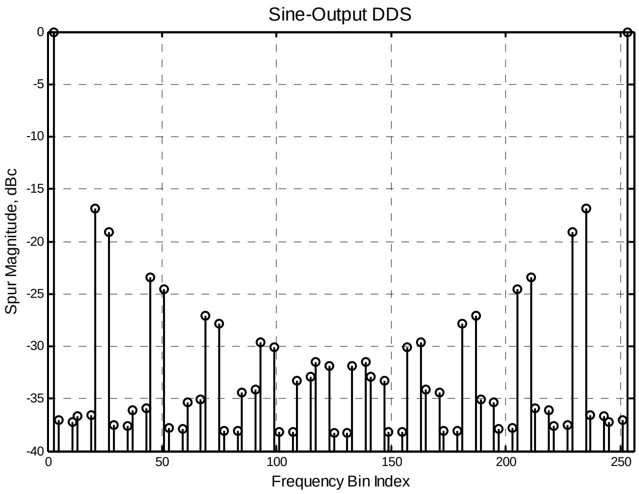


**Figure 5-24** DDS output spectrum after the phase $\theta$ has been perfectly converted to sin($\theta$). The sine-weighting strongly suppresses many of the spurs that would otherwise be present. The desired signal is present in bin #3 with its alias in bin #253. The worst-case undesired spur level 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 $g(\ )$ 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.

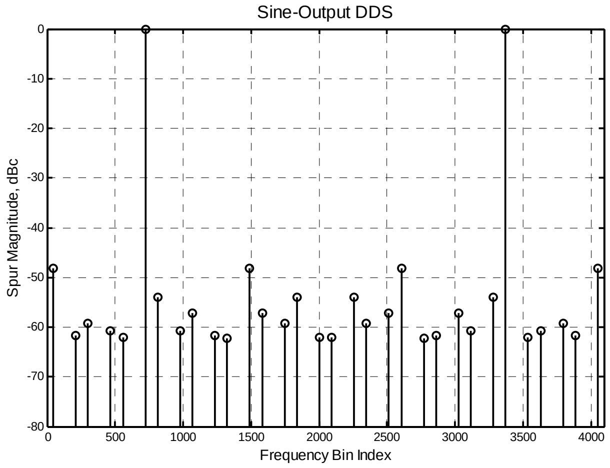


**Figure 5-25** DDS output spectrum showing the worst-case spur level at about –48 dBc which is again consistent with the bound of –44.1 dBc predicted by (5.40). $N = 12$, $B = 4$, $F_{r1} = 723$.

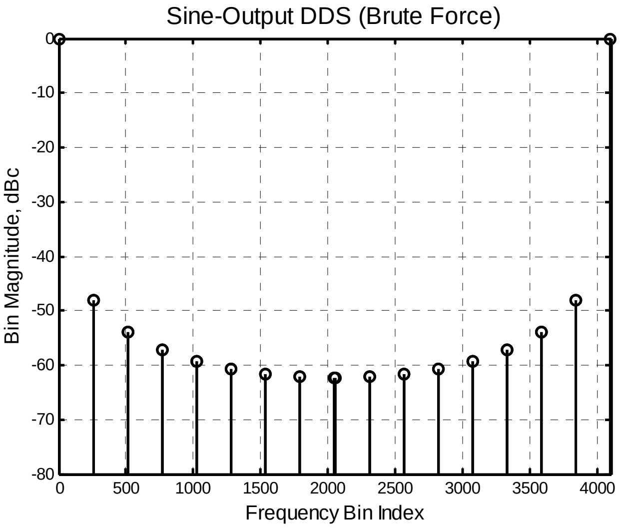


**Figure 5-26** Phase truncation related spurs associated with Example II for the basis $F_{r2}$ ( = 000000000001 ) case. $N = 12$, $B = 4$, $F_{r1} = 1$.

### 5.5.2.7 Phase Truncation Summary

The spurious performance related to phase truncation has been summarized in terms of a simple index-mapping method applied to a single basis spectrum corresponding to $F_{r1}$ = 000…0001 in this chapter. Only one basis spectrum is needed to compute the spectrum for all other DDS frequency word values so long as the LSB of every frequency word is always unity.

A good starting point for any DDS design is to simply compute the phase-related spur levels associated with the $F_{r2}$ frequency word case. The frequency locations of these spurs move around a lot with other frequency word choices, but the spur levels do not. It is a simple matter to compute how different spurs move in the DDS output frequency spectrum with different frequency words applied given the index-mapping relationships presented in this chapter.

### 5.5.3 DAC Noise and Spurious Performance Related to DAC Quantization

The DAC shown in Figure 5-15 is limited to $D$-bits of resolution and this limitation leads to DDS output noise and spurious tones being created. Unlike the phase truncation spurs just discussed that always lead to fairly distinct spurious tones, however, DAC quantization usually creates more random spurious tones in comparison.

The quantization noise that arises with even an ideal $D$-bit DAC is not necessarily uncorrelated as discussed in Section 8.3.3 of [6]. If the quantization errors can be viewed as a wide-sense stationary random process represented by $v( n )$ having an autocorrelation function given by $R_v( k )$, however, it has a corresponding power spectral density given by [6]

$$S_v\left( f \right) = 2T_{Clock}\left[ \frac{\sin\left( \pi f T_{Clock} \right)}{\pi f T_{Clock}} \right]^2 \left\{ \frac{R_v\left(0\right)}{2} + \sum_{k=1}^{\infty} R_v\left(k\right)\cos\left(2\pi f k T_{Clock}\right) \right\} \qquad (5.71)$$

when the time samples are uniformly $T_{Clock}$ seconds apart. Large values of $| R_v ( k ) |$ automatically imply the presence of strong spurious tones in the output spectrum. An obvious question to consider is whether the occurrence of these spurs can be predicted in a manner similar to that obtained for the phase truncation spurs.

Based upon the material presented earlier in Section 5.5.2.5, DAC quantization still results in a one-to-one mapping between $\theta$ and $g(\theta)$, and the resultant spurs consequently obey the same index-rearrangement rules developed earlier in Sections 5.5.2.3 and 5.5.2.4. DAC quantization related spurs behave quite disjointly from their phase-related spurious counterpart making their independent assessment relatively straight forward. The following example illustrates this behavior.

Consider a DDS case using a 12-bit phase accumulator ( $N = 12$ ), truncated $\theta$ to sin( $\theta$ ) lookup table using 8-bits ( $W = 8$ ), and an output DAC having 8 or 16-bit precision ( $D$ ). The resultant spur levels are shown in Figure 5-27 for the 16-bit DAC case. When the DAC precision is reduced to 8-bits, many more discrete spurs appear as shown in Figure 5-28 but the phase truncation spur levels and positions remain unchanged.

Since the same index mapping relationships developed earlier in (5.49) and (5.62) still apply with DAC-related quantization noise, the worst-case spur level for all frequency control words having the same minimum LSB position set can be found by considering only one control word as done in Section 5.5.2.2 for the phase truncation spurs. To pursue this statement further, the worst-case spur levels can be computed for all *W*-bit phase truncation cases that have *D*-bit sin( $\theta$ ) quantization (corresponding to a DAC with *D*-bits of precision) as shown in Figure 5-22. In each case, the frequency control word has the LSB set to unity. The results are shown in Figure 5-29 with *W* as the independent variable and in Figure 5-30 where *D* is the independent variable. The horizontal lines break sharply from the sloped curve because the phase truncation and DAC quantization spurs are essentially independent of one another in amplitude and position in frequency as already mentioned.

The worst-case spur level due to DAC quantization alone can be found by considering the endpoints with $W = 17$ in Figure 5-29. A linear regression of these points shows that the DAC-related worst-case spur level is given approximately by

$$L_{wcqr} = -7.63D - 6.65 \quad \text{dBc} \tag{5.72}$$

All of the preceding results apply when rounding is used immediately ahead of the DAC. If the DAC values are simply truncated to *D*-bits rather than rounded, the output spurious performance can be slightly better or worse than the rounded case depending upon the specific *W* and *D* value used. The linear regression equivalent to (5.72) for the DAC truncated-input case is

$$L_{wcqt} = -7.83D - 4.54 \quad \text{dBc} \tag{5.73}$$

and the associated worst-case spurious results are shown in Figure 5-31 and Figure 5-32.

It is important to remember that none of these results include the sin( $x$ ) / $x$ frequency-domain shaping that occurs due to the zero-order sample-and-hold inherent within the DAC. This function should always be included when comparing spurious levels to the level of the desired sine wave component.

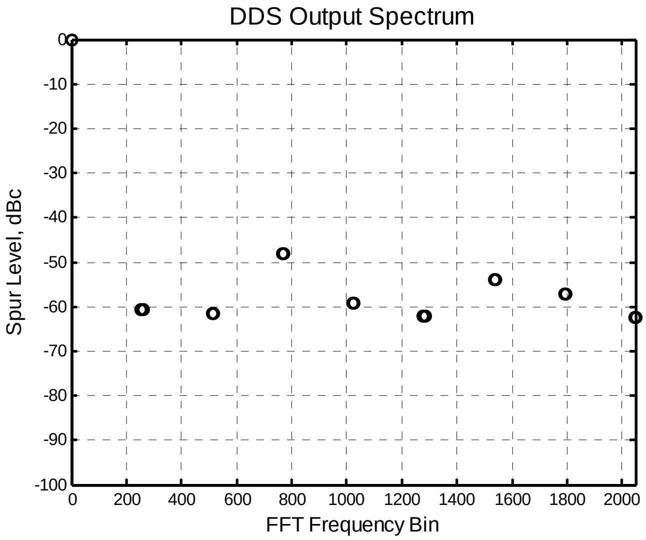


**Figure 5-27** DDS output spectrum for $N = 12$, $W = 8$, and $D = 16$.[20] The only spurs visible are phase truncation related spurs and the worst-case spur level is about –48 dBc as predicted by (5.40).

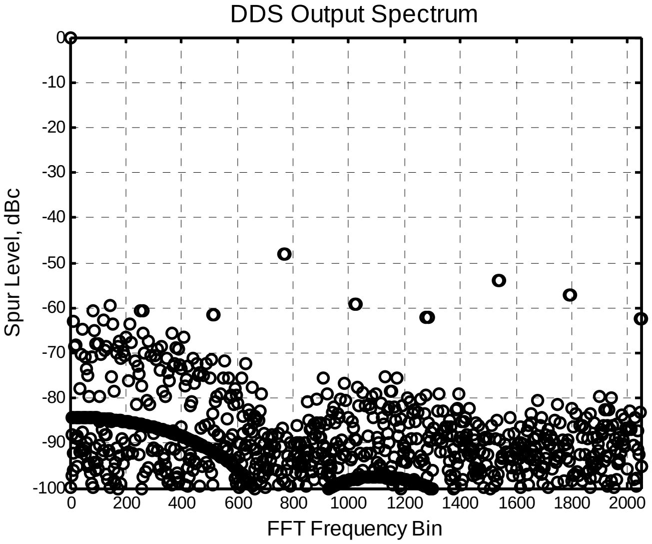


**Figure 5-28** DDS output spectrum for $N = 12$, $W = 8$, and $D = 6$.[21] Many more discrete spurs have appeared due to the smaller number of DAC bits used, but the magnitude and position of the phase truncation related spurs remain unchanged compared to Figure 5-27.

---

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

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

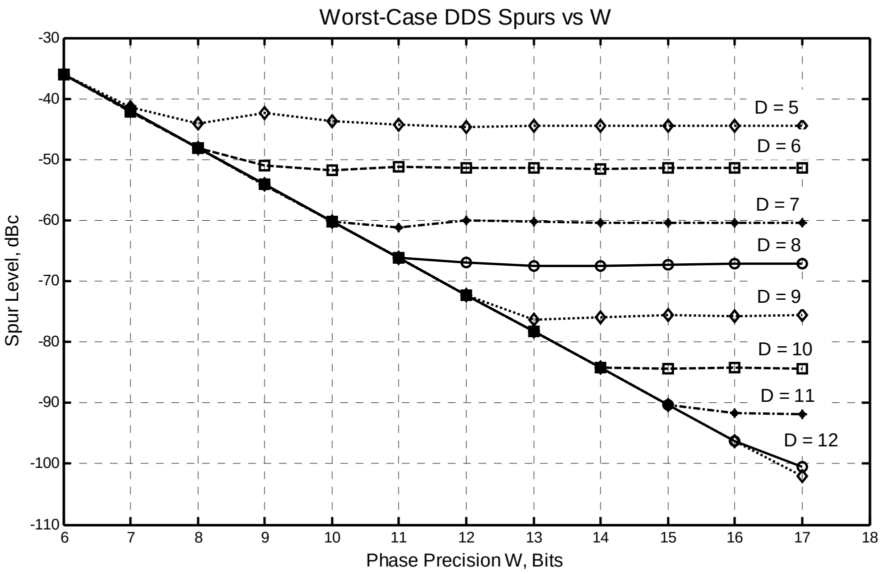


**Figure 5-29** Worst-case DDS spur level versus number of phase bits *W* and number of DAC bits *D*. [22] (DAC input values rounded to *D*-bits.)

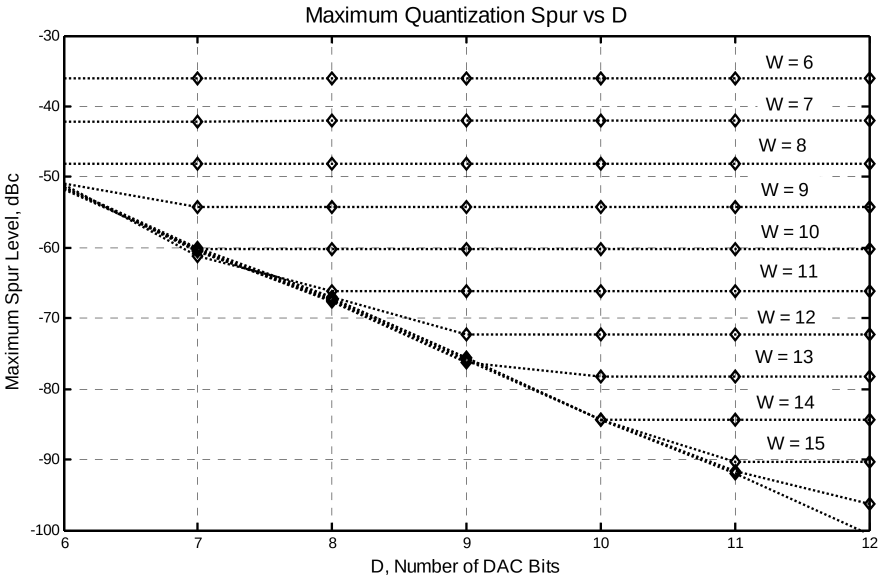


**Figure 5-30** Worst-case DDS spur level versus number of phase bits *W* and number of DAC bits *D* (as in Figure 5-29) but with *D* as the independent variable. [23] (DAC input values rounded to *D*-bits.)

---

[22] From u16539_dac_quantization_errors.m.

[23] From u16539_dac_quantization_errors.m.

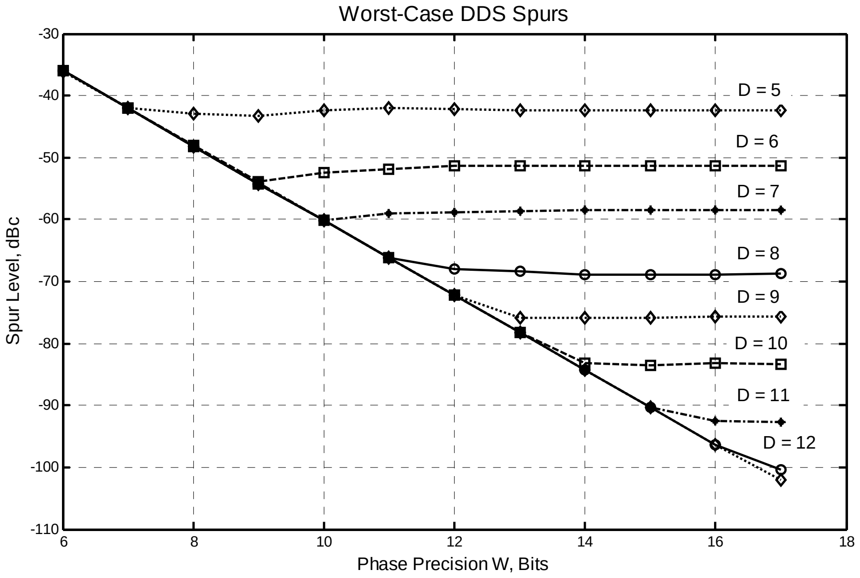


**Figure 5-31** Worst-case DDS spur level versus number of phase bits *W* and number of DAC bits *D*.[24] (DAC input values truncated to *D*-bits.)

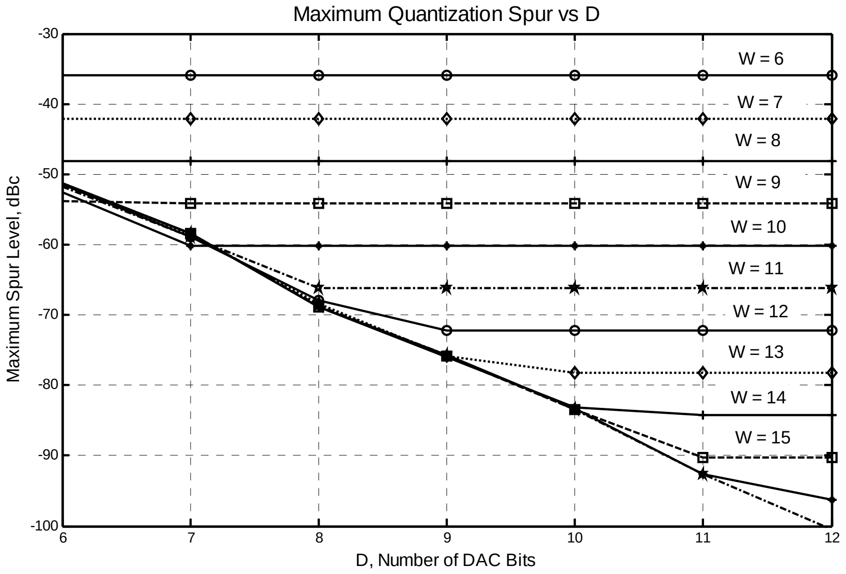


**Figure 5-32** Worst-case DDS spur level versus number of phase bits *W* and number of DAC bits *D* (as in Figure 5-31) but with *D* as the independent variable.[25] (DAC input values truncated to *D*-bits.)

---

[24] From u16539_dac_quantization_errors.m.

[25] From u16539_dac_quantization_errors.m.

### 5.5.4 Output SNR Due to DAC Quantization

A full-scale sine wave from an ideal D-bit DAC has numerical values that span from 0 to $2^D - 1$. This corresponds to a sine wave peak value of approximately $\pm\, 2^{D-1}$ and a corresponding power level (voltage squared) of $P_{Sig} = 2^{2D-3}$. Assuming that the quantization errors are uncorrelated and uniformly distributed between –1/2 and +1/2, equation (5.71) can be used to show that the (two-sided ) power spectral density of the quantization noise is given by

$$\begin{aligned} N_q &= T_{Clock} \left[ \frac{\sin\left(\pi f\, T_{Clock}\right)}{\pi f\, T_{Clock}} \right]^2 \sigma^2 \\ &\approx \frac{T_{clock}}{12} \; \frac{V^2}{Hz} \qquad \text{for } \left| f \right| \le \frac{F_{Clock}}{2} \end{aligned} \tag{5.74}$$

where the variance is $\sigma^2 = 1/12$. The signal to noise power spectral density ratio is then given by

$$\frac{P_{Sig}}{N_q} = \frac{12}{T_{Clock}} \times 2^{2D-3} = \frac{3}{2} 2^{2D} \frac{1}{T_{Clock}} \tag{5.75}$$

and this can be written in decibel form as

$$\begin{aligned} SNR &= 10\log_{10}\left( \frac{2^{2D}}{T_{Clock}} \frac{3}{2} \right) \\ &\cong 6.02D + 1.76 + 10\log_{10}\left( F_{Clock} \right) \text{ dB-Hz} \end{aligned} \tag{5.76}$$

Increasing the clock frequency by a factor of 10× only improves the SNR by 10 dB whereas increasing the DAC resolution by a single bit increases it by 6 dB. Once $F_{Clock}$ is greater than several hundred MHz, the SNR is more easily improved by adding additional bits of resolution than by increasing the clock frequency.

## 5.6 Techniques for Mapping $\theta$ to Sin($\theta$)

A number of different methods can be used to map the phase accumulator value $\theta$ to the desired sinusoidal output value given by sin($\theta$) as outlined in Table 5-4. Several of these methods are quantitatively compared in [17].

The table look-up method is widely used for its simplicity. For better spurious performance, however, a straight table look-up approach is not space efficient and other techniques like the trigonometric method by Sunderland should be adopted. In this method, the phase accumulator value represented by $\theta$ is first reflected to a value that falls within the first quadrant as $\phi = \theta \bmod ( \pi / 2 )$. This resultant angle is then broken into a sum of coarse and fine angles as $\phi = ( \alpha + \beta ) + \gamma$ with $\alpha < \pi / 2$ and

$$\begin{aligned} \beta &< \frac{\pi}{2} 2^{-A} \\ \gamma &< \frac{\pi}{2} 2^{-(A+B)} \end{aligned} \tag{5.77}$$

Simple trigonometry and the use of the small angle approximations lead to

$$\begin{aligned} \sin(\alpha+\beta+\gamma) &= \sin(\alpha+\beta)\cos(\gamma)+\cos(\alpha+\beta)\sin(\gamma) \\ &= \sin(\alpha+\beta)\cos(\gamma)+ \\ &\quad \left[\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta)\right]\sin(\gamma) \\ &\cong \sin(\alpha+\beta)+\cos(\alpha)\sin(\gamma) \end{aligned} \tag{5.78}$$

The two quantities stored in table lookups are sin( $\alpha + \beta$ ) and sin( $\gamma$ ). Additional look-up table storage compression can be obtained by making use of the first-order Taylor series expansion for sin( ) and storing the quantity

**Table 5-4** Summary of Sin( $\theta$ ) Mapping Techniques

| Method | Description | Pros & Cons |
|---|---|---|
| Table Look-Up | Truncated sine values are located in a read-only memory table using $\theta$ as an index [18]. | Simplicity.<br>Relatively poor spurious performance. |
| Combinatorial Logic | Makes use of the Taylor series for sin( $\theta$ ) and representing $\theta$ as an algebraic polynomial which is expanded and bit-terms collected [7]. | Small size.<br>Widely varying spurious performance versus frequency word. |
| Trigonometric (i.e., Sunderland) | Makes use of a coarse and fine table look-up while also exploiting small angle approximations [7]. | Good spurious performance vs complexity.<br>For very low spurious performance, other methods are more compact. |
| Polynomial | Taylor or Chebyshev series expansion used for sin( $\theta$ ). | Arbitrarily good spurious performance.<br>Computationally more complex. |
| CORDIC | Uses CORDIC algorithm to compute sin( ) and cos( ). | Arbitrary precision possible.<br>Can be pipelined for speed.<br>No ROM involved. |
| Table Look-Up Plus Interpolation | Coarse sin( ) values are saved in a table and intermediate values interpolated using Taylor series. | Arbitrarily good precision.<br>Excellent compromise between complexity and precision. |

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

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

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

**Table 5-5** CORDIC-computable functions

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

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

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

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

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

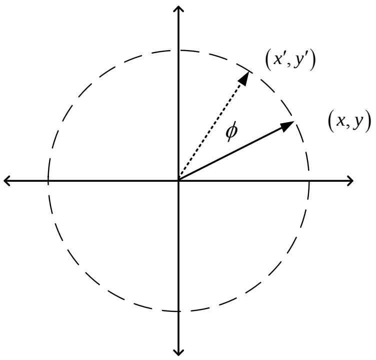


**Figure 5-33** Simple angular phase rotation of vector ( $x, y$ ) by angle $\phi$ to produce ( $x', y'$ )

Assume the total phase rotation to be performed on ( $x$, $y$ ) is given by $\theta$. When $\theta$ is represented by an N-bit binary quantity within the range of $-\pi / 2$ to $\pi / 2$, it can be represented by a sum of restricted elementary phase rotations $\pm\phi_k$ where $\tan(\phi_k) = 2^{-k}$ thereby making the multiplication by $\tan(\phi_k)$ in (5.82) a simple binary shift. In this context, the initial value $\theta$ is given by

$$\theta = \sum_{n=0}^{N-1} d_n \phi_n \tag{5.83}$$

The complete CORDIC algorithm for $| \theta | \le \pi / 2$ takes the form

```
for ii = 0 to N-1
    $\phi = \tan^{-1}\left(2^{-ii}\right)$
    $\Delta x = x\,2^{-ii}$
    $\Delta y = y\,2^{-ii}$
    if $(\theta > 0)$ $d = 1$ else $d = -1$
    $\theta = \theta - d\phi$                                    (5.84)
    $x = x - d\Delta y$
    $y = y + d\Delta x$
end
$x = x\,K_N^{-1}$
$y = y\,K_N^{-1}$
```

where ( $x$, $y$ ) must be initialized to ( 1, 0 ) and

$$K_N = \prod_{n=0}^{N-1} \sqrt{1 + 2^{-2n}} \Rightarrow 0.60725292 \text{ for } N \text{ large} \tag{5.85}$$

When implemented in hardware, all of the $\phi_k$ values should be saved in read-only-memory as well as the value for $1 / K_N$ for greatest computational economy.

The other functions shown in Table 5-5 can be computed using similar methods to (5.84) as discussed in [20] and [21]. The CORDIC method is used for direct digital frequency synthesis in the sections that follow.

### 5.6.1.1 CORDIC with First-Order Correction

Symmetry arguments can be used to reduce the computation range required from the CORDIC computation to $0 \leq \theta \leq \pi / 4$ as with most other methods. Since each CORDIC iteration reduces the worst-case phase error residual by a factor of two, it only takes six iterations to reduce this error down to less than 1° where small-angle approximations can be conveniently used. The first-order correction method described here performs a relatively few number of CORDIC iterations and then uses first-order approximation methods to complete the computation.

Assume that a minimum of six CORDIC iterations have been performed thereby reducing the residual rotational angle that must be handled $\phi$ to less than 1°. The sin( ) and cos( ) values computed up to this point are denoted here by $Q$ and $I$ respectively. Making use of the first-order approximation

$$e^{j\phi} \approx 1 + j\phi \tag{5.86}$$

the final cosine and sine values can be approximated as the real and imaginary parts of

$$e^{j\phi}\left(I + jQ\right) \approx I - \phi Q + j\left(Q + \phi I\right) = U + jV \tag{5.87}$$

Even though the modulus of $\exp( j\phi )$ and $I + jQ$ are both unity, the modulus of (5.87) is not precisely unity because of the approximations being used. The amplitude of (5.87) is in fact equal to

$$\left|U + jV\right| = \sqrt{1+\phi^2} \tag{5.88}$$

Unity-amplitude can be restored to (5.87) by multiplying by the reciprocal of (5.88) and this can be approximated by

$$g_1 = \frac{1}{\sqrt{1+\phi^2}} \approx 1 - \frac{\phi^2}{2} \tag{5.89}$$

An equivalent solution can be obtained by rewriting (5.88) as

$$\left|U + jV\right|^2 = 1 + \varepsilon \tag{5.90}$$

where $|\varepsilon| << 1$. Assuming that the gain-correction term must be applied equally to $U$ and to $V$, this gain correction can be written as

$$\begin{aligned} g_2 &= \frac{1}{\sqrt{1+\varepsilon}} \approx 1-\frac{\varepsilon}{2} \\ &\approx 1-\frac{1}{2}\left(U^2+V^2-1\right)=\frac{1}{2}\left[3-\left(U^2+V^2\right)\right] \end{aligned} \tag{5.91}$$

A block diagram of the complete first-order CORDIC correction method using (5.91) is shown in Figure 5-34. The spurious performance of this method is truly exceptional as shown in Figure 5-36. As shown in this figure, five CORDIC iterations are sufficient to reduce the worst-case spur level to less than –100 dBc when the residual phase rotation method described here is incorporated.

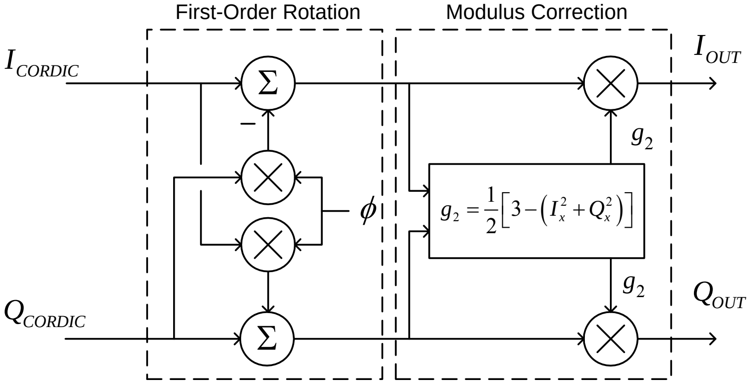


**Figure 5-34** First-order CORDIC correction method

### 5.6.1.2 CORDIC with Second-Order Correction

The so-called first-order correction method just described is really a second-order method because the corrective action is taken in two sequential steps, the first for phase and the second for amplitude. It is consequently worthwhile to reconsider the approximation made in (5.86) and look at the benefits obtained by using a second-order approximation to begin with. Taking this approach,

$$\begin{aligned} e^{j\phi} &= \cos(\phi)+j\sin(\phi) \\ &\approx 1-\frac{\phi^2}{2}+j\phi \end{aligned} \tag{5.92}$$

Using this approximation, the residual error phase rotation becomes

$$e^{j\phi}\left(I+jQ\right)\approx I\left(1-\frac{\phi^2}{2}\right)-\phi Q+j\left[Q\left(1-\frac{\phi^2}{2}\right)+\phi I\right]=U+jV \tag{5.93}$$

The modulus of this result is given by

$$\left|U+jV\right|=\sqrt{1+\frac{\phi^4}{4}} \tag{5.94}$$

The complete second-order CORDIC correction method is shown in Figure 5-35. The worst-case spurious performance for a CORDIC appended with this method is shown in Figure 5-37. For a given number of CORDIC stages (or iterations), this method is superior to the first-order method by 6 dB or more. Owing to its greater simplicity and better spurious performance, the second-order algorithm is recommended for use over the first-order method.

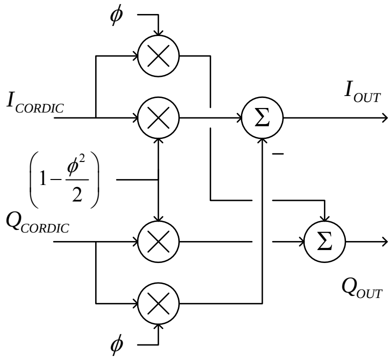


**Figure 5-35** Second-order CORDIC correction method

When the number of CORDIC iterations is only five or six, it is worthwhile replacing the CORDIC function with a simple table look-up function. The table contains the quantities $I_{CORDIC}$, $Q_{CORDIC}$, and $\theta_Q$ which is the quantized amount of phase represented by ($I_{CORDIC}$, $Q_{CORDIC}$). The $\theta_Q$ quantity makes it possible to compute the residual phase that still remains to be rotated that is represented by $\phi$ in Figure 5-35.

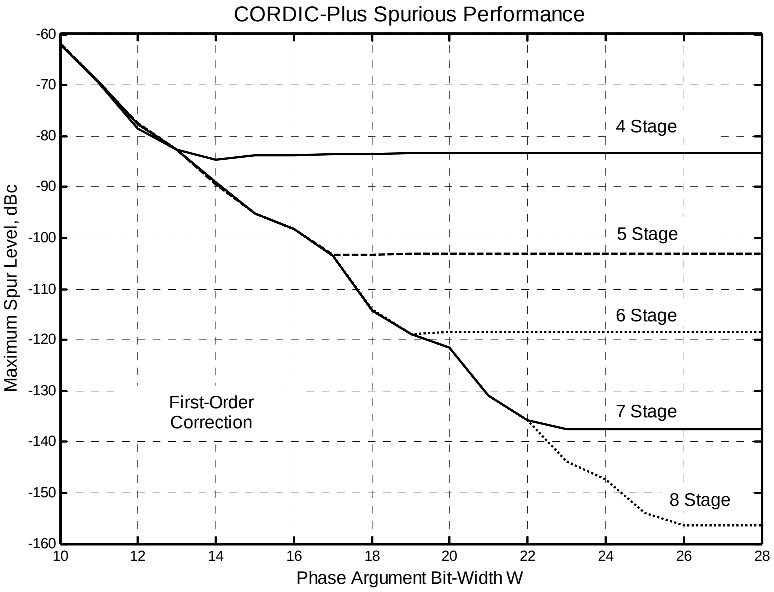


**Figure 5-36** Worst-case spur level for CORDIC using the first-order residual phase rotation method described in this section[26]

---

[26] MATLAB script u16455_cordic2.m with an input parameter of 3.

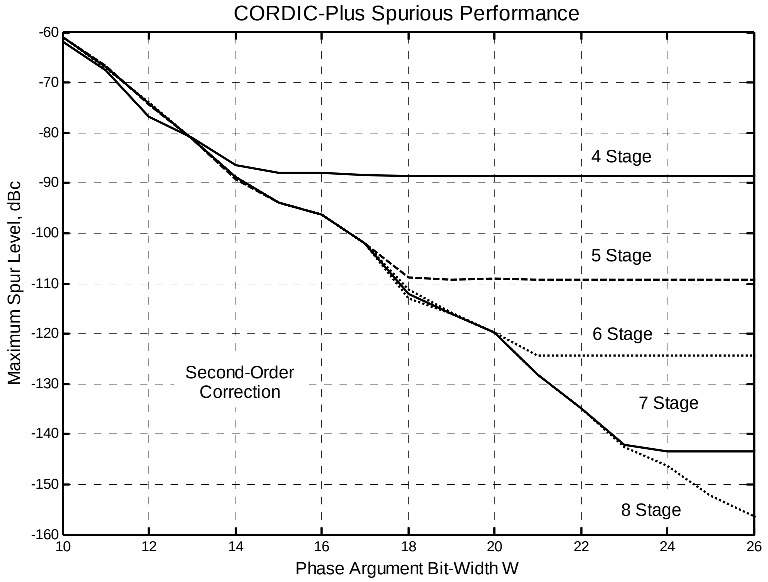


**Figure 5-37** Worst-case spur level for CORDIC using the second-order residual phase rotation method described in this section.[27] These results are markedly better than the first-order method described in the previous section for a given number of CORDIC iterations.

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.

*Integral Nonlinearity (INL)* – the normalized error between the actual converter output value and the best straight-line regression through all of the converter's codes, with respect to an ideal LSB based upon the straight-line regression.

*Signal-to-Noise Ratio (SNR)* – the ratio of desired signal power to noise power at the converter output for an applied full-scale sine wave, normally expressed in dB. For an ideal Nyquist-rate DAC, the SNR is given by $1.76 + 6.02N$ dB where $N$ is the number of bits in each digital codeword used.

*Signal-to-Noise-plus-Distortion (SNDR)* – the ratio of desired signal-power-to-noise-plus-distortion power including any dc offset that may be present, up to the Nyquist frequency for a full-scale sine wave output. Normally expressed in dB.

*Total Harmonic Distortion (THD)* – measured in the frequency domain for a full-scale sine wave output, as a ratio (expressed in dB) between the desired sine wave amplitude and the root-mean-square sum of selected harmonics present in the DAC output.

*Effective Number of Bits (ENOB)* – calculated from the SNDR for a full-scale sine wave as

$$ENOB = \frac{SNDR_{dB} - 1.76}{6.02} \text{ bits} \tag{5.95}$$

Several of these important quantities are shown graphically in Figure 5-38 for additional clarity.

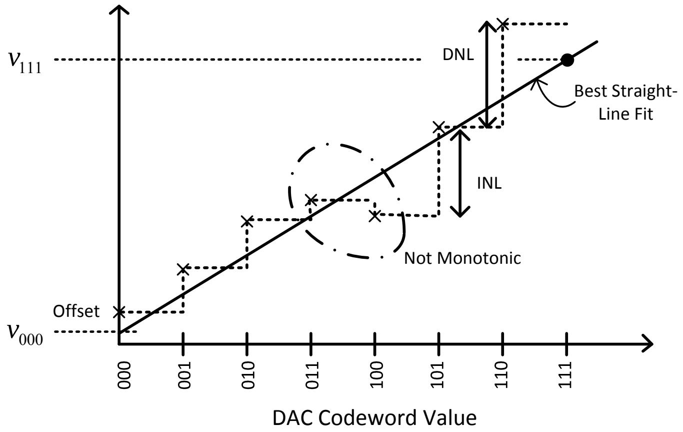


**Figure 5-38** DAC imperfections illustrated for a 3-bit converter

### 5.7.1 Unequal Rise and Fall Times

Unequal rise and fall times in the buffer electronics following the DAC core shown in Figure 5-39 can lead to additional harmonic content at the DAC output. The ideal DAC output is shown in the top of the figure whereas the DAC buffer's finite rise and fall times lead to the familiar decayed exponential shaping in the middle and lower portions of the figure.

An infinite precision DDS / DAC ideally produces a sinusoidal stair-step output voltage waveform having only one Fourier component within the first Nyquist zone (i.e., $0 < f < F_{Clock} / 2$). Unequal rise and fall times will, however, create harmonic distortion products at integermultiples of the desired sinusoidal output frequency which can also fall within the first Nyquist zone. An exaggerated situation is shown in Figure 5-40 for the case of a sinusoidal signal.

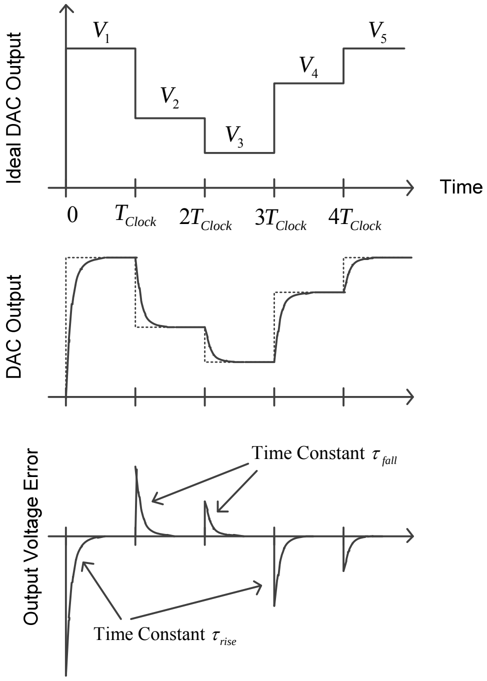


**Figure 5-39** DAC finite rise and fall times. Output noise and spurious levels are increased when $\tau_{rise}$ and $\tau_{fall}$ are unequal due to the underlying nonlinearities involved. The $\tau_{rise} = \tau_{fall}$ case corresponds to a simple lowpass filtering operation of the ideal stair-step signal when $T_{Clock} >> \tau_{rise}$.

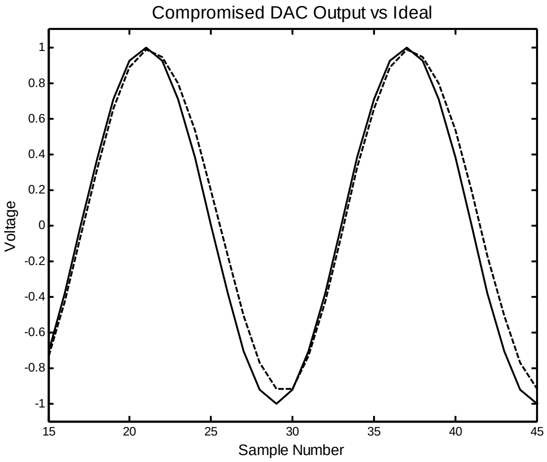


**Figure 5-40** Ideal DAC output versus compromised output[28] in the case of a sinusoid. Exaggerated case shown with $F_{Sine} = 1$ kHz, sample rate $F_{Clock} = 16$ kHz, $\tau_{rise} = 0.5 / F_{Clock}$, $\tau_{fall} = 1 / F_{Clock}$.

---

[28] Computed using u16937_dac_risefall.m.

The Fourier transform of the output sinusoid can be computed by breaking the time domain signal into its constituent steps and summing their respective transforms. The Laplace transform for the $n^{th}$ time segment spanning $(n-1)T_{Clock}$ to $n\ T_{Clock}$ is given by

$$L_n = \left\{ \begin{array}{l} V_n \dfrac{1-\exp\left(-sT_{Clock}\right)}{s} \\ +\left[V_n - V_{x(n-1)}\right] \dfrac{\exp\left[-\left(s+\dfrac{1}{\tau_n}\right)T_{Clock}\right]-1}{1+s\tau_n}\tau_n \end{array} \right\} \exp\left[-s\left(n-1\right)T_{Clock}\right] \tag{5.96}$$

where the $V_{x(n-1)}$ value corresponds to the voltage value at the *end* of the previous time interval, and $\tau_n$ is equal to $\tau_{rise}$ if the signal is increasing in value during the time segment or $\tau_{fall}$ if the signal is decreasing during the time segment. The Fourier transform can be directly computed from (5.96) leading to the power spectral density, and predicted harmonic levels easily follow. Harmonic levels are shown in Figure 5-41 and Figure 5-42 for two different rise-time cases. In both cases, the appearance of unwanted harmonic distortion products is crushed if the rise and fall times can be kept approximately equal.

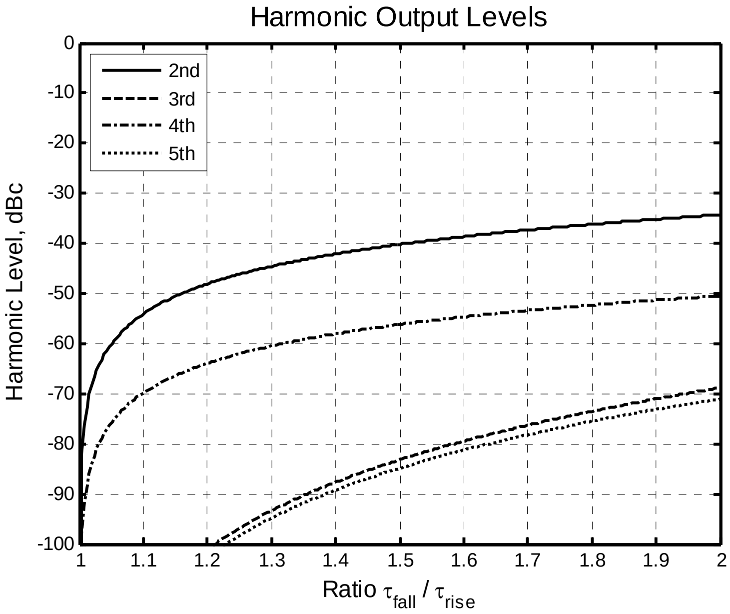


**Figure 5-41** Predicted harmonic levels versus imbalance in rise and fall times. $F_{Sine}$ = 1 kHz, $F_{Clock}$ = 16 kHz, $\tau_{rise} = T_{Clock} / 4$.

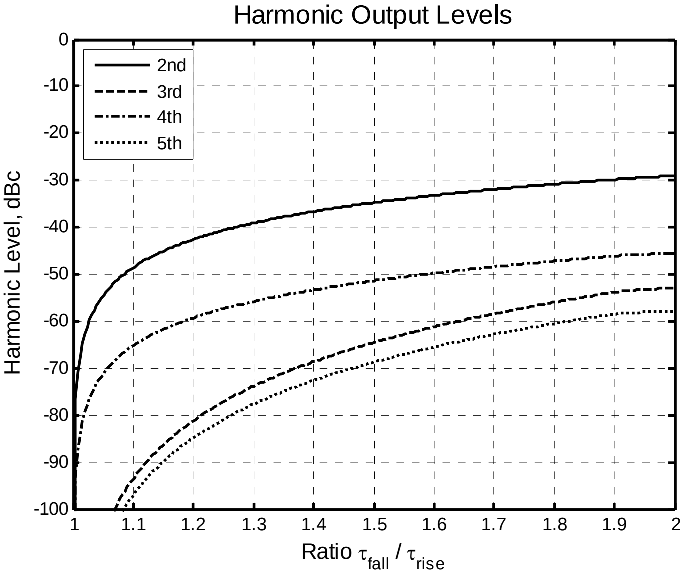


**Figure 5-42** Predicted harmonic levels versus imbalance in rise and fall times. $F_{Sine}$ = 1 kHz, $F_{Clock}$ = 16 kHz, $\tau_{rise} = T_{Clock}$.

### 5.7.1.1 Return-to-Zero to Combat Unequal Rise and Falls Times

Unequal DAC rise and fall times lead to harmonic distortion as discussed in the previous section. Even though the harmonic distortion becomes negligible as the rise and fall times become equal, sufficient equality is difficult to achieve at high clock rate frequencies. Operating the DAC in a return-to-zero manner as shown in Figure 5-43 can combat this issue very effectively.

The basic problem illustrated in Figure 5-43 is that the individual areas of each pulse region shown in the center diagram are not uniformly proportional to the respective pulse-heights $V_k$. When the DAC output is composed of return-to-zero pulses as shown in Figure 5-43, however, each pulse contains ample time to fully exercise complete pulse rise and fall regions.

Referring to the first pulse region shown in Figure 5-43, the pulse area $A_p$ is given by

$$\begin{aligned} A_p &= \int_0^{T_p} V_1\left(1-e^{-t/\tau_{rise}}\right)dt + Vx_1 \int_0^{T_{Clock}-T_p} e^{-t/\tau_{fall}}\,dt \\ &= V_1 T_p + V_1\tau_{rise}\left(e^{-T_p/\tau_{rise}}-1\right) + Vx_1\tau_{fall}\left[1-e^{-\left(T_{Clock}-T_p\right)/\tau_{fall}}\right] \end{aligned} \qquad (5.97)$$

where $Vx_1$ is the ultimate pulse amplitude attained prior to the fall-time region. Normally the rise and fall times are very short with $T_P >> \tau_{rise}$ and $T_P >> \tau_{fall}$ being valid assumptions. This makes it possible to neglect the exponential terms in (5.97) and take $Vx_1 = V_1$ thereby producing

$$A_p \cong V_1 T_p \left(1 + \frac{\tau_{fall}-\tau_{rise}}{T_p}\right) \qquad (5.98)$$

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.

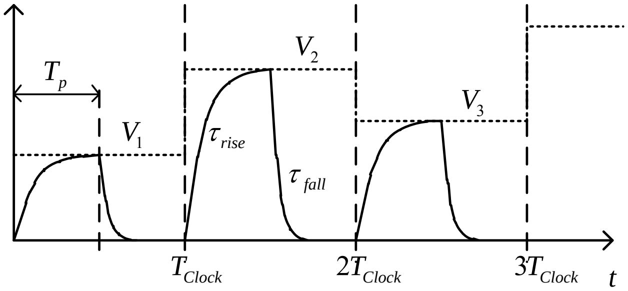


**Figure 5-43** Return-to-zero DAC output pulses

### 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 Sin(x) / x Compensation

An ideal DAC creates aliasing and a 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 by[29]

---

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

$$D(s)=\frac{1-\exp\left(-sT_{Clock}\right)}{s}\left[\frac{1}{T_{Clock}}\sum_{m=-\infty}^{+\infty}V\left(s-j\frac{2\pi m}{T_{Clock}}\right)\right] \tag{5.99}$$

The input signal spectrum $V(\ s\ )$ is assumed to already be band-limited as required by the Nyquist theorem. Equation (5.99) can be rewritten as

$$D(s)\Big|_{s=j\omega}=\exp\left(-j\omega T_{Clock}/2\right)\frac{\sin\left(\omega T_{Clock}/2\right)}{\omega T_{Clock}/2}\sum_{m=-\infty}^{+\infty}V\left(j\omega-jm\omega_{Clock}\right) \tag{5.100}$$

where $\omega_{Clock} = 2\ \pi / T_{Clock}$. The first six Nyquist zones in Figure 5-3 are labeled here in Figure 5-44. Each zone contains essentially the same spectral information of $V(\ s\ )$ albeit shifted in frequency, spectrally inverted for the even-numbered zones, and predictably weighted by the sin( $x$ ) / $x$ frequency domain shaping factor.

In general, it is possible to use the signal content of any Nyquist zone for the DDS output although practical considerations usually limit consideration to the first two or three zones shown in the diagram. DAC imperfections (e.g., nonlinearities) normally degrade the spectral performance for the higher-order zones. The desired signal becomes increasingly difficult to filter out amid the stronger lower-order Nyquist zones if a higher-order zone is adopted for use.

The quantization noise spectrum is also shaped by the same sin( $x$ ) / $x$ frequency domain shaping as the desired signal. Consequently, even though the DDS signal amplitude in the 2nd or 3rd Nyquist zone is smaller than in the first Nyquist zone, the quantization noise level is theoretically reduced by the same amount.

The sin( $x$ ) / $x$ shaping can only be compensated for in one Nyquist zone at a time by virtue of the Nyquist theorem. The $n^{th}$ Nyquist zone extends from $(\ n-1\ )\ F_{Clock} / 2$ to $n\ F_{Clock} / 2$ with the fundamental Nyquist zone designated as the first zone. It is helpful to substitute $ju = j\omega - jm\omega_{Clock}$ into (5.100) and rewrite it as

$$D_m\left(ju+jm\omega_{Clock}\right)=\left(-1\right)^m\exp\left(-j\frac{uT_{Clock}}{2}\right)\left[\frac{\sin\left(\frac{uT_{Clock}}{2}+m\pi\right)}{\frac{uT_{Clock}}{2}+m\pi}\right]V\left[ju\right] \tag{5.101}$$

In this context, $u$ is limited to $-\omega_{Clock} / 2 < u < \omega_{Clock} / 2$ because $V(\ )$ is band-limited to that range. Compensating the $n^{th}$ Nyquist zone entails using the correct the $D_m$ along with a possible spectral inversion of $u$ as summarized in Table 5-6.

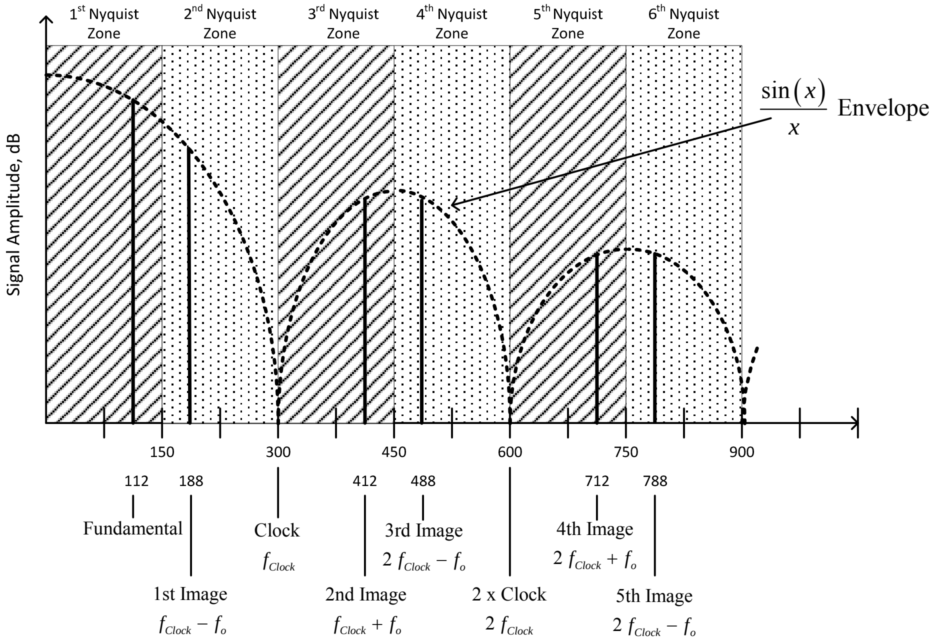


**Figure 5-44** Ideal DAC output sin( *x* ) / *x* shaping with the first six Nyquist zones shown

**Table 5-6** Parameter Values for $n^{th}$ Nyquist Zone Compensation

| **Nyquist Zone, *n*** | ***m*** | **Polarity for *u, p*** |
|---|---|---|
| 1 | 0 | + |
| 2 | 1 | – |
| 3 | 1 | + |
| 4 | 2 | – |
| $n$ | $\left\lfloor \frac{n}{2} \right\rfloor$ | $(-1)^{n+1}$ |

In order to faithfully reproduce *V* ( *ju* ) for the $n^{th}$ Nyquist zone, the sin( *x* ) / *x* portion of (5.101) must be counteracted by the compensation function

$$G_{DAC}(u,m,p) = \left[ \frac{\sin\left( m\pi + p\frac{uT_{Clock}}{2} \right)}{m\pi + p\frac{uT_{Clock}}{2}} \right]^{-1} \tag{5.102}$$

This compensation can be implemented by using a FIR filter immediately before the DAC that adequately matches the frequency domain description given by (5.102). The associated impulse response of the filter $g_{DAC}$ can be obtained using a number of traditional digital filter design methods including (i) evaluation of (5.102) over a grid of frequencies followed by an inverse

FFT,[30] (ii) use of the eigenfilter design method,[31] or (iii) using the *fir2* function in MATLAB which is based upon the frequency sampling method.[32] For the second and higher Nyquist zones, the gain in the vicinity of $u = 0$ in (5.102) must be constrained because it otherwise becomes infinite. This gain modification can be done by limiting the maximum gain that is allowed to be used in the compensation function or by following (5.102) with an additional highpass filtering function as done in the computed examples which follow.

Two examples can provide additional insight. In the first example, the $\sin(x)/x$ shaping is to be compensated for in the first Nyquist zone using a length-17 FIR filter. The required frequency-domain compensation function is shown in Figure 5-45 and the computed length-17 FIR tap-weights are shown in Figure 5-46. The resultant passband flatness in the first Nyquist zone is shown in Figure 5-47. Higher-order FIR filters are usually required to compensate higher-order Nyquist zones as illustrated in the second example which deals with the third Nyquist zone. In this example, a length-41 FIR filter is required to achieve approximately the same degree of flatness. The required frequency domain compensation function is shown in Figure 5-48, the computed FIR tap-weights in Figure 5-49, and the resultant compensation flatness for the third Nyquist zone is shown in Figure 5-50.

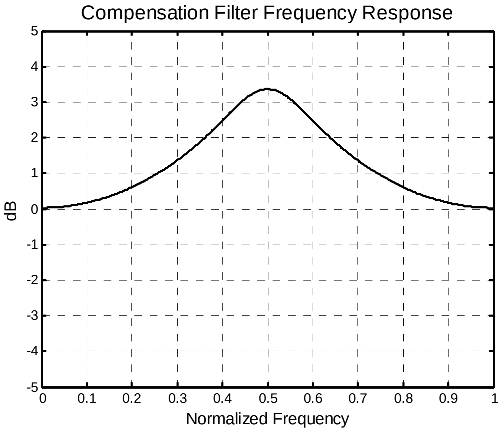


**Figure 5-45** Example 1 frequency domain compensation function for the first Nyquist zone[33]

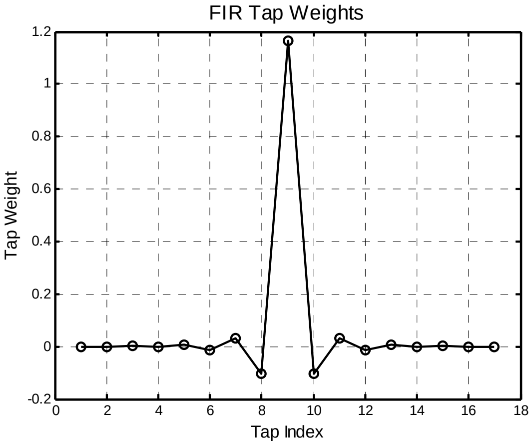


**Figure 5-46** Example 1 FIR filter tap weights computed

### 5.8.1 Return-to-Zero to Reduce sin(x)/x Roll-Off

One of the most simple methods to lessen $\sin(x)/x$ roll-off in the first Nyquist zone is to configure DAC operation so that data-zeros are inserted between each input sample. Although this normally requires the DAC to be operated at twice the speed, the first spectrum null is pushed out to twice the frequency thereby lessening the lower frequency spectrum roll-off. Operation in the return-to-zero mode was discussed earlier in Section 5.7.1 as a very effective means for combatting distortion products that arise from unequal DAC rise and fall times.

---

[30] MATLAB script u16870_sinc_compensation.m.

[31] Eigenfilter method is described in Section 3.8 of [6].

[32] MATLAB script u16872_sinc_comp.m.

[33] MATLAB script u16872_sinc_comp( 1, 16 ).

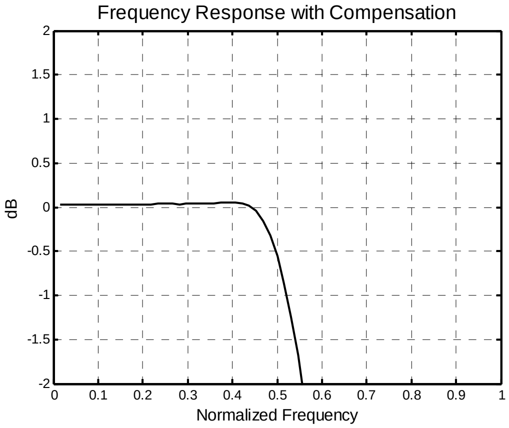


**Figure 5-47** Example 1 resultant frequency domain flatness showing that the first Nyquist zone is compensated for very well. The normalized frequency is with respect to the sampling rate.

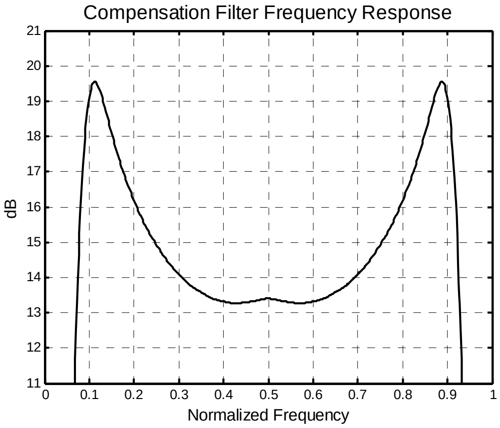


**Figure 5-48** Example 2 frequency domain compensation function for the third Nyquist zone[34]

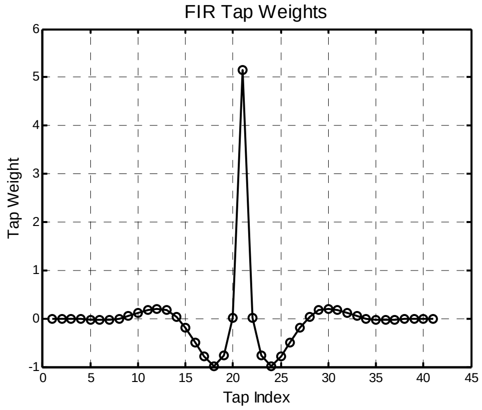


**Figure 5-49** Example 2 FIR filter tap weights computed

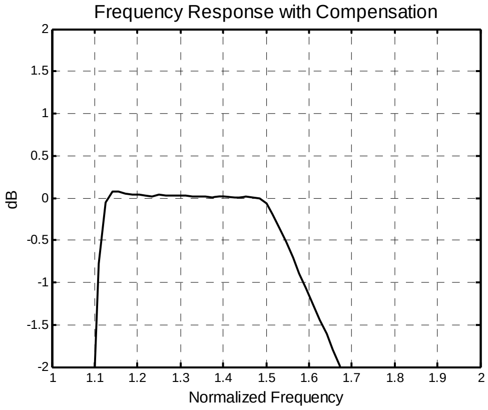


**Figure 5-50** Example 2 resultant frequency domain flatness showing that the third Nyquist zone is compensated for very well. The normalized frequency is with respect to the sampling rate.

---

[34] MATLAB script u16872_sinc_comp( 3, 40 ).

## 5.9 Avoiding Harmonic Distortion Products

Harmonic distortion terms arising from imperfections in the DAC are difficult to deal with because techniques like dither are rather ineffective in suppressing them. One of the most effective methods to avoid these products is to employ a variable modulus divider following the DDS which makes it possible to operate the DDS over a frequency-limited "sweet spot" [$F_{Min}$, $F_{Max}$] where its performance is excellent, and realize extended coverage bandwidth [ $F_1$, $F_2$ ] with appropriate changes in the output divide ratio. The resultant configuration is shown in Figure 5-51.

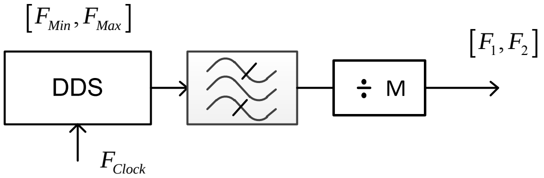


**Figure 5-51** DDS augmented with variable-modulus output divider in order to realize an expanded percentage bandwidth while operating the DDS over a restricted frequency range for better harmonic distortion performance

As an illustrative example, assume a DDS is operating from a 200 MHz clock. Using the theory developed in Section 5.4, the second Nyquist zone spanning 137 MHz to 147 MHz is void of harmonic distortion products through the 5th order. The bandpass filter is used to strongly attenuate any out-of-band spurious tones which may be present. This 10 MHz tuning range represents a bandwidth percentage of only ±3.52%. If this DDS is combined with a variable-modulus divider having divide ratios from 14 to 23, an output frequency range in excess of 6 MHz to 10 MHz can be realized as shown in Table 5-7; a percentage tuning range of ±25% centered on 8 MHz.

**Table 5-7** DDS Coverage from 6 MHz to 10 MHz Avoiding Harmonic Distortion

| **Output Frequency Span** | | | | |
|---|---|---|---|---|
| **$F_1$, MHz** | **$F_2$, MHz** | **$M$** | **DDS $F_{Min}$** | **DDS $F_{Max}$** |
| 9.80 | 10.00 | 14 | 137.20 | 140.00 |
| 9.15 | 9.80 | 15 | 137.25 | 147.00 |
| 8.60 | 9.15 | 16 | 137.60 | 146.40 |
| 8.15 | 8.60 | 17 | 138.55 | 146.20 |
| 7.70 | 8.15 | 18 | 138.60 | 146.70 |
| 7.25 | 7.70 | 19 | 137.75 | 146.30 |
| 6.90 | 7.25 | 20 | 138.00 | 145.00 |
| 6.60 | 6.90 | 21 | 138.60 | 144.90 |
| 6.25 | 6.60 | 22 | 137.50 | 145.20 |
| 6.00 | 6.25 | 23 | 138.00 | 143.75 |

In order to have continuous frequency coverage at the divider output, the minimum divide ratio must be constrained as

$$M_{Min} > \frac{F_{Min}}{F_{Max} - F_{Min}} \tag{5.103}$$

The lower frequency limit $F_1$ can be made arbitrarily low by increasing $M$ as needed whereas the upper frequency limit for continuous coverage $F_2$ is limited by (5.103) to $F_{Max}$ / $M_{Min}$. In the present example, $M_{Min} = 14$ and this limits $F_2$ to be less than or equal to 10.5 MHz.

### 5.10 Sinusoidal DDS Output Spur Reduction Using Dither

There are a number of ways in which unwanted spurious tones can be created at the output of a sinusoidal DDS. These mechanisms include:

- Deterministic DAC clock jitter (Section 5.5.1.1)
- Phase truncation (Section 5.5.2)
- DAC quantization (Section 5.5.3)
- $\theta$ to sin( $\theta$ ) imperfections (Section 5.6)
- DAC imperfections (Section 5.7)

*Dithering* can be incorporated into a DDS in order to combat some of these spurious mechanisms. In general, dithering reduces the spurious sidebands by randomizing the underlying quantization or error process responsible for creating the spurs. The performance cost associated with dithering is an increase in the output noise floor level.

Dithering takes the form of a random numerical quantity which is normally added into the computational flow of the DDS. Dithering can be inserted into the DDS by (i) dithering the input frequency control word, (ii) adding it in immediately before the $\theta$ to sin( $\theta$ ) conversion, (iii) modifying the amplitude of the sin( $\theta$ ) output after the conversion, or other equivalent means as suggested in Figure 5-52. These techniques are examined in the sections that follow.

Dithering is not the only way in which periodic errors within a DDS can be defeated, but it was one of the first methods appended to the original DDS concept to fulfill this purpose. The noise shaping methods described in Section 5.11 are also very effective for reducing unwanted spurious tones and they offer additional benefits as well.

### 5.10.1 Frequency Control Word Dithering

Frequency control word dithering is represented by Dither Source 1 in Figure 5-52. This dither source must be void of any average dc term so that the desired output center frequency is not altered. It is usually desirable to form the dithering source as a uniform random number generator followed by a 1st-order differentiator as shown in Figure 5-53 in order to avoid this frequency error issue at the DDS output. In this form, the highpass filter precisely compensates for the subsequent phase integration which occurs in the phase accumulator making this method equivalent to the phase dithering method discussed in the next section.

In the context of Figure 5-52 with $N = 17$ bits, $W = 5$ bits,[35] and no dithering, the worst-case phase-truncation related spur levels are about –30 dBc as shown in Figure 5-54. The output noise level is completely overshadowed by the discrete spurious spectrum elements. Inclusion of the frequency-word dithering shown in Figure 5-53 eliminates all of the observable discrete spurious tones when $\alpha = 0.90$, but this choice for $\alpha$ results in substantial close-in phase noise degradation as shown in Figure 5-55. Setting the highpass filter parameter $\alpha = 1.0$ eliminates this unwanted close-in phase noise as shown in Figure 5-56 but some spurious tones still survive. The $\alpha$ parameter provides a clear means to trade off spurious performance with close-in phase noise performance.

---

[35] Accumulator value spans over [ 0, $2^N - 1$]. $\theta_W$ spans over [0, $2^W - 1$], both in unsigned binary form here.

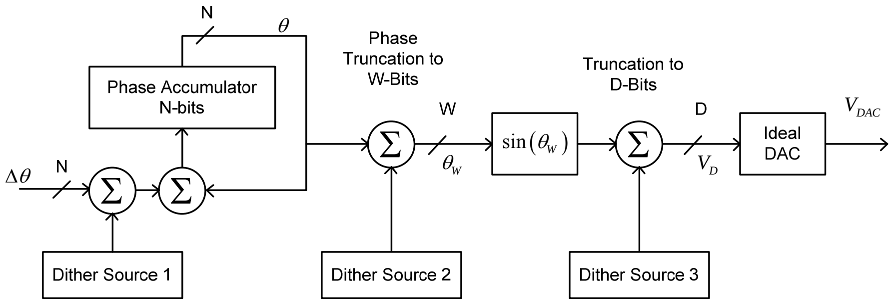


**Figure 5-52** Possible dither insertion points within a DDS

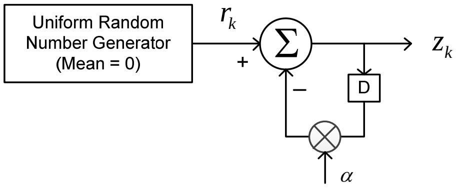


**Figure 5-53** Dither source configured with a 1st-order differentiator / highpass filter making it suitable for dithering the frequency word input in Figure 5-52. Parameter $\alpha$ must be unity in order to remove any hint of random center frequency walking at the DDS output.

### 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 $\theta$ 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]

$$SNR_U = 6.02W - 6.93 \text{ dB} \tag{5.104}$$

and a lower-bound is similarly given by

$$SNR_L = 6.02W - 8.18 \text{ dB} \tag{5.105}$$

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

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

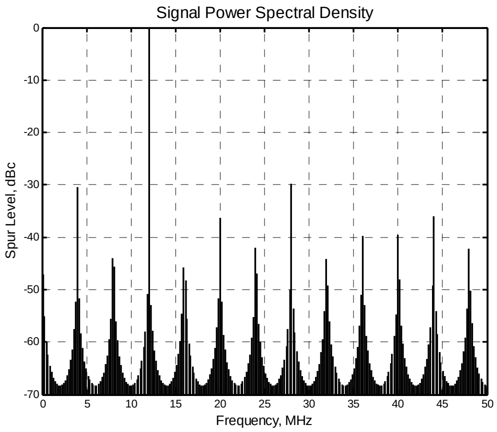


**Figure 5-54** DDS with $F_{Clock}$ = 100 MHz, $N$ = 17 bits, $W$ = 5 bits, without dithering

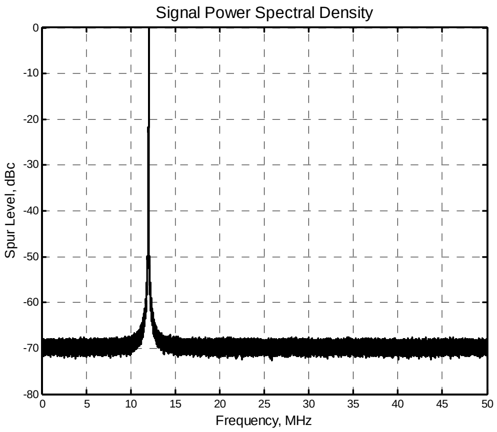


**Figure 5-55** DDS with $F_{Clock}$ = 100 MHz, $N$ = 17 bits, $W$ = 5 bits, with frequency-word dithering employed.[36] Dithering value $r_k$ in Figure 5-53 is uniformly distributed over $[-2^{11}, 2^{11}]$. Parameter $\alpha$ = 0.90 which leads to substantial close-in phase noise appearing.

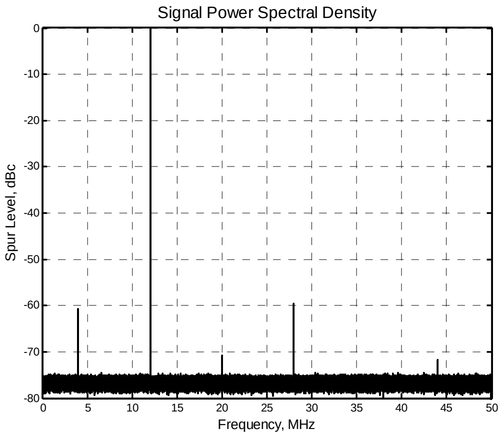


**Figure 5-56** Identical to Figure 5-55 except that $\alpha$ = 1.[37] Although some spurious artifacts remain, no close-in phase noise degradation is observable.

The worst-case spur level with phase dithering applied is given by [24]

$$L_{Spur} = -12.04W + 7.84 \text{ dBc} \tag{5.107}$$

The spur level observed is frequently better than that predicted by (5.107) because this is a fairly conservative worst-case bound.

---

[36] Using u16972_dds_dither2( 5, 12, 0.90, 1).

[37] Using u16972_dds_dither2( 5, 12, 1.0, 1).

The benefits of phase dithering are very visible in the following example. In the original DDS, only 5 phase-bits are used for the phase representation after truncation and the typical worst-case spur levels are about –30 dBc as shown in Figure 5-57. When phase dithering is added to the same DDS, the worst-case spur level is reduced to about –60 dBc as shown in Figure 5-58. The slight penalty in $C / N_o$ due to the dither is inconsequential compared to the severe discrete spur nature of the DDS output without dithering present.

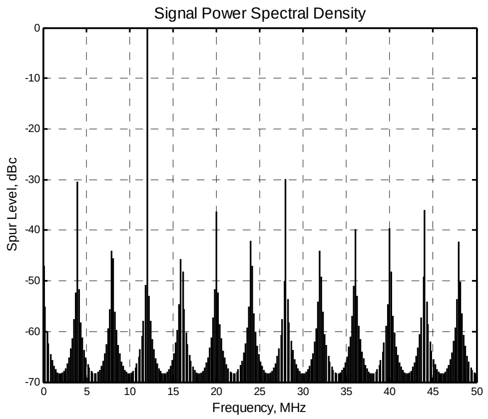


**Figure 5-57** DDS output power spectral density for $F_{Clock}$ = 100 MHz, $F_{DDS}$ = 12 MHz, 5 phase-bits without phase-dithering.[38]

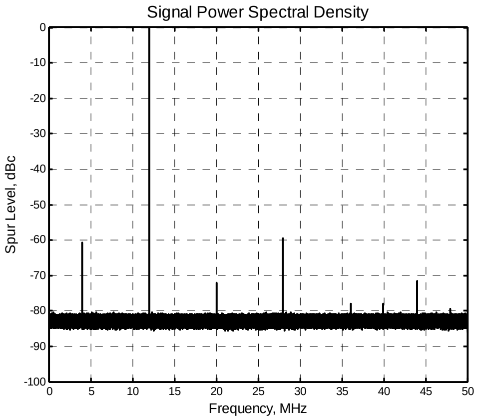


**Figure 5-58** DDS output power spectral density for $F_{Clock}$ = 100 MHz, $F_{DDS}$ = 12 MHz, 5 phase-bits and phase-dithering used.[39]

### 5.10.3 Sin( $\theta$ ) Amplitude Dithering

Amplitude dithering is implemented by using *Dither Source 3* in Figure 5-52. Without any dithering present, DAC-related quantization can lead to appreciable spurious levels because of the significant correlation in the error sequences. An example output spectrum is shown in Figure 5-59. If rounding is incorporated immediately before the DAC quantization, the output spectrum is changed appreciably to that shown in Figure 5-60. The rounding does very little to suppress the more serious spurs, but it does lower the apparent noise floor significantly. When dithering is added immediately before the DAC quantization, the discrete spurs are completely eliminated as shown in Figure 5-61. The only degradation caused by inserting the dither is that the total quantization noise is effectively doubled [26], but since this noise is spread across the full Nyquist frequency range, this is usually a very acceptable performance penalty to accommodate.

Even though the peak-to-peak amplitude of the dither source is only a single DAC LSB, it is effective in eliminating the discrete spurious frequencies because it randomizes the resultant quantization error sequence thereby removing any residual periodicities.

---

[38] From u16968_dds_dither1.m.

[39] From u16968_dds_dither1.m.

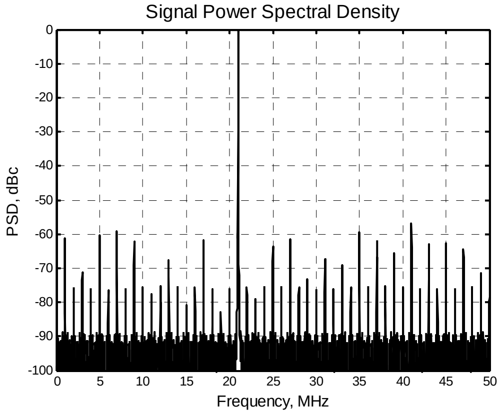


**Figure 5-59** Straight DDS with no spur-reduction methods used.[40] $F_{Clock}$ = 100 MHz, $F_{Out}$ = 21 MHz thereby leading to 1 MHz spurs. Ideal DAC with 8-bit precision assumed.

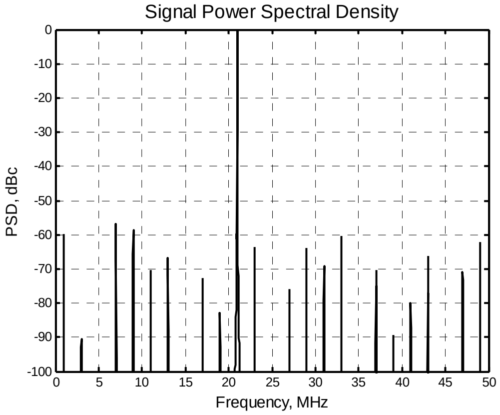


**Figure 5-60** DDS with rounding immediately ahead of the DAC quantization.[41] $F_{Clock}$ = 100 MHz, $F_{Out}$ = 21 MHz thereby leading to 1 MHz spurs. Ideal DAC with 8-bit precision assumed.

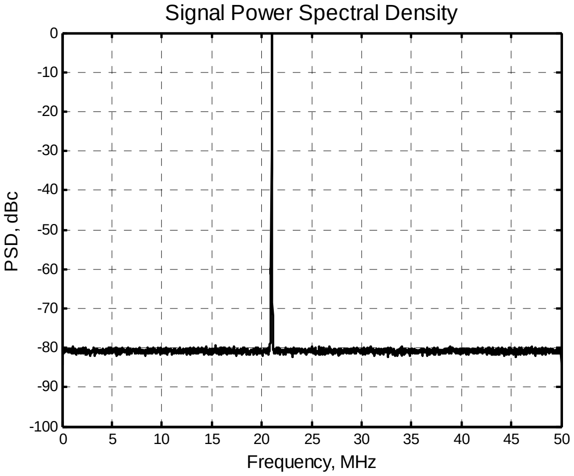


**Figure 5-61** DDS with dithering added immediately ahead of the DAC quantization.[42] $F_{Clock}$ = 100 MHz, $F_{Out}$ = 21 MHz. Ideal DAC with 8-bit precision assumed. No evidence of discrete spurs remains.

---

[40] MATLAB call u16976_dds_dither3(1).
[41] MATLAB call u16976_dds_dither3(2).
[42] MATLAB call u16976_dds_dither3(3).

### 5.10.4 Sin( $\theta$ ) Amplitude Dithering Using Lowpass-Filtered Noise

Noise insertion can be used to combat output spurious components that arise from DAC-related finite precision effects.[43] This method is not effective for suppressing spurious tones which result from phase truncation effects, however. A block diagram of the situation considered here is shown in Figure 5-62 where the *ideal* sine wave input has already been created elsewhere within the DDS. The filtered noise is intended to remove any periodicity in the quantization error sequence in the same way Nyquist-rate dithering was used in the previous section.

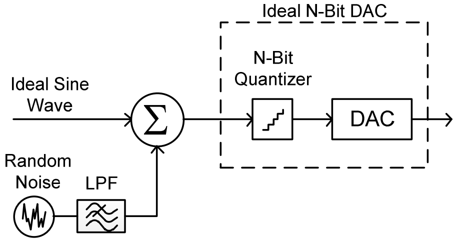


**Figure 5-62** Additive noise approach for reducing DAC finite-precision related spurious tone outputs

In the following discussion, the amount of random noise added is measured in terms of the noise variance at the output of the LPF and the bandwidth of the LPF is given in terms of a percentage of the DAC clock frequency $F_{Clock}$. The noise at the LPF input is assumed to be mean-zero, uniformly distributed, and created at the same $F_{Clock}$ rate. As shown momentarily, the relationship between the noise variance, LPF bandwidth, and resultant spurious suppression is fairly simple. In each example, the DAC precision is always 8-bits, the sampling rate is fixed at 100 Msps, and the lowpass filter is composed of two cascaded $N = 5$ Butterworth lowpass filters.

The spurious signature can vary dramatically depending upon the frequency of the output sine wave being synthesized as shown in Figure 5-63 and Figure 5-64. When the lowpass filter bandwidth is 2% of $F_{Clock}$, only a small amount of noise is required to eliminate the discrete spurs as shown in Figure 5-65 and Figure 5-66. If the lowpass filter bandwidth is decreased, the variance of the filtered noise must be increased in order to obtain the same amount of spurious suppression and flat noise floor behavior as shown in Figure 5-67 and Figure 5-68.

In the Nyquist-rate dithering discussed in the previous section, the residual quantization error was uncorrelated because the dithering noise was completely random on a sample-by-sample basis. In the filtered noise case, however, the filtering introduces correlation between samples which can only be offset by using more noise.

For the Nyquist-rate dithering case, sample-adjacent dither values are assumed to be completely uncorrelated so that $\mathbf{E}( d_k d_{k-1} ) = 0$ and

$$\mathrm{E}\left\{\left[d_k - d_{k-1}\right]^2\right\} = 2\mathrm{E}\left\{d_k^2\right\} \tag{5.108}$$

where the discrete-time dither values are represented by $d_k$. In the filtered noise case, however, adjacent dither values are correlated. Owing to the high over-sampling rate at the lowpass filter output, it is convenient to discuss this correlation issue in terms of continuous-time values rather than discrete-time, and the correlation function is given by

---

[43] Sinusoidal tones may also be used but this method is not addressed here.

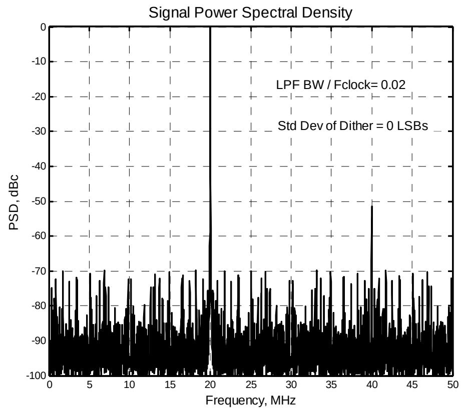


**Figure 5-63** In the absence[44] of any corrective dithering, the worst-case spur is only about –50 dBc with a plethora of other spurs at about –70 dBc when synthesizing $F_{DDS}$ = 20 MHz.

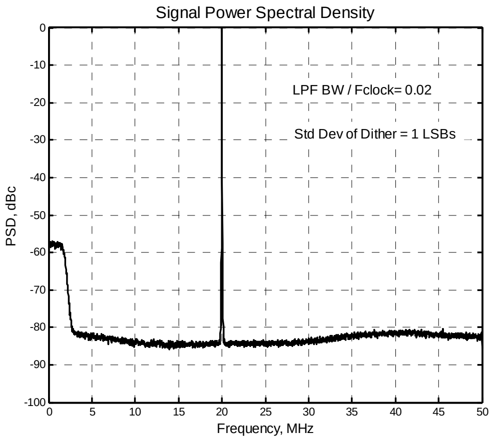


**Figure 5-65** Insertion of lowpass filtered noise suppresses the discrete spurs entirely but some noise floor irregularity is still present.[46] Lowpass filter bandwidth = 2% of $F_{Clock}$. Standard deviation of filtered noise = 1 DAC LSB.

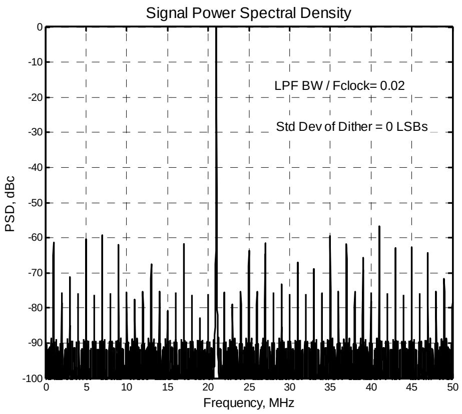


**Figure 5-64** In the absence[45] of any corrective dithering, the worst-case spur is about –57 dBc with a plethora of other spurs at about –60 dBc when synthesizing $F_{DDS}$ = 21 MHz.

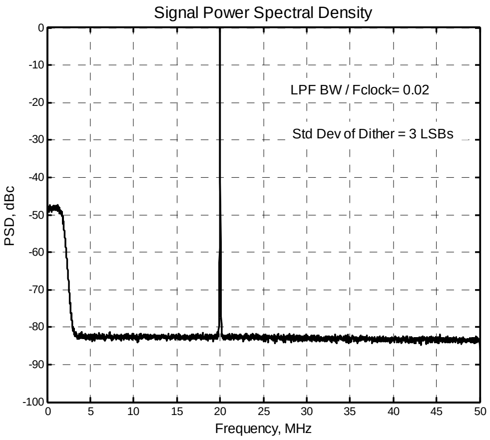


**Figure 5-66** Insertion of lowpass filtered noise suppresses the discrete spurs entirely but some noise floor irregularity is still present.[47] Lowpass filter bandwidth = 2% of $F_{Clock}$. Standard deviation of filtered noise increased to 3 DAC LSBs in order to obtain a relatively flat noise floor.

---

[44] From calling u16977_dds_dither4( 8, 20, 2, 0 ).

[45] From calling u16977_dds_dither4( 8, 21, 2, 0 ).

[46] From From calling u16977_dds_dither4( 8, 20, 2, 1 ).

[47] From From calling u16977_dds_dither4( 8, 20, 2, 3 ).

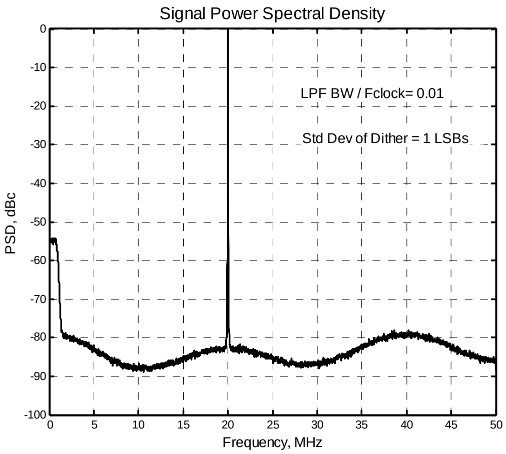


**Figure 5-67** When the lowpass filter bandwidth is reduced to only 1% of $F_{Clock}$, 1 LSB of dithering is still sufficient to suppress discrete spurs but the noise floor behavior is erratic as shown.[48]

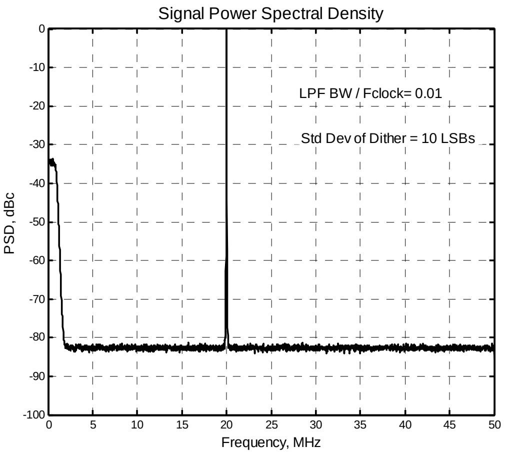


**Figure 5-68** The dither standard deviation must be increased to roughly 10 LSBs to obtain consistent noise floor behavior when the lowpass filter bandwidth is reduced to 1%.[49]

$$R_d(\tau) = \int_{-\infty}^{+\infty} S_d(f) e^{j2\pi f\tau} df \tag{5.109}$$

by way of the Wiener-Khintchine theorem for wide-sense stationary random processes. In this form, $S_d(f)$ represents the power spectral density of the random noise source at the lowpass filter input. When this noise source is spectrally white and the lowpass filter has a brick wall attenuation characteristic with a corner frequency of $F_c$ Hertz, (5.109) is given by

$$R_d(\tau) = \int_{-F_c}^{F_c} S_d(0) e^{j2\pi f\tau} df = 2F_c S_d(0) \frac{\sin(2\pi F_c \tau)}{2\pi F_c \tau} \tag{5.110}$$

Dither samples which are only separated by a few clock intervals $T_{Clock}$ are highly correlated since $F_c << F_{Clock}$.

The Nyquist-rate dithering used a random value having a peak-to-peak value range of only one LSB to completely eliminate perceptible discrete spurious outputs. In a similar manner, it is reasonable to conclude that the variation of the filtered noise from one sample time to the next must also be at least one LSB in order to be effective for reducing unwanted spurious tones. The quantity of interest in this respect is given by

$$\begin{aligned} \mathrm{E}\left\{\left[d(t) - d(t - T_{Clock})\right]^2\right\} &= 2\left[R_d(0) - R_d(T_{Clock})\right] \\ &= 8\int_0^{+\infty} S_d(f) \sin^2(\pi f\, T_{Clock})\, df \end{aligned} \tag{5.111}$$

---

[48] From From calling u16977_dds_dither4( 8, 20, 1, 1 ).
[49] From From calling u16977_dds_dither4( 8, 20, 1, 10 ).

When $S_d(f)$ is spectrally white and the lowpass filter bandwidth is much less than $F_{Clock}$, (5.111) can be closely approximated by

$$\mathrm{E}\left[\Delta_d^2\left(T_{Clock}\right)\right] \cong \frac{4}{3}\left[2F_c S_d\left(0\right)\right]\left(\pi F_c T_{Clock}\right)^2 \tag{5.112}$$

where the bracketed quantity is equal to the variance of the filtered noise. Generally speaking, the expectation must evaluate to at least one LSB$^2$ in order to be effective for suppressing unwanted discrete spurs. This result provides some guidance about how the noise variance and the noise filter bandwidth can be traded off against each other to achieve the same amount of dithering benefit. Roughly speaking, if the lowpass filter bandwidth $F_c$ is cut in half, the noise variance at the lowpass filter output must be increased by about 4× in order to compensate.

In general, the complexity involved with using lowpass filtered noise for dithering rather than the Nyquist-rate dithering discussed in the previous section is substantial. The most egregious difference between the two methods is that the noise must be filtered out at the DAC output using a highpass or bandpass filter when the Nyquist-rate method is not used. In most cases, using the Nyquist-rate dithering method described in Section 5.10.3 is preferable.

## 5.11 Noise Shaping Methods for Improved Direct Digital Synthesis

Noise shaping techniques can be used effectively to improve the output phase noise and spurious performance of a direct digital synthesizer.[50] This technique was briefly discussed in Section 7.3.7 and Section 9.3.3 of [7] and several pertinent references were provided [29], [30]. A second-order bandpass Δ-Σ method is described in the following section and then subsequently applied to higher-order implementations in order to achieve improved noise and spurious performance.

### 5.11.1 Second-Order Bandstop Noise Shaping

A second-order Δ-Σ architecture is shown in Figure 5-69. The high precision input $x_k$ is truncated to M-bits of precision by the quantizer in order to match the resolution of the output D-to-A converter. The residual error $\Delta_k$ is processed by the feedback paths as shown. The quantizer is represented as a linear summation involving (presumably) uncorrelated quantization error samples $q_k$ as discussed in Section 8.3.3 of [6]. In the DDS application where $x_k = \sin(\theta_k)$, it is assumed the $\sin(\theta_k)$ values are computed with arbitrary precision using CORDIC or other equivalent techniques.

The noise transfer function $H_n(z)$ corresponding to Figure 5-69 is given by

$$H_n\left(z\right) = 1 + a_1 z^{-1} + a_2 z^{-2} \tag{5.113}$$

and the magnitude is given by

---

[50] See [27] and [28] for a partial introduction to this topic.

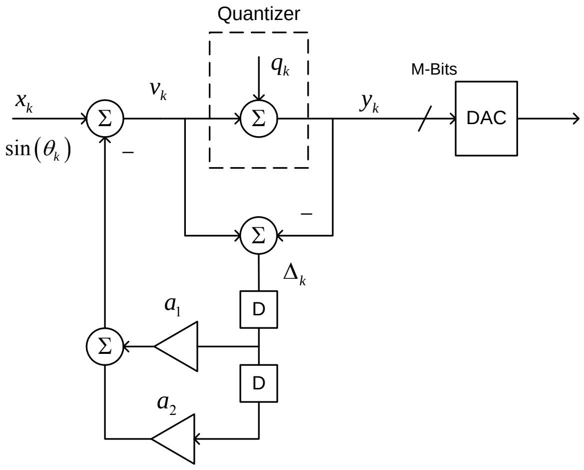


**Figure 5-69** Second-order Δ-Σ modulator used to make up for the finite precision of the M-bit DAC

$$\begin{aligned}\left|H_n(z)\right|^2 &= 1+a_1^2+a_2^2+2a_1(1+a_2)\cos(\theta)+2a_2\cos(2\theta) \\ &= 1+a_1^2+a_2^2-2a_2+2a_1(1+a_2)\cos(\theta)+4a_2\cos^2(\theta)\end{aligned} \tag{5.114}$$

where $\theta = 2\pi f\, T_{Clock}$, $f$ is the frequency of interest, and $T_{Clock}$ is the sampling period. In order for this architecture to serve as a tunable bandstop filter, (5.114) must ideally equate to zero at a prescribed value of $\theta$. It so happens that this is nicely obtained by setting $a_2 \equiv 1$ and choosing

$$a_1 = -2\cos\left(2\pi f_o T_{Clock}\right) \tag{5.115}$$

in which $f_o$ is the center of the desired bandstop characteristic. The resultant noise shaping characteristic using (5.114) is given by

$$\left|H_B(\theta)\right|^2 = a_1^2 + 4a_1\cos(\theta) + 4\cos^2(\theta) \tag{5.116}$$

with $a_1$ given by (5.115).

To illustrate the effectiveness of this method, assume a 200 MHz sampling rate is used along with a 9-bit D-to-A converter, and output frequencies of 41.1 MHz and 21.1 MHz are to be synthesized. Figure 5-70 and Figure 5-71 show the resultant spectrums in the case where no noise shaping is used (i.e., $a_1 = a_2 = 0$). Note the presence of the many high-level spurs even though no DAC harmonic distortion has been included here. If the DAC harmonic distortion is assumed to be –60 dBc for the 2nd through 7th harmonics, the spurious performance is changed from Figure 5-70 and Figure 5-71 to Figure 5-72 and Figure 5-73 respectively.

In order to invoke the bandstop noise shaping, $a_2$ is set to 1 and $a_1$ is set to –0.551946 for the 41.1 MHz case, and set to –1.57645 for the 21.1 MHz case. The noise shaping dramatically reduces the spurious tones present and also suppresses the phase noise close to the carrier as shown in Figure 5-74 and Figure 5-75 when no DAC harmonic distortion is present.

When the same –60 dBc harmonic distortion is present for the DAC 2nd through 7th harmonics, however, the principle spurs become readily apparent as shown in Figure 5-76 and

Figure 5-77. Although all of the other discrete spurs have been eliminated by the noise shaping, the harmonic distortion sets the limit for achievable spurious performance.

It is helpful to look at the noise transfer function characteristic (5.116) in greater detail before leaving this section and re-write it as

$$\left|H_B(\theta)\right|^2 = 4\cos^2(\theta_o) - 8\cos(\theta_o)\cos(\theta) + 4\cos^2(\theta) \tag{5.117}$$

in which $\theta_o$ corresponds to the angle for the desired frequency null. Sometimes it may be convenient to use the gain at dc as a reference gain and it is given by

$$\left|H_B(0)\right|^2 = 4\cos^2(\theta_o) - 8\cos(\theta_o) + 4 = 16\sin^4\left(\frac{\theta_o}{2}\right) \tag{5.118}$$

Similarly, the gain at $F_{Clock}$ / 2 corresponds to $\theta = \pi$ and is given by

$$\left|H_B(\pi)\right|^2 = 4\cos^2(\theta_o) + 8\cos(\theta_o) + 4 = 16\cos^4\left(\frac{\theta_o}{2}\right) \tag{5.119}$$

By inspection, the only time these two gain functions are equal to each other occurs for $\theta_o$ = 90° which corresponds to $F_{Clock}$ / 4.

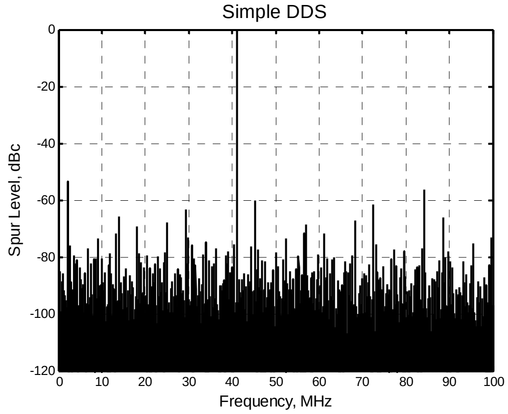


**Figure 5-70** Simple DDS output spectrum[51] for 41.1 MHz. Floating point precision used for sin($\theta$ ) values, 9-bit ideal DAC, sample rate of 200 Msps.

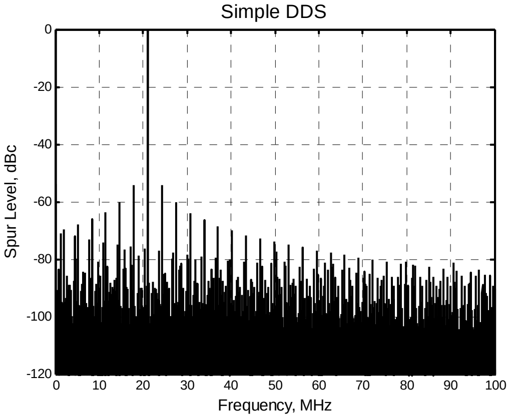


**Figure 5-71** Simple DDS output spectrum for 21.1 MHz. Floating point precision used for sin($\theta$) values, 9-bit ideal DAC, sample rate of 200 Msps.

---

[51] MATLAB script u16283_dds_noiseshaping.m.

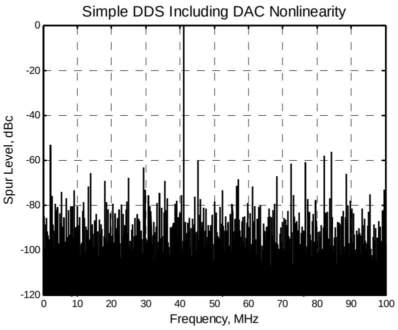


**Figure 5-72** Spurious result corresponding to Figure 5-70 when 2nd through 7th harmonic performance is assumed to be –60 dBc for the DAC

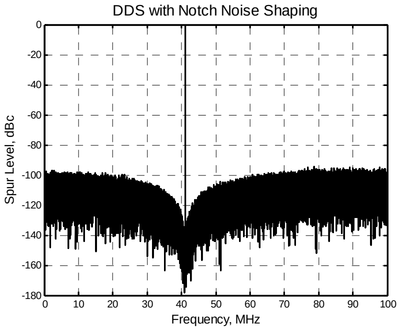


**Figure 5-74** Noise shaping based on Figure 5-69 essentially eliminates the discrete spurious tones that are otherwise present in Figure 5-70, other conditions remaining constant (i.e., 9-bit ideal DAC, 200 MHz sampling rate, 41.1 MHz output)

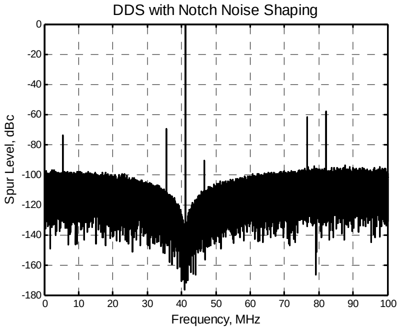


**Figure 5-76** DDS output spurious performance[52] corresponding to Figure 5-74 when –60 dBc harmonic distortion is present for the 2nd through 7th harmonics. The close-in spurious tones are at 35.6 MHz = 200 MHz – 4 $F_{DDS}$ and 46.6 MHz = 6 $F_{DDS}$ – 200 MHz.

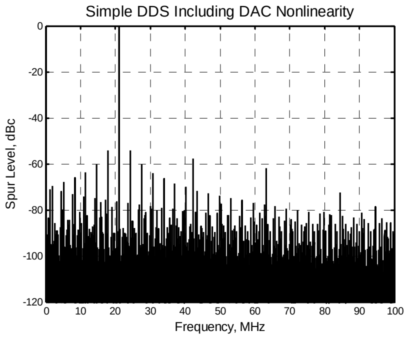


**Figure 5-73** Spurious result corresponding to Figure 5-71 when 2nd through 7th harmonic performance is assumed to be –60 dBc for the DAC

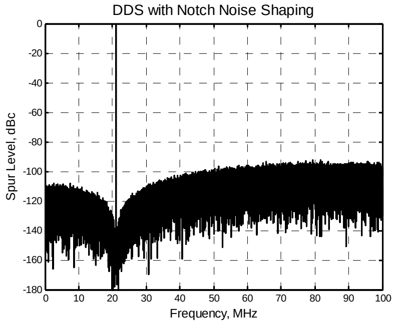


**Figure 5-75** Noise shaping based on Figure 5-69 essentially eliminates the discrete spurious tones that are otherwise present in Figure 5-71, other conditions remaining constant (i.e., 9-bit ideal DAC, 200 MHz sampling rate, 41.1 MHz output)

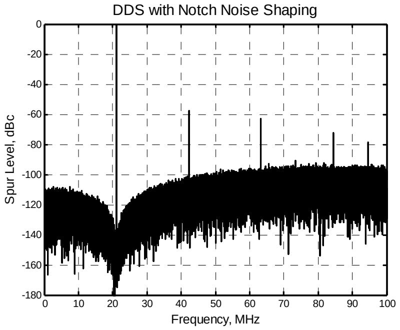


**Figure 5-77** DDS output spurious performance[53] corresponding to Figure 5-75 when –60 dBc harmonic distortion is present for the 2nd through 7th harmonics. The first four spurs are simply harmonics of the fundamental frequency $F_{DDS}$ = 21.1 MHz.

---

[52] MATLAB script u16283_dds_noiseshaping.m.

[53] Ibid.

Since there is only one design parameter $\theta_o$, it is straight forward to create a set of attenuation curves as shown in Figure 5-78. The frequencies ( denoted here by $\theta_\gamma$ ) at which the noise transfer function power-gain is numerically equal to $\gamma$ ( < 1 ) are given by

$$\theta_\gamma = \cos^{-1}\left[\cos\left(\theta_o\right) \pm \sqrt{\frac{\gamma}{4}}\right] \tag{5.120}$$

and the range of legal arguments ( $\theta_o$, $\gamma$ ) is clearly restricted since the arc-cosine is only defined for arguments with a magnitude less than or equal to unity. In general, it is necessary that

$$-1 + \sqrt{\frac{\gamma}{4}} \leq \cos\left(\theta_o\right) \leq 1 - \sqrt{\frac{\gamma}{4}} \tag{5.121}$$

for the result to be defined.

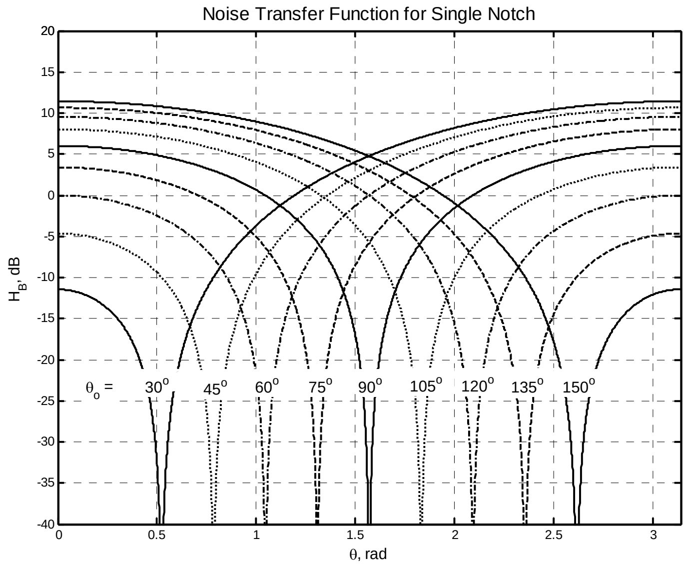


**Figure 5-78** Noise transfer function[54] (5.117) for a range of $\theta_o$ values

### 5.11.2 Fourth-Order Bandstop Noise Shaping

Given the earlier results represented by (5.113) and (5.115), it is relatively easy to extend the noise shaping technique to a wider band of stopband frequencies. In order to do this, assume the DDS output frequency is again represented by the radian angle $\theta_o$ and the stopband nulls are positioned at $\theta_1 = \theta_o - \Delta\theta / 2$ and $\theta_2 = \theta_o + \Delta\theta / 2$. The lower frequency second-order noise shaping function is realized using (5.113) with $a_2 \equiv 1$ and $a_1$ computed with (5.115) using $\theta_1$. A second noise transfer function denoted by

---

[54] MATLAB script u16283_dds_noiseshaping.m.

$$H_n(z) = 1 + b_1 z^{-1} + b_2 z^{-2} \tag{5.122}$$

is similarly computed using $\theta_2$. The cascade of noise transfer functions is then assembled as

$$\begin{aligned} H_{cas}(z) &= \left(1 + a_1 z^{-1} + a_2 z^{-2}\right)\left(1 + b_1 z^{-1} + b_2 z^{-2}\right) \\ &= 1 + c_1 z^{-1} + c_2 z^{-2} + c_3 z^{-3} + c_4 z^{-4} \end{aligned} \tag{5.123}$$

where

$$\begin{aligned} c_1 &= a_1 + b_1 \\ c_2 &= a_2 + b_2 + a_1 b_1 \\ c_3 &= a_1 b_2 + a_2 b_1 \\ c_4 &= a_2 b_2 \end{aligned} \tag{5.124}$$

and

$$\begin{aligned} a_2 &= b_2 \equiv 1 \\ a_1 &= -2\cos\left(\theta_o - \Delta\theta\right) \\ b_1 &= -2\cos\left(\theta_o + \Delta\theta\right) \end{aligned} \tag{5.125}$$

The associated block diagram is given by Figure 5-79 and representative results are shown in Figure 5-80 and Figure 5-81. Selecting the separation parameter $\Delta\theta$ involves a tradeoff between the stopband noise level and the bandwidth of the stopband region. In this first pair of results, $\Delta\theta$ was chosen to correspond to 2.5 MHz whereas in the second pair of results shown in Figure 5-82 and Figure 5-83, the stopband width has been doubled at the expense of the stopband depth.

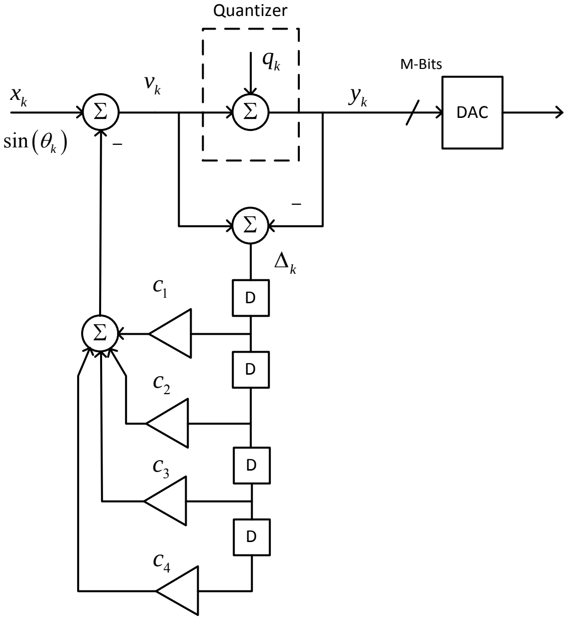


**Figure 5-79** Fourth-order Δ-Σ modulator used to make up for the finite precision of the M-bit DAC

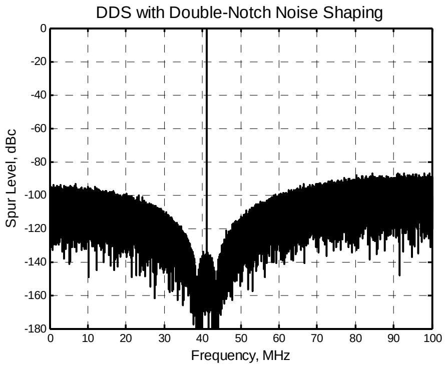


**Figure 5-80** 4th-order Δ-Σ noise shaping result[55] based upon (5.123). 200 MHz clock rate, 41.1 MHz output signal, 9-bit ideal DAC, $\Delta\theta$ corresponded to 2.5 MHz.

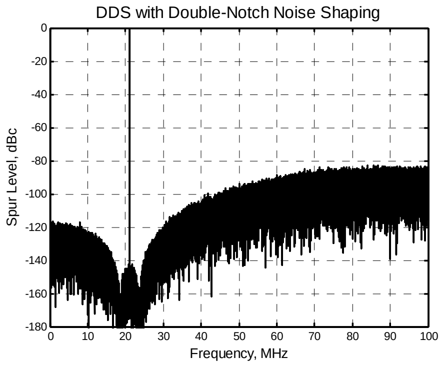


**Figure 5-81** 4th-order Δ-Σ noise shaping result[56] based upon (5.123). 200 MHz clock rate, 21.1 MHz output signal, 9-bit ideal DAC, $\Delta\theta$ corresponded to 2.5 MHz.

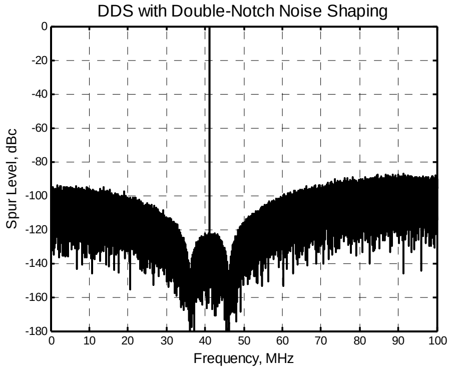


**Figure 5-82** 4th-order Δ-Σ noise shaping result[57] based upon (5.123). 200 MHz clock rate, 41.1 MHz output signal, 9-bit ideal DAC, $\Delta\theta$ corresponded to 5 MHz.

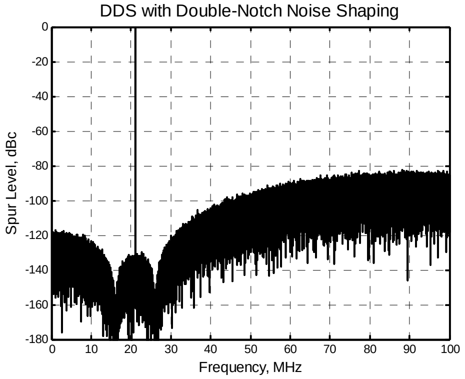


**Figure 5-83** 4th-order Δ-Σ noise shaping result[58] based upon (5.123). 200 MHz clock rate, 21.1 MHz output signal, 9-bit ideal DAC, $\Delta\theta$ corresponded to 5 MHz.

### 5.11.3 Sixth-Order Bandstop Noise Shaping

It is a simple matter to extend the results of the previous sections to a 6th-order noise shaping solution by cascading three second-order sections to form the noise transfer function

$$\begin{aligned} H_n(z) &= \left(1+a_1z^{-1}+a_2z^{-2}\right)\left(1+b_1z^{-1}+b_2z^{-2}\right)\left(1+c_1z^{-1}+c_2z^{-2}\right) \\ &= 1+\sum_{k=1}^{6} e_k z^{-k} \end{aligned} \tag{5.126}$$

In this form, the $e_k$ are given by

---

[55] MATLAB script u16283_dds_noiseshaping.m.

[56] MATLAB script u16283_dds_noiseshaping.m.

[57] MATLAB script u16283_dds_noiseshaping.m.

[58] MATLAB script u16283_dds_noiseshaping.m.

$$\begin{aligned} e_1 &= c_1 + d_1 \\ e_2 &= c_2 + d_2 + c_1 d_1 \\ e_3 &= d_3 + c_1 d_2 + c_2 d_1 \\ e_4 &= d_4 + c_1 d_3 + c_2 d_2 \\ e_5 &= c_1 d_4 + c_2 d_3 \\ e_6 &= c_2 d_4 \end{aligned} \tag{5.127}$$

with

$$\begin{aligned} d_1 &= a_1 + b_1 \\ d_2 &= a_2 + b_2 + a_1 b_1 \\ d_3 &= a_1 b_2 + a_2 b_1 \\ d_4 &= a_2 b_2 \end{aligned} \tag{5.128}$$

and

$$\begin{aligned} a_2 &= b_2 = c_2 \equiv 1 \\ a_1 &= -2\cos(\theta_1) \\ b_1 &= -2\cos(\theta_2) \\ c_1 &= -2\cos(\theta_3) \end{aligned} \tag{5.129}$$

The associated block diagram is just an extension of Figure 5-79 to accommodate the additional two taps.

A representative output spectrum for the $6^{\text{th}}$-order case is shown in Figure 5-84 where the noise transfer function nulls have again been positioned 5 MHz above and 5 MHz below the desired signal at 41.1 MHz. The wider stopband region makes it that much easier for a subsequent PLL or narrow bandpass filter to suppress the shaped noise further away from the desired signal.

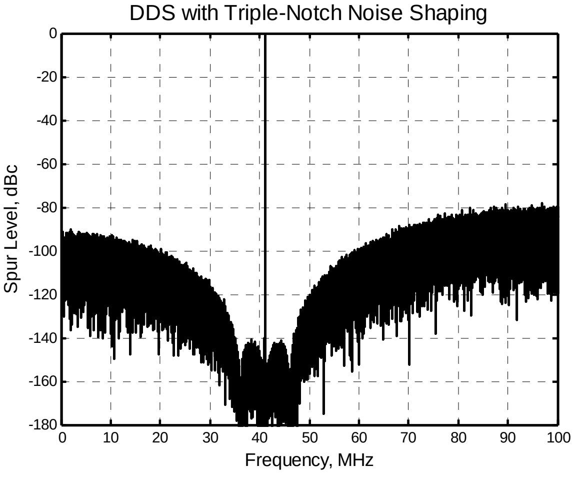


**Figure 5-84** Sixth-order bandstop noise shaping example synthesizing 41.1 MHz with sampling rate = 200 MHz and 9-bit ideal DAC. Design parameters were chosen as $\theta_1 = \theta_o - \Delta\theta$, $\theta_2 = \theta_o$, and $\theta_3 = \theta_o + \Delta\theta$, with $\Delta\theta$ corresponding to 5 MHz.

### 5.11.4 Δ-Σ Summary

A summary of the DDS examples just presented is provided in Figure 5-85. Many variants of the techniques presented in this section are possible. While these techniques can be very effective in suppressing quantization-related spurious signals, they do nothing to suppress spurious signals which arise from DAC-related harmonic distortion. Once these harmonic distortion terms are aliased by the underlying DAC sampling process, they can produce spurious signals that fall within the stopband region. The most effective remedies to harmonic distortion related problems are good frequency planning as discussed in Section 5.4 and using a low-distortion DAC in the first place.

**Summary of Examples: $F_{Clk}$= 200 MHz, 9-Bit Ideal DAC**

2nd-Order Section: 41.1 MHz Output ( Figure 5-74 )
$[ a_1, a_2 ] = ( -0.5519, 1 )$
2nd-Order Section: 21.1 MHz Output ( Figure 5-75 )
$[ a_1, a_2 ] = ( -1.5765, 1 )$
4th-Order Section: 41.1 MHz Output, ±2.5 MHz Spacing ( Figure 5-80 )
$[ a_1, a_2 ] = ( -0.7011, 1 )$
$[ b_1, b_2 ] = ( -0.3994, 1 )$
$[ c_1, c_2, c_3, c_4 ] = ( -1.1005, 2.2800, -1.1005, 1)$
4th-Order Section: 21.1 MHz Output, ±2.5 MHz Spacing ( Figure 5-81 )
$[ a_1, a_2 ] = (-1.6682, 1 )$
$[ b_1, b_2 ] = (-1.4750, 1 )$
$[ c_1, c_2, c_3, c_4 ] = (-3.1432, 4.4606, -3.1432, 1)$
4th-Order Section: 41.1 MHz Output, ±5 MHz Spacing ( Figure 5-82 )
$[ a_1, a_2 ] = (-0.8459, 1 )$
$[ b_1, b_2 ] = (-0.2444, 1 )$
$[ c_1, c_2, c_3, c_4 ] = (-1.0903, 2.2068, -1.0903, 1 )$
4th-Order Section: 21.1 MHz Output, ±5 MHz Spacing ( Figure 5-83 )
$[ a_1, a_2 ] = (-1.7496, 1 )$
$[ b_1, b_2 ] = (-1.3645, 1 )$
$[ c_1, c_2, c_3, c_4 ] = (-3.1141, 4.3873, -3.1141, 1 )$
6th-Order Section: 41.1 MHz Output, ±5 MHz Spacing ( Figure 5-84 )
$[ a_1, a_2 ] = (-0.8459, 1 )$
$[ b_1, b_2 ] = (-0.2444, 1 )$
$[ c_1, c_2 ] = (-0.5519, 1 )$
$[ e_1, e_2, e_3, e_4, e_5, e_6 ] = (-1.6422, 3.8085, -3.3986, 3.8085, -1.6422, 1 )$

**Figure 5-85** Summary of Δ-Σ examples presented in this section

### 5.12 Hybrid PLL-DDS Systems

Gilmore received a patent[59] for a PLL driven by a DDS in 1990 [31]. The original configuration is shown in Figure 5-86. The benefits of using a variable modulus divide-by-M block as discussed in Section 5.9 to avoid DAC-related harmonic distortion spurs were not realized until

---

[59] A subsequent related patent was also awarded to Gilmore in 1992 [31], [32].

some time later. The PLL in Figure 5-86 made it possible to synthesize relatively clean RF signals at frequencies well beyond the DDS capabilities of the time.

The DDS / PLL configuration shown in Figure 5-86 is, however, hampered by the 20 $\text{Log}_{10}( N / M )$ dB increase in DDS-related spurious levels at the PLL output. One alternative to circumvent this issue is to use a dual-channel DDS to create a fine-resolution frequency offset which is used with a single-sideband mixing operation as shown in Figure 5-87. The increase in circuit complexity is, however, substantial. Careful frequency planning is required in this configuration to ensure that leakage of the VCO signal or the unwanted sideband mixing product into the divide-by-N block input does not result in close-in spurious terms due to sampling effects associated with the divider.

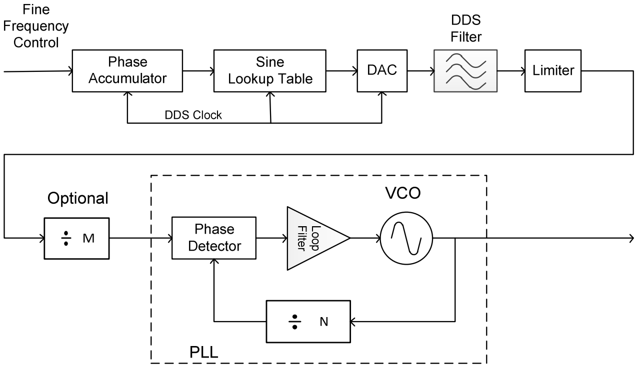


**Figure 5-86** DDS-PLL hybrid configuration patented by Gilmore [31]

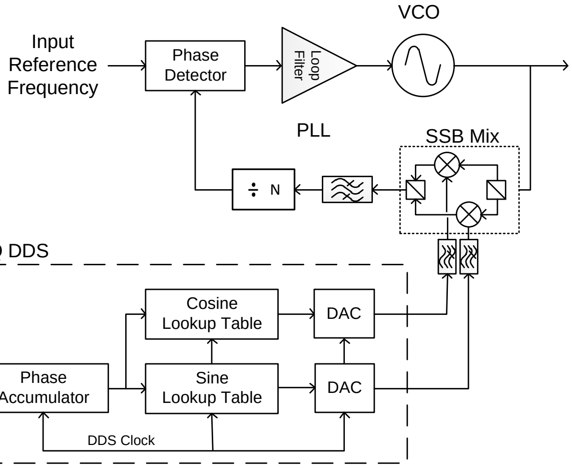


**Figure 5-87** Alternative DDS-PLL hybrid that does not suffer from 20 $\text{Log}_{10}( N )$ spurious enhancement but does involve considerably more complexity than Figure 5-86

A simplified variant of the method shown in Figure 5-87 can prove useful for navigating around troublesome *integer-boundary spurs* which can occur in fractional-N PLLs. The only other way to avoid these spurs is to have more than one reference frequency available to the fractional-N PLL and appropriately select between them. The boundary spur problem is addressed more completely in Section 12.3.

## 5.13 References

1. Tierney, J., C. Rader, and B. Gold, "A Digital Frequency Synthesizer," *IEEE Trans. Audio Electoacoustic*, vol. 18, no. 1, pp. 48-57, Mar. 1971.

2. Gorski-Popiel, J., *Frequency Synthesis, Techniques and Applications*, IEEE Press, New York, 1975.

3. Kroupa, V.F., *Direct Digital Frequency Synthesizers*, IEEE Press, 1999, New York.

4. Buchanan, David, "Choosing DACs for Direct Digital Synthesis," Analog Devices Application Note AN-237.

5. Huang, C., L. Ren, and Er-ke Mao, "A Systematic Frequency Planning Method in Direct Digital Synthesizer (DDS) Design," IEEE International Conference on Wireless Communications and Signal Processing, 2009.

6. Crawford, J.A., *Advanced Phase-Lock Techniques*, Boston, MA: Artech House, 2008.

7. ____________, *Frequency Synthesizer Design Handbook*, Boston, MA: Artech House, 1994.

8. "A Technical Tutorial on Digital Signal Synthesis," Analog Devices, 1999.

9. Nicholas, H., and H. Samueli, "An Analysis of the Output Spectrum of Direct Digital Frequency Synthesizers in the Presence of Phase-Accumulator Truncation," *41st Annual Frequency Control Symposium*, 1987.

10. Kroupa, V.F., "Spurious Signals in Direct Digital Frequency Synthesizers Due to the Phase Truncation," *1999 Joint Meeting EFTF*.

11. Nicholas, H., H. Samueli, and B. Kim, "The Optimization of Direct Digital Frequency Synthesizer Performance in the Presence of Finite Word Length Effects," *42nd Annual Frequency Control Symposium*, 1988.

12. Cheng, S., et al., "Further Enhancements to the Analysis of Spectral Purity in the Application of Practical Direct Digital Synthesis," 2004 IEEE International Ultrasonics, Ferroelectrics, and Frequency Control Joint 50th Anniversary Conference.

13. Torosyan, A., and A.N. Wilson, "Exact Analysis of DDS Spurs and SNR due to Phase Truncation and Arbitrary Phase-to-Amplitude Errors," *Proc. 2005 IEEE International Frequency Control Symposium,* 2005.

14. Curticapean, F., and J. Niittylahti, "Exact Analysis of Spurious Signals in Direct Digital Frequency Synthesizers Due to Phase Truncation," *Electronic Letters*, 20 March 2003.

15. Li, Zhengming., et al., "Complete Analysis of the Spurious Spectrum of DDS in the Presence of Phase Truncation," *IEEE International Conf. Service Operations, Logistics, and Informatics*, 2009.

16. Torosyan, A., “Direct Digital Frequency Synthesizers: Complete Analysis and Design Guidelines,” Ph.D. Dissertation, University of California, Los Angeles, 2003.

17. Essenwanger, K., and V.S. Reinhardt, “Sine Output DDSs A Survey of the State of the Art,” *1998 IEEE International Frequency Control Symposium*.

18. Mehrgardt, S., “Noise Spectra of Digital Sine-Generators Using the Table-Lookup Method,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, Aug. 1983.

19. Volder, Jack, “The CORDIC Trigonometric Computing Technique,” *IRE Trans. Electronic Computers*, Sept. 1959.

20. Andraka, Ray, “A Survey of CORDIC Algorithms for FPGA Based Computers,” *Proc. 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays*, Feb. 22-24, 1998, Monterey, CA.

21. Walther, J.S., “A Unified Algorithm for Elementary Functions,” *Proc. of Spring Joint Computer Conference*, American Federation of Information Processing Socieities, 1971.

22. Cavone, G., A. Di Nisio, et al., “A Maximum Likelihood Estimator for ADC and DAC Linearity Testing,” 13th Workshop on ADC Modeling and Testing, Florence, Italy, Sept. 2008.

23. Maeda, A., “A Method to Generate a Very Low Distortion High Frequency Sine Waveform Using an AWG,” *International Test Conference*, 2008.

24. Zimmerman, G., and M. Flanagan, “Spur-Reduced Numerically-Controlled Oscillator for Digital Receivers,” *IEEE Asilomar Conference 1992*.

25. Vankka, J., and K.A. Halonen, *Direct Digital Synthesizers: Theory, Design and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.

26. Vankka, J., “Spur Reduction Techniques in Sine Output Direct Digital Synthesis,” *1996 IEEE International Frequency Control Symposium*, 1996.

27. Cordesses, L., “Direct Digital Synthesis: A Tool for Periodic Wave Generation (Part I),” *IEEE Signal Processing Magazine*, July 2004.

28. ___________, “Direct Digital Synthesis: A Tool for Periodic Wave Generation (Part II),” *IEEE Signal Processing Magazine*, Sept. 2004.

29. O’Leary, P., and F. Maloberti, “A Direct Digital Synthesizer with Improved Spectral Performance,” *IEEE Trans. Comm.*, Vol. COM-39, July 1991, pp. 1046-1048.

30. Candy, J.A., and A.N. Huynh, “Double Interpolation for Digital to Analog Conversion,” *IEEE Trans Comm.*, Vol. COM-34, January 1984, pp. 77-81.

31. Gilmore, R.P., “Direct Digital Synthesizer Driven Phase Lock Loop Frequency Synthesizer,” US patent 4,965,533, 23 Oct. 1990.

32. ___________, "Direct Digital Synthesizer Driven Phase Lock Loop Frequency Synthesizer with Hard Limiter," US Patent 5,028,887, 2 July 1991.

## Selected Bibliography

Khilar, S., et al., "Design and Analysis of Direct Digital Frequency Synthesizer," *First International Conference on Emerging Trends in Engineering and Technology*, 2008.

Maeda, Akinori, "A Method to Generate a Very Low Distortion High-Frequency Sine Waveform Using an AWG," *International Test Conference*, 2008.

Matic, V., et al., "Comparison of Digital Signal Processing Methods for Sine Wave Signal Generation," *Telecommunications Symposium*, *ITS '98 Proceedings,* 1998.

Olsen, James, and Philip Fishman, "Truncation Effects in Direct Digital Frequency Synthesis," MIT Lincoln Laboratory, 1987.

Papay, Z., "Numerical Distortion in Single-Tone DDS," *IEEE Instrumentation and Measurement Technology Conference*, 2001.

Popek, G., and M. Kampik, "Low-Spur Numerically Controlled Oscillator Using Taylor Series Approximation," *XI International PhD Workshop*, OWD, 2009.

Reinhardt, V.S., "Spur Reduction Techniques in Direct Digital Synthesizers," *Proc. 1993 International Frequency Control Symposium*, 1993.

## 5.14 Appendix 5-1: Noise Spectrum Associated with Random Phase Truncation Errors

The phase errors associated with phase truncation are generally periodic in nature as discussed at length in Section 5.5.2. If, however, the phase truncation errors are truly random, the simplified analysis discussed in this section can be applied.

If an *average noise floor* is computed based upon the phase truncation errors being uncorrelated as in the following section, it is quickly seen that the discrete spurs completely dwarf the computed *average noise floor* level. In Figure 5-17 and Figure 5-18, for example, the so-called average noise floor associated with phase truncation is at about –135 dBc which is completely masked by the high spurious levels present. A simple derivation for the average noise floor associated with phase truncation, assuming that the error terms are random, is given in the section to substantiate this important point.

Consider a continuous-time signal that is given by

$$r(t)=\sin\left[\omega_o t+\theta_n(t)\right] \tag{5.130}$$

This can be easily expanded as

$$r(t)=\sin(\omega_o t)\cos\left[\theta_n(t)\right]+\cos(\omega_o t)\sin\left[\theta_n(t)\right] \tag{5.131}$$

Assuming that $|\theta_n(t)| << 1$ radian, (5.131) can be closely approximated by

$$r(t)\approx\sin(\omega_o t)+\theta_n(t)\cos(\omega_o t) \tag{5.132}$$

Assume now that the phase term is a simple sinusoidal function with a peak phase deviation of $\theta_p$ radians and radian frequency of $\omega_m$. This assumption makes it easy to rewrite (5.132) as

$$r(t)\approx\sin(\omega_o t)+\frac{\theta_p}{2}\left\{\sin\left[(\omega_o+\omega_m)t\right]-\sin\left[(\omega_o-\omega_m)t\right]\right\} \tag{5.133}$$

The relative level of each sideband term at ± $\omega_m$ rad/sec offset is clearly

$$L_{Side}\approx 20\log_{10}\left(\frac{\theta_p}{2}\right)\text{ dBc} \tag{5.134}$$

Taking this result one step further, if $\theta_{RMS}$ is the RMS phase noise (in radians) in a 1 Hertz bandwidth at the same offset frequencies as represented by (5.133), the corresponding phase noise level is

$$L_{phs}\approx 20\log_{10}\left(\frac{\theta_{RMS}}{\sqrt{2}}\right)\quad\text{dBc/Hz} \tag{5.135}$$

Assume now that $\theta_n(t)$ in (5.130) represents the phase quantization noise present in an NCO due to finite phase truncation. The discrete-time output of the NCO can be written as

$$r_k=\sin\left[\omega_o kT_{Clock}+\theta_n(k)\right] \tag{5.136}$$

Assume that the phase quantization phase noise is uniformly distributed across one LSB [ –Δ / 2, Δ / 2 ] and that its corresponding variance is given by

$$\sigma_\theta^2 = \frac{\Delta^2}{12} \text{ rad}^2 \tag{5.137}$$

The corresponding one-sided power spectral density for $\theta_n$ is uniformly spread from 0 Hz to $F_{Clock}$ / 2 Hz with a level of

$$L_q = \frac{\Delta^2}{12} \frac{2}{F_{Clock}} \text{ rad}^2 / \text{Hz} \tag{5.138}$$

Assume that the $2^W$ available phase steps are uniformly spread across the range [0, 2π) in which case

$$\Delta = 2\pi\, 2^{-W} = \pi 2^{-W+1} \text{ rad} \tag{5.139}$$

The power spectral density of $\theta_n$ is then given by

$$P_\theta = \left( \frac{2\pi}{2^W} \right)^2 \frac{1}{6} = \frac{\pi^2}{3F_{Clock}} 2^{-2W+1} \quad \frac{\text{rad}^2}{\text{Hz}} \tag{5.140}$$

Based upon the earlier result given by (5.135), the resultant quantization-related phase noise floor of (5.136) is given by

$$\begin{aligned} \mathcal{L} &= 10\log_{10}\left[ \frac{\pi^2}{3F_{Clock}} 2^{-2W} \right] \\ &= -6.02W + 10\log_{10}\left( \frac{\pi^2}{3F_{Clock}} \right) \\ &= -6.02W + 5.17 - 10\log_{10}\left( F_{Clock} \right) \quad \frac{\text{dBc}}{\text{Hz}} \end{aligned} \tag{5.141}$$

This result is only meaningful if the phase truncation errors are truly random. Because of the cyclical nature of the error process, however, this is not the case. A close-up of the noise spectrum shown in Figure 5-17 reveals that the low-level spectral terms are *all* discrete line-spectrum terms as shown in Figure 5-88. It is therefore somewhat meaningless to talk about an *average noise floor* due to phase truncation because the spectrum is always a discrete-line spectrum.

Some applications will be indifferent about whether the spectrum floor is a continuous noise spectrum or actually composed of discrete spectral lines whereas others may not be. Carrying more bits of precision will reduce the amplitude of the discrete lines of course, but not the discrete line nature of the spectrum. Phase dithering as discussed in Section 5.10.2 and noise-shaping as discussed in Section 5.11 are very effective methods for eliminating low-level discrete spurious lines like these in the output spectrum.

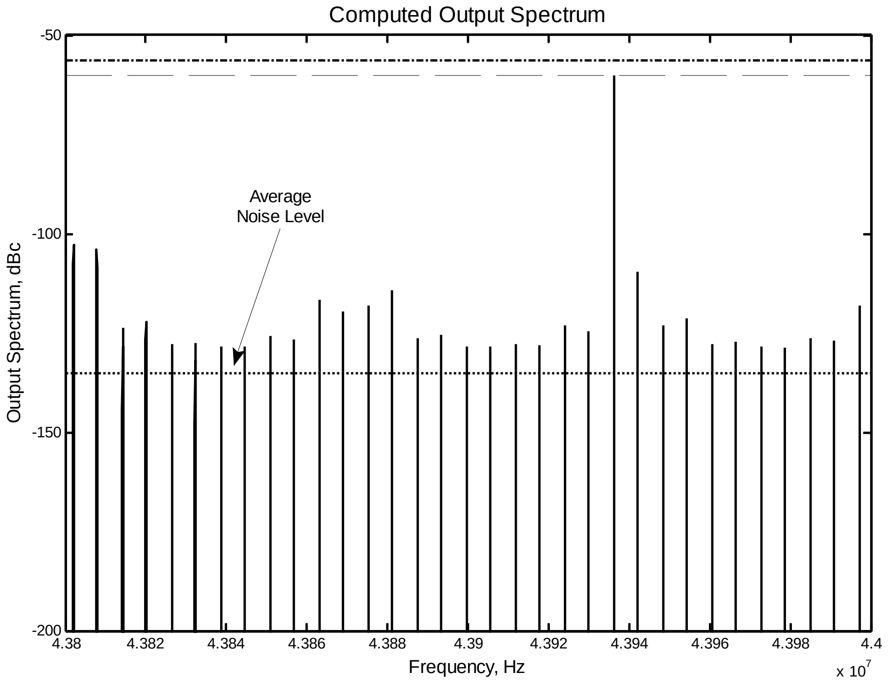
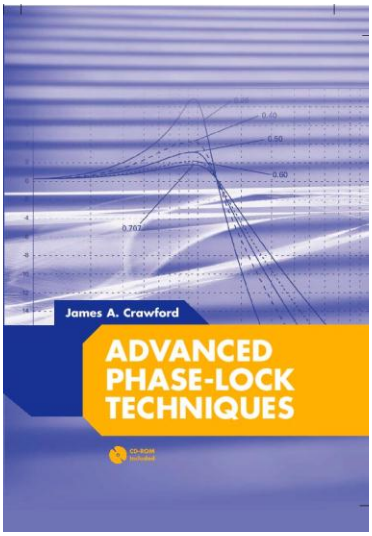


**Figure 5-88** Close-up of Figure 5-17 reveals that the low-level spectrum is composed of a comb of spurious terms rather than a uniform flat noise spectrum. The *average noise level* is based upon (5.141).



# Advanced Phase-Lock Techniques

James A. Crawford

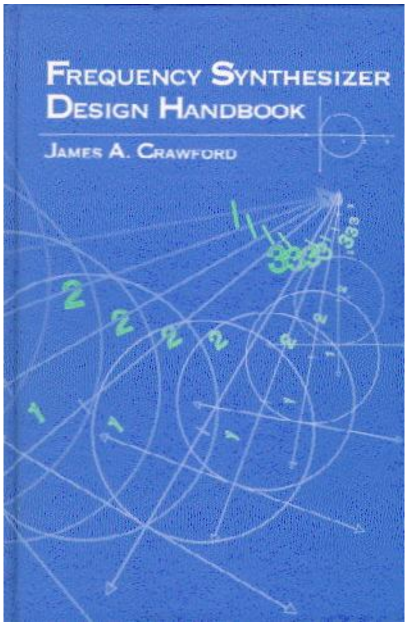
2008

Artech House

510 pages, 480 figures, 1200 equations
CD-ROM with all MATLAB scripts

ISBN-13: 978-1-59693-140-4
ISBN-10: 1-59693-140-X

| Chapter | Brief Description | *Pages* |
|---|---|---|
| 1 | *Phase-Locked Systems—A High-Level Perspective*<br>An expansive, multi-disciplined view of the PLL, its history, and its wide application. | 26 |
| 2 | *Design Notes*<br>A compilation of design notes and formulas that are developed in details separately in the text. Includes an exhaustive list of closed-form results for the classic type-2 PLL, many of which have not been published before. | 44 |
| 3 | *Fundamental Limits*<br>A detailed discussion of the many fundamental limits that PLL designers may have to be attentive to or else never achieve their lofty performance objectives, e.g., Paley-Wiener Criterion, Poisson Sum, Time-Bandwidth Product. | 38 |
| 4 | *Noise in PLL-Based Systems*<br>An extensive look at noise, its sources, and its modeling in PLL systems. Includes special attention to 1/*f* noise, and the creation of custom noise sources that exhibit specific power spectral densities. | 66 |
| 5 | *System Performance*<br>A detailed look at phase noise and clock-jitter, and their effects on system performance. Attention given to transmitters, receivers, and specific signaling waveforms like OFDM, M-QAM, M-PSK. Relationships between EVM and image suppression are presented for the first time. The effect of phase noise on channel capacity and channel cutoff rate are also developed. | 48 |
| 6 | *Fundamental Concepts for Continuous-Time Systems*<br>A thorough examination of the classical continuous-time PLL up through 4th-order. The powerful Haggai constant phase-margin architecture is presented along with the type-3 PLL. Pseudo-continuous PLL systems (the most common PLL type in use today) are examined rigorously. Transient response calculation methods, 9 in total, are discussed in detail. | 71 |
| 7 | *Fundamental Concepts for Sampled-Data Control Systems*<br>A thorough discussion of sampling effects in continuous-time systems is developed in terms of the *z*-transform, and closed-form results given through 4th-order. | 32 |
| 8 | *Fractional-N Frequency Synthesizers*<br>A historic look at the fractional-N frequency synthesis method based on the U.S. patent record is first presented, followed by a thorough treatment of the concept based on $\Delta$-$\Sigma$ methods. | 54 |
| 9 | *Oscillators*<br>An exhaustive look at oscillator fundamentals, configurations, and their use in PLL systems. | 62 |
| 10 | *Clock and Data Recovery*<br>Bit synchronization and clock recovery are developed in rigorous terms and compared to the theoretical performance attainable as dictated by the Cramer-Rao bound. | 52 |



# *Frequency Synthesizer Design Handbook*

James A. Crawford

1994

Artech House

435 pages, 265 figures, 620 equations

ISBN 0-89006-440-7

| Chapter | Brief Description | *Pages* |
|---|---|---|
| 1 | *Introduction to Computer-Aided Frequency Synthesizer Design* | 2 |
| 2 | *Building Blocks for Frequency Synthesis Using Phase-Locked Loops* | 38 |
| 3 | *Phase Noise and its Impact on System Performance* | 106 |
| 4 | *Phase-Locked Loop Analysis for Continuous Linear Systems* | 55 |
| 5 | *Frequency Synthesis Using Sampled-Data Control Systems* | 43 |
| 6 | *Fast-Switching Frequency Synthesizer Design Considerations* | 40 |
| 7 | *Hybrid Phase-Locked Loops* | 66 |
| 8 | *MACSET: A Computer Program for the Design and Analysis of Phase-Locked Loop Frequency Synthesizers* | 33 |
| 9 | *Fractional-N Frequency Synthesis* | 34 |