A Technical Tutorial

on Digital Signal Synthesis

Outline

Section 1. Fundamentals of DDS technology

Theory of operation
Circuit architecture
Tuning equation
Elements of DDS circuit functionality and capabilities
DAC integration
Trends in functional integration

Section 2. Understanding the Sampled Output of a DDS Output

Implications of the Nyquist Theorum
Aliased images in the output
Source of aliased images
Calculating the occurrence of aliased images
Quantization considerations
Sin(X)/X response
AC and DC linearity of the output

Section 3. Frequency/phase-hopping Capability of DDS

Calculating the output tuning word
Determining maximum tuning resolution
Determining maximum tuning speed
Understanding the DDS control interface
Pre-programming profile registers

Section 4. The DDS Output Spectrum

The effect of DAC resolution on spurious performance
The effect of oversampling on spurious performance
The effect of truncating the phase accumulator on spurious performance
Additional DDS Spur sources
Wideband spur performance
Narrowband spur performance
Predicting and exploiting spur "sweet spots" in a DDS' tuning range
Jitter and phase noise considerations in a DDS system
Output filtering considerations

Section 5. High-speed Reference Clock Considerations

Implications of jitter and phase noise in the reference clock
Reference clock multipliers
SFDR performance vs. the REFCLK Multiplier function

Section 6. Interfacing to the DDS Output

Output power considerations
FS output current range and tradeoffs vs. spur performance
Single-ended vs. differential DAC output
Driving an output amplifier

Section 7. DDS as a Clock Generator

Definition of clock generator application for a DDS
Squaring the DDS output with an LP filter and comparator
Managing jitter in the clock generator application

Section 8. Replacing/Integrating a PLL with a DDS Solution

Traditional analog synthesizer vs. the DDS implementation
How DDS can eliminate analog upconverter stages
Example of implementation of DDS as an LO

Section 9. Digital Modulator Application of DDS

Basic digital modulator theory
System architecture and requirements
Digital filters
Multirate DSP
Clock and input data synchronization considerations
Data encoding methodologies and DDS implementations

Section 10. Using Aliased Images to Generate Nyquist + Frequencies from a DDS

Creating and isolating aliased images in the DDS output spectrum
SFDR performance expectations of the aliased image
Amplitude prediction of the aliased image
Frequency hopping considerations in the aliased image application

Section 11. Ancillary DDS Techniques, Features, and Functions

Improving SFDR with the addition of phase dither in the phase accumulator
Understanding DDS frequency “chirp” functionality
Achieving output amplitude control/modulation within a DDS device
Synchronization multiple DDS devices

Section 12. Techniques for Bench Evaluation of a DDS Solution

PC-based evaluation platforms and reference designs

Section 1. Fundamentals of DDS Technology

Overview

Direct digital synthesis (DDS) is a technique for using digital data processing blocks as a means to generate a frequency- and phase-tunable output signal referenced to a fixed-frequency precision clock source. In essence, the reference clock frequency is "divided down" in a DDS architecture by the scaling factor set forth in a programmable binary tuning word. The tuning word is typically 24-48 bits long which enables a DDS implementation to provide superior output frequency tuning resolution.

Today's cost-competitive, high-performance, functionally-integrated, and small package-sized DDS products are fast becoming an alternative to traditional frequency-agile analog synthesizer solutions. The integration of a high-speed, high-performance, D/A converter and DDS architecture onto a single chip (forming what is commonly known as a *Complete-DDS* solution) enabled this technology to target a wider range of applications and provide, in many cases, an attractive alternative to analog-based PLL synthesizers. For many applications, the DDS solution holds some distinct advantages over the equivalent agile analog frequency synthesizer employing PLL circuitry.

DDS advantages:

- Micro-Hertz tuning resolution of the output frequency and sub-degree phase tuning capability, all under complete digital control.
- Extremely fast "hopping speed" in tuning output frequency (or phase), phase-continuous frequency hops with no over/undershoot or analog-related loop settling time anomalies.
- The DDS digital architecture eliminates the need for the manual system tuning and tweaking associated with component aging and temperature drift in analog synthesizer solutions.
- The digital control interface of the DDS architecture facilitates an environment where systems can be remotely controlled, and minutely optimized, under processor control.
- When utilized as a quadrature synthesizer, DDS afford unparalleled matching and control of I and Q synthesized outputs.

Theory of Operation

In its simplest form, a direct digital synthesizer can be implemented from a precision reference clock, an address counter, a programmable read only memory (PROM), and a D/A converter (see Figure 1-1).

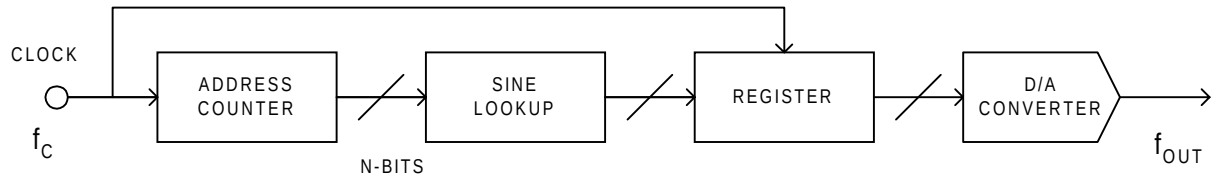


Figure 1-1. Simple Direct Digital Synthesizer

In this case, the digital amplitude information that corresponds to a complete cycle of a sinewave is stored in the PROM. The PROM is therefore functioning as a sine lookup table. The address counter steps through and accesses each of the PROM's memory locations and the contents (the equivalent sine amplitude words) are presented to a high-speed D/A converter. The D/A converter generates an analog sinewave in response to the digital input words from the PROM. The output frequency of this DDS implementation is dependent on 1.) the frequency of the reference clock, and 2.) the sinewave step size that is programmed into the PROM. While the analog output fidelity, jitter, and AC performance of this simplistic architecture can be quite good, it lacks tuning flexibility. The output frequency can only be changed by changing the frequency of the reference clock or by reprogramming the PROM. Neither of these options support high-speed output frequency hopping.

With the introduction of a phase accumulator function into the digital signal chain, this architecture becomes a numerically-controlled oscillator which is the core of a highly-flexible DDS device. As figure 1-2 shows, an N-bit variable-modulus counter and phase

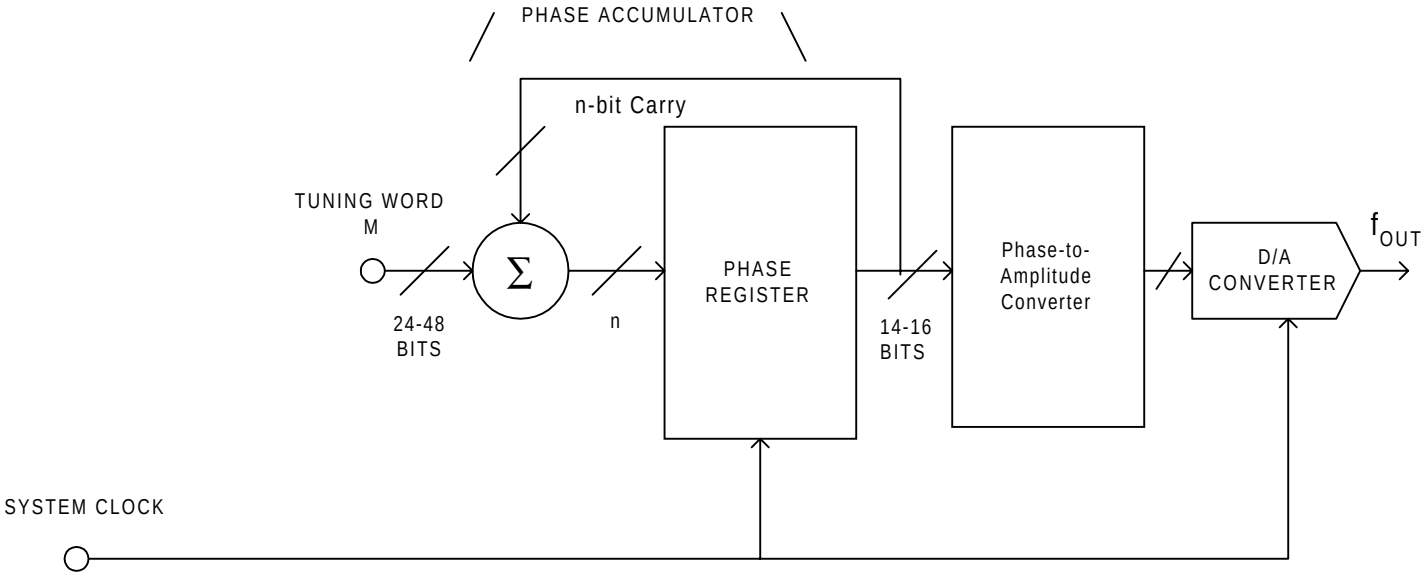
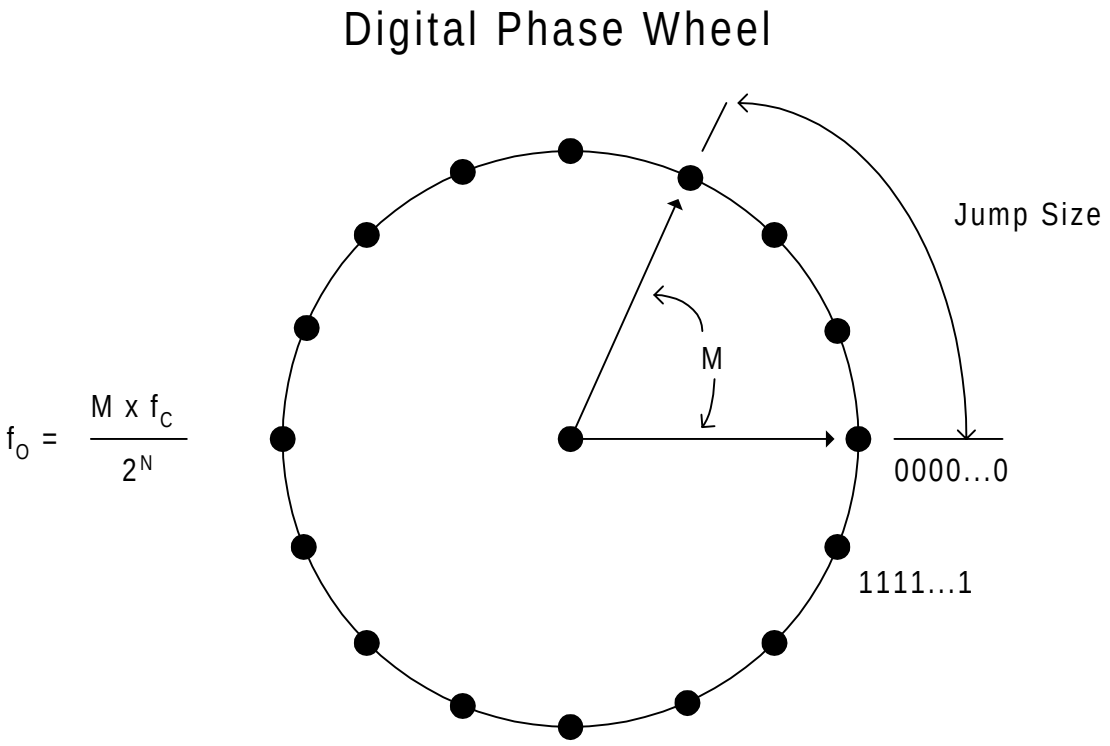


Figure 1-2. Frequency-tunable DDS System

register are implemented in the circuit before the sine lookup table, as a replacement for the address counter. The carry function allows this function as a "phase wheel" in the DDS architecture. To understand this basic function, visualize the sinewave oscillation as a vector

rotating around a phase circle (see Figure 1-3). Each designated point on the phase wheel corresponds to the equivalent point on a



n	NUMBER OF POINTS
8	256
12	4096
16	65535
20	1048576
24	16777216
28	268435456
32	4294967296
48	281474976710656

Figure 1-3. Digital Phase Wheel

cycle of a sine waveform. As the vector rotates around the wheel, visualize that a corresponding output sinewave is being generated. One revolution of the vector around the phase wheel, at a constant speed, results in one complete cycle of the output sinewave. The phase accumulator is utilized to provide the equivalent of the vector's linear rotation around the phase wheel. The contents of the phase accumulator correspond to the points on the cycle of the output sinewave. The number of discrete phase points contained in the "wheel" is determined by the resolution, N, of the phase accumulator. The output of the phase accumulator is linear and cannot directly be

used to generate a sinewave or any other waveform except a ramp. Therefore, a phase-to-amplitude lookup table is used to convert a truncated version of the phase accumulator's instantaneous output value into the sinewave amplitude information that is presented to the D/A converter. Most DDS architectures exploit the symmetrical nature of a sinewave and utilize mapping logic to synthesize a complete sinewave cycle from ¼ cycle of data from the phase accumulator. The phase-to-amplitude lookup table generates all the necessary data by reading forward then back through the lookup table.

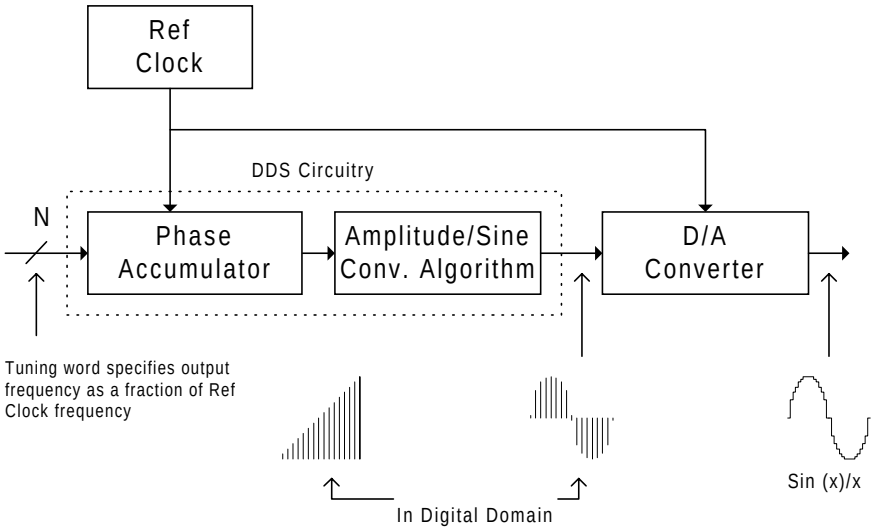


Figure 1-4. Signal flow through the DDS architecture

The phase accumulator is actually a modulus M counter that increments its stored number each time it receives a clock pulse. The magnitude of the increment is determined by a digital word M contained in a "delta phase register" that is summed with the overflow of the counter. The word in the delta phase register forms the phase step size between reference clock updates; it effectively sets how many points to skip around the phase wheel. The larger the jump size, the faster the phase accumulator overflows and completes its equivalent of a sinewave cycle. For a N=32-bit phase accumulator, an M value of 0000…0001(one) would result in the phase accumulator overflowing after 2^{32} reference clock cycles (increments). If the M value is changed to 0111…1111, the phase accumulator will overflow after only 2^{1} clock cycles, or two reference clock cycles. This control of the jump size constitutes the frequency tuning resolution of the DDS architecture.

The relationship of the phase accumulator and delta phase accumulator form the basic tuning equation for DDS architecture:

$$F_{OUT} = (M\ (REFCLK))\ /2^{N}$$

Where: F_{OUT} = the output frequency of the DDS
M = the binary tuning word
REFCLK = the internal reference clock frequency (system clock)
N = The length in bits of the phase accumulator

Changes to the value of M in the DDS architecture result in immediate and phase-continuous changes in the output frequency. In practical application, the M value, or frequency tuning word, is loaded into an internal serial or byte-loaded register which precedes the parallel-output delta phase register. This is generally done to minimize the package pin count of the DDS device. Once the buffer register is loaded, the parallel-output delta phase register is clocked and the DDS output frequency changes. Generally, the only speed limitation to changing the output frequency of a DDS is the maximum rate at which the buffer register can be loaded and executed. Obviously, a parallel byte load control interface enhances frequency hopping capability.

Trends in Functional Integration

One of the advantages to the digital nature of DDS architecture is that digital functional blocks can readily be added to the core blocks to enhance the capability and feature set of a given device. For general purpose use, a DDS device will include an integrated D/A converter function to provide an analog output signal. This “complete-DDS” approach greatly enhances the overall usefulness and “user-friendliness” associated with the basic DDS devices. DDS devices are readily available with integrated 10-bit D/A converters supporting internal REFCLK speeds to 180 MHz. The present state of the art for a complete-DDS solution is at 300 MHz clock speeds with an integrated 12-bit D/A converter.

Along with the integrated D/A converter, DDS solutions normally contain additional digital blocks that perform various operations on the signal path. These blocks provide a higher level of functionality in the DDS solution and provide an expanded set of user-controlled features. The block diagram of an expanded-feature DDS device is shown in Figure 1-5.

The individual functional blocks are described below:

- (A) A programmable REFCLK Multiplier function include at the clock input, multiplies the frequency of the external reference clock, thereby reducing the speed requirement on the precision reference clock. The REFCLK Multiplier function also enhances the ability of the DDS device to utilize available system clock sources.

- (B) The addition of an adder after the phase accumulator enables the output sinewave to be phase-delayed in correspondence with a phase tuning word. The length of the adder circuit determines the number of bits in the phase tuning word, and therefore, the resolution of the delay. In this architecture, the phase tuning word is 14-bits.

- (C) An Inverse SINC block inserted before the D/A converter compensates for the SIN(X)/X response of the quantized D/A converter output, and thereby provides a constant amplitude output over the Nyquist range of the DDS device

- (D) A digital multiplier inserted between the Sine look-up table and the D/A converter enables amplitude modulation of the output sinewave. The width of the digital multiplier word determines the resolution of the output amplitude step size.

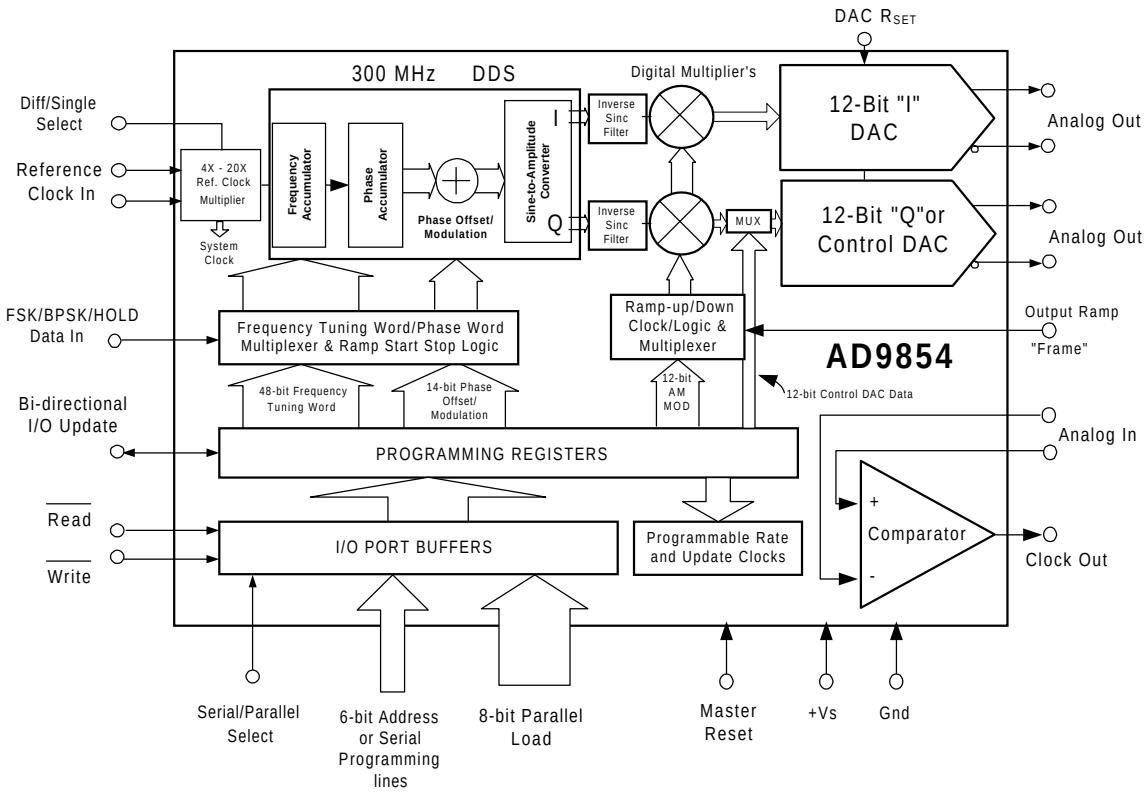


Figure 1-5. Full-featured 12-bit/300 MHz DDS Architecture

- (E) An additional high-speed D/A converter can be included to provide the cosine output from the DDS. This allows the DDS device to provide I and Q outputs which are precisely matched in frequency, quadrature phase, and amplitude. The additional D/A converter may also be driven from the control interface and used as a control DAC for various applications.

- (F) A high-speed comparator function can be integrated which facilitates use of the DDS device as a clock generator. The comparator is configured to convert the sinewave output from the DDS D/A converter into a square wave.

- (G) Frequency/phase registers can be added which allow frequency and phase words to be pre-programmed and their contents executed via a single control pin. This configuration also supports frequency-shift keying (FSK) modulation with the single-pin input programmed for the desired “mark” and “space” frequencies.

DDS devices are available that incorporate all of this functionality (and more) and support internal clock rates up to 300 MHz. The growing popularity in DDS solutions is due to the fact that all of this performance and functionality is available at a reasonable price and in a comparatively small package.

The following is a general guideline for the level of performance available from the dual 12-bit/300 MHz complete-DDS solution described in Figure 1-4. (Conditions assume 30 MHz external reference clock multiplied internally by 10 to yield an internal clock rate of 300 MHz):

-Frequency tuning word length = 48 bits which gives an output frequency tuning resolution of 1 μHz.

-Phase tuning word length = 14 bits which provides .022 degrees of phase delay control resolution.

-REFCLK Multiplier range = programmable in integer increments over the range of 4× to 20×

-Output frequency bandwidth (assuming one-third of REFCLK rate) = 100 MHz

-Frequency tuning rate = 100 MHz with 8-bit byte parallel load

-Output amplitude control = zero output to fullscale in 8128 steps (12-bit control word)

-Output spurious performance = 50 dB worst case wideband spurs at 80 MHz output.

-I/Q output matching = .01 Degree

-Output flatness DC to Nyquist = .01 dB

Section 2. Understanding the Sampled Output of a DDS Device

An understanding of sampling theory is necessary when analyzing the sampled output of a DDS-based signal synthesis solution. The spectrum of a sampled output is illustrated in Figure 2-1. In this example, the sampling clock (f_{CLOCK}) is 300 MHz and the fundamental output frequency ($f_{OUT)}$ is 80 MHz.

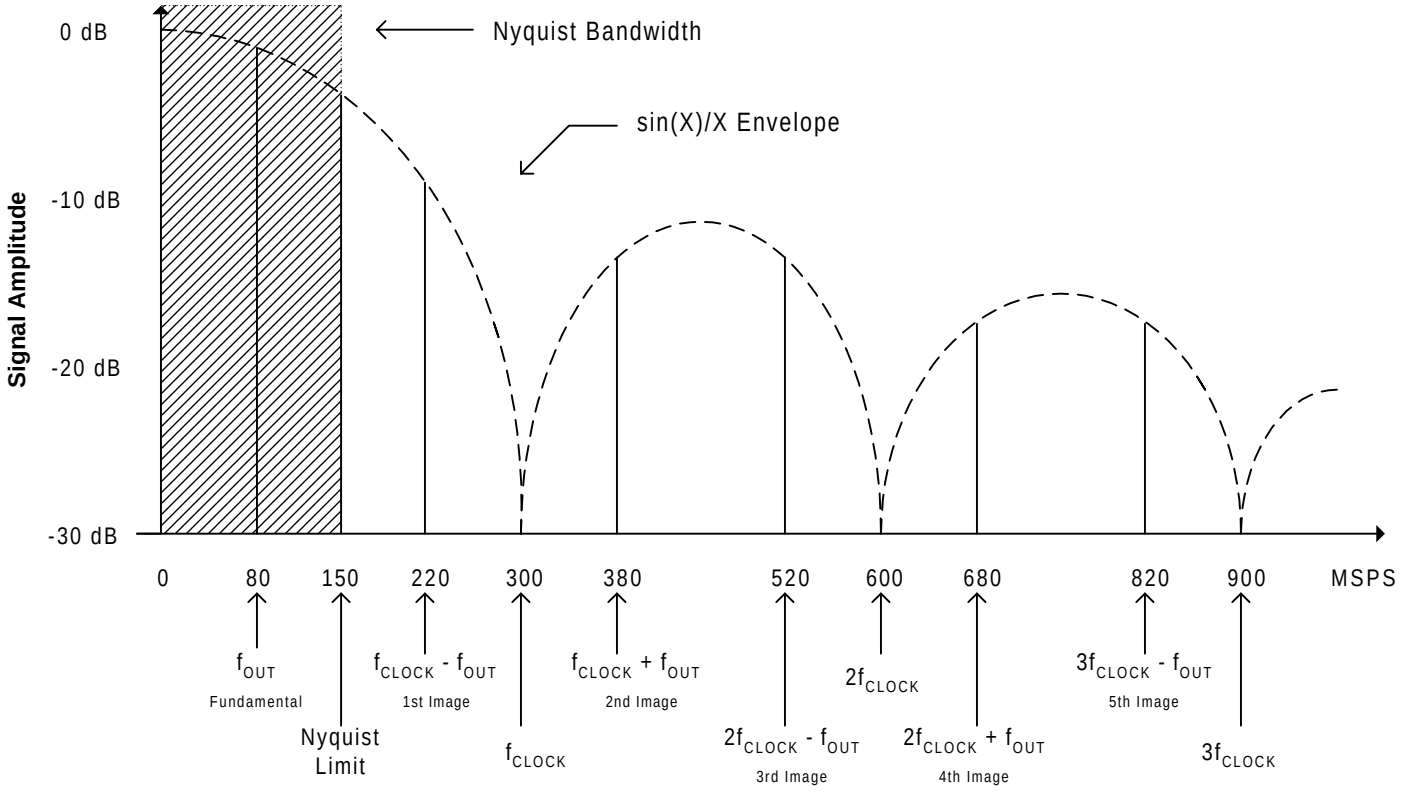


Figure 2-1. Spectral Analysis of Sampled Output

The Nyquist Theorum dictates that there is a minimum of two samples per cycle required to reconstruct the desired output waveform. Images responses are created in the sampled output spectrum at $f_{CLOCK} \pm f_{OUT}$. The 1st image response occurs in this example at $f_{CLOCK} - f_{OUT}$ or 220 MHz . The 3rd, 4th, and 5th images appear at 380 MHz, 520 MHz, 680 MHz, and 820 MHz (respectively). Notice that nulls appear at multiples of the sampling frequency.

In the case of the f_{OUT} frequency exceeding the f_{CLOCK} frequency, the 1st image response will appear within the Nyquist bandwidth (DC - ½ f_{CLOCK}) as an aliased image. The aliased image cannot be filtered from the output with the traditional Nyquist anti-aliasing filter.

In typical DDS applications, a lowpass filter is utilized to suppress the effects of the image responses in the output spectrum. In order to keep the cutoff requirements on the lowpass filter

reasonable, it is an accepted rule to limit the f_{OUT} bandwidth to approximately 40% of the f_{CLOCK} frequency. This facilitates using an economical lowpass filter implementation on the output. In section X of this seminar, there will be discussion on creating and isolating image responses as a mechanism for synthesizing higher agile frequencies from DDS devices.

As can be seen in Figure 2-1, the amplitude of the F_{OUT} and the image responses follows a sin(X)/X rolloff response. This is due to the quantized nature of the sampled output. The amplitude of the fundamental and any given image response can be calculated using the sin(X)/X formula. Per the rolloff response function, the amplitude of the fundamental output will decrease inversely to increases in its tuned frequency. The amplitude rolloff due to sin(X)/X in a DDS system is –3.92 dB over its DC to Nyquist bandwidth. As was previously shown in Figure 1-4, DDS architectures can include an inverse SINC filtering which pre-compensates for the sin(X)/X rolloff and maintains a flat output amplitude (± .1 dB) from the D/A converter over a bandwidth of up to 45% of the clock rate or 80% of Nyquist.

It is important to note in the sin(X)/X response curve shown in Figure 2-1 that the amplitude of the 1st image is substantial: it is within 3dB of the amplitude of the fundamental at $f_{OUT} = .33$ f_{CLOCK}. It is important to generate a frequency plan in DDS applications and analyze the spectral considerations of the image response and the sin(X)/X amplitude response at the desired f_{OUT} and f_{CLOCK} frequencies.

The other anomalies in the output spectrum, such as integral and differential linearity errors of the D/A converter, glitch energy associated with the D/A converter, and clock feed-through noise, will not follow the sin(X)/X roll-off response. These anomalies will appear as harmonics and spurious energy in the output spectrum and will generally be much lower in amplitude than the image responses. The general noise floor of a DDS device is determined by the cumulative combination of substrate noise, thermal noise effects, ground coupling, and a variety of other sources of low-level signal corruption. The noise floor, spur performance, and jitter performance of a DDS device is greatly influenced by circuit board layout, the quality of its power supplies, and the quality of the input reference clock. Each of these subjects will be addressed individually in following sections of this tutorial.

Section 3. Frequency/phase-hopping Capability of DDS

Calculating the Frequency Tuning Word

The output frequency of a DDS device is determined by the formula:

$F_{OUT} = (M (REFCLK)) /2^N$

Where: F_{OUT} = the output frequency of the DDS
M = the binary tuning word
REFCLK = the internal reference clock frequency
N = The length in bits of the phase accumulator

The length of the phase accumulator (N) is the length of the tuning word which determines the degree of frequency tuning resolution of the DDS implementation. Let's find the frequency tuning word for an output frequency of 41 MHz where REFCLK is 122.88 MHz and the tuning word length is 32 bits (binary). The resulting equation would be:

$41 \text{ MHz} = (M (122.8 \text{ MHz})) /2^{32}$

solving for M…

$M = (41 \text{ MHz}(2^{32}))/122.8 \text{ MHz}$

M= 556AAAAB hex

Loading this value of M into the frequency control register would result in a frequency output of 41 MHz, given a reference clock frequency of 122.8 MHz.

Determining Maximum Tuning Speed

The maximum tuning speed of a DDS implementation is determined by the loading configuration selected, parallel byte or serial word, and the speed of the control interface. In some DDS applications, maximum output frequency tuning speed is desired. Applications such as GMSK and ramped-FSK modulation, require maximum frequency tuning speeds to support spectrally-shaped transitions between modulation frequencies. When the tuning word is loaded by the control interface, the constraint to frequency update is in the speed of the interface port. Typically a DDS device will provide a parallel byte load which facilitates getting data into the control registers at a higher rate. Control data clocking rates of 100 MHz are typically supported for a byte-load parallel control interface. This means that a new tuning word can be present on the output of a DDS device every 10 nS. The phase-continuous output of DDS frequency transitions is well-suited for high-speed frequency-hopping applications.

DDS devices also usually provide a set of registers that can be pre-programmed with tuning words. The contents of these registers are executed with an external pin on the device package. This provides for the maximum output frequency hopping speed between pre-programmed frequency values. This arrangement is especially suitable for FSK modulation applications where the "mark" and "space" frequencies can be readily pre-programmed. When using the pre-

programmed registers, DDS output frequency hopping speeds of up to 250 MHz can be achieved with the latest technology devices.

The DDS Control Interface

All of the functions, features, and configurations of a DDS device are generally programmed through the device's control interface port. The control interface for DDS devices is available in a variety of configurations. The common configurations are serial interface and byte-load parallel interface. The interface conventions range from a single 40-bit register that stores all of the functional control words, to a microprocessor-compatible a synchronous serial communications port. Control interface functionality and timing diagrams are detailed in the data sheets for the individual DDS devices.

Profile Registers

Pre-programmed registers are typically available in a DDS device that allow enhanced frequency or phase hopping of the output signal. The data contained in these registers are executed via a dedicated pin on the package and allow the use to change an operating parameter without going through the control interface instruction cycle. Examples of the types of functions that can be pre-programmed are:

- Output frequency tuning word – this allows the user to achieve the maximum frequency hopping capability with a DDS device. The availability of frequency select registers also facilitates using the DDS device as an FSK modulator where the input data directly steers the output to the desired mark and space frequencies.

- Phase of the output frequency – this function allows the user to execute pre-programmed increments of phase delay to the output signal. The amount of delay resolution ranges from ± 11.5° increments (5-bits) to ± .02° increments (14-bits). Phase-shift keying modulation (PSK) can readily be accomplished with the use of pre-programmed phase registers.

- In digital modulator and quadrature upconverter implementations of DDS architectures (to be covered in Section 8 of this seminar), additional functions can be pre-programmed in profile registers. These functions include FIR filter response, interpolation (upsampling) rates , and output spectral inversion enable/disable.

Section 4. The Effect of DAC Resolution on Spurious Performance

By Ken Gentile, Systems Engineer, Analog Devices, Inc.

The resolution of a DAC is specified by the number of its input bits. For example, the resolution of a DAC with 10 input bits is referred to as having "10-bit resolution". The impact of DAC resolution is most easily understood by visualizing the reconstruction of a sine wave.

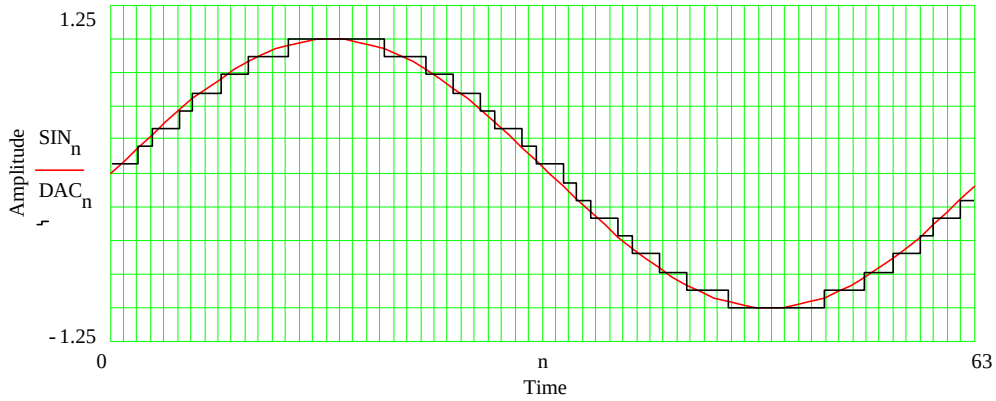


Figure 4.1. Effect of DAC Resolution

Consider Figure 4.1 in which a 4-bit DAC (quantized black trace) is used to reconstruct a perfect sine wave (smooth red trace). The vertical lines are time markers and identify the instants in time at which the DAC output is updated to a new value. Thus, the horizontal distance between the vertical lines represents the sample period. Note the deviation between the DAC output signal and the perfect sine wave. The vertical distance between the two traces at the sampling instants is the error introduced by the DAC as a result of its finite resolution. This error is known as **quantization error** and gives rise to an effect known as **quantization distortion**.

To understand the nature of the quantization distortion, note the sharp edges in the DAC output signal. These sharp edges imply the presence of high frequency components superimposed on the fundamental. It is these high frequency components that constitute quantization distortion. In the frequency domain, quantization distortion errors are aliased within the Nyquist band and appear as discrete spurs in the DAC output spectrum.

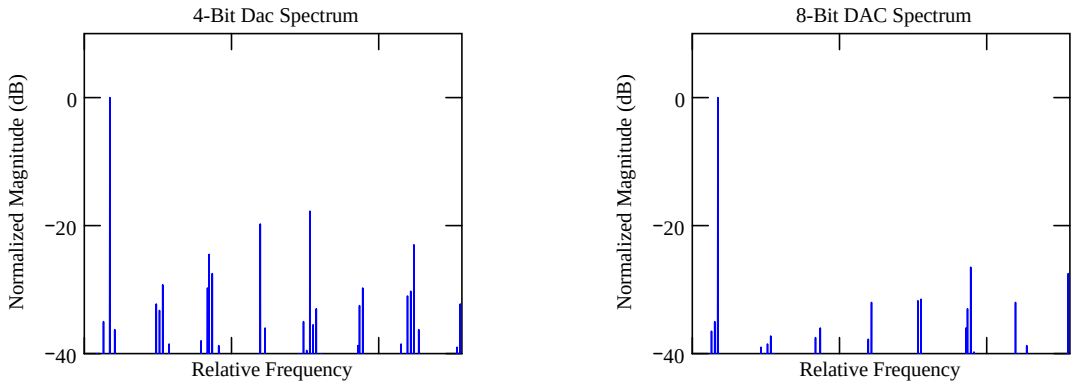


Figure 4.2. 4-Bit vs. 8-Bit DAC Output Spectra

As the DAC resolution increases the quantization distortion decreases; i.e., the spurious content of the DAC output spectrum decreases. This makes sense because an increase in resolution results in a decrease in quantization error. This, in turn, results in less error in the reconstructed sine wave. Less error implies less distortion; i.e., less spurious content. This is graphically depicted in Figure 4.2. Note that the spurs associated with the 8-bit DAC are generally lower than those of the 4-bit DAC.

In fact, the relationship between DAC resolution and the amount of distortion is quantifiable. If the DAC is operated at its fullscale output level, then the ratio of signal power to quantization noise power (SQR) is given by:

$$SQR = 1.76 + 6.02B \quad (dB)$$

Where B is the number of bits of DAC resolution.

For example, an 8-bit DAC exhibits an SQR of 49.92dB. It should be noted that the SQR equation only specifies the total noise power due to quantization errors. It does not provide any information as to the distribution of the spurs or the maximum spur level, only the combined power of all the spurs relative to the fundamental.

A second point to consider is that the SQR equation applies only if the DAC operates at fullscale. At output levels below full scale the power in the fundamental is reduced, but the quantization error remains constant. The net effect is a reduction in SQR; that is, the quantization noise becomes more significant relative to the fundamental. The effect of operating the DAC at less than fullscale is quantifiable and is given as:

$$A = 20\log(FFS) \quad (dB)$$

where FFS is the fraction of fullscale at which the DAC operates. Thus, the SQR equation becomes:

$$SQR = 1.76 + 6.02B + A$$
$$= 1.76 + 6.02B + 20\log(FFS) \quad (dB)$$

Continuing the previous example, if operate the DAC at 70% of fullscale (A=0.7) the resulting SQR is 46.82dB (a 3.1dB reduction from the original SQR performance).

The Effects of Oversampling on Spurious Performance

In oversampling, a sample rate is used that is higher than that required by the Nyquist criteria. Remember, Nyquist requires that the bandwidth of the sampled signal be constrained to ½ of the sample rate. If the bandwidth of the sampled signal is intentionally constrained to a fraction of the Nyquist requirement, then the sample rate is in excess of the Nyquist requirement and oversampling is employed.

Figure 4.3 shows how oversampling improves SQR. The amount of quantization noise power is dependent on the resolution of the DAC. It is a fixed quantity and is proportional to the shaded area. In the oversampled case, the total amount of quantization noise power is the same as in the Nyquist sampled case. Since the noise power is the same in both cases (it's constant), and the area of the noise rectangle is proportional to the noise power, then the height of the noise rectangle in the oversampled case must be less than the Nyquist sampled case in order to maintain the same area. Note that in the band of interest the area of the noise rectangle is less for the oversampled case. Thus, for a given amount of signal power in the band of interest, the signal to noise ratio is greater when oversampling is employed.

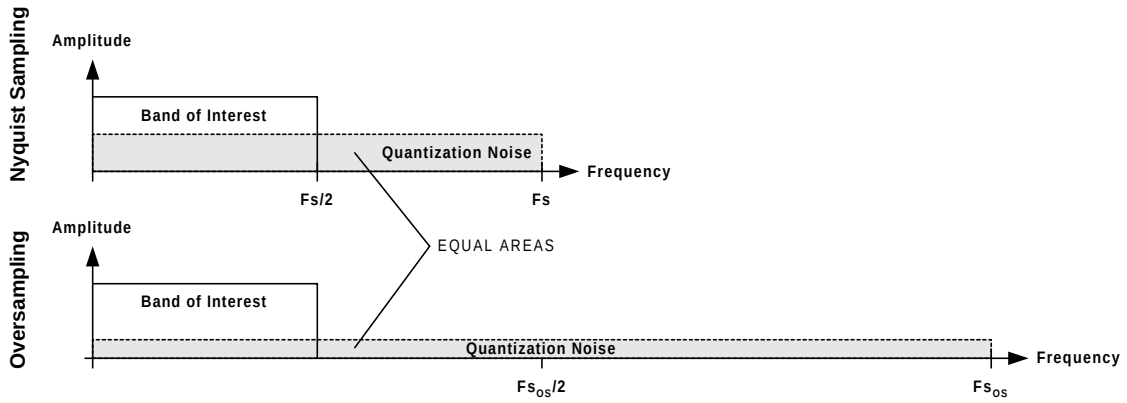


Figure 4.3. The Effect of Oversampling on SQR

The effect of oversampling is quantifiable and is given as:

$C = 10\log(Fs_{OS}/Fs)$ (dB)

Where Fs is the Nyquist sampling rate and Fs_{OS} is the oversampling rate. The modified SQR equation is:

$SQR = 1.76 + 6.02B + A + C$
$= 1.76 + 6.02B + 20\log(FFS) + 10\log(Fs_{OS}/Fs)$ (dB)

Returning to the previous example, if we operate the DAC at 70% of fullscale and oversample by a factor of 3, the SNR becomes 51.59dB. This constitutes an overall improvement of 1.67dB

over the original fullscale SQR performance. In this case, oversampling more than compensated for operating the DAC at only 70% of fullscale.

The Effect of Truncating the Phase Accumulator on Spurious Performance

Phase truncation is an important aspect of DDS architectures. Consider a DDS with a 32-bit phase accumulator. To directly convert 32 bits of phase to a corresponding amplitude would require 2^{32} entries in a lookup table. That's 4,294,967,296 entries! If each entry is stored with 8-bit accuracy, then 4-gigabytes of lookup table memory would be required. Clearly, it would be impractical to implement such a design.

The solution is to use a fraction of the most significant bits of the accumulator output to provide phase information. For example, in a 32-bit DDS design, only the upper most 12 bits might be used for phase information. The lower 20 bits would be ignored (truncated) in this case.

To understand the implications of truncating the phase accumulator output it is helpful to use the concept of the "digital phase wheel". Consider a simple DDS architecture that uses an 8-bit accumulator of which only the upper 5 bits are used for resolving phase. The phase wheel depiction of this particular model is shown in Figure 4.4.

With an 8-bit accumulator, the phase resolution associated with the accumulator is $1/256^{th}$ of a full circle, or 1.41° ($360/2^8$). In Figure 4.4, the accumulator phase resolution is identified by the outer circle of tic marks. If only the most significant 5 bits of the accumulator are used to convey phase information, then the resolution becomes $1/32^{nd}$ of a full circle, or 11.25° ($360/2^5$). These are identified by the inner circle of tic marks.

Now let us assume that a tuning word value of 6 is used. That is, the accumulator is to count by increments of 6. The first four phase angles corresponding to 6-count steps of the accumulator are depicted in Figure 4.4. Note that the first phase step (6 counts on the outer circle) falls short of the first inner tic mark. Thus, a discrepancy arises between the phase of the accumulator (the outer circle) and the phase as determined by 5-bit resolution (the inner circle). This descrepancy results in a phase error of 8.46° (6 x 1.41°), as depicted by arc ***E1*** in the figure.

On the second phase step of the accumulator (6 more counts on the outer circle) the phase of the accumulator resides between the 1^{st} and 2^{nd} tic marks on the inner circle. Again, there is a discrepancy between the phase of the accumulator and the phase as determined by 5 bits of resolution. The result is an error of 5.64° (4 x 1.41°) as depicted by arc ***E2*** in the figure. Similarly, at the 3^{rd} phase step of the accumulator an error of 2.82° (2 x 1.41°) results. On the 4^{th} phase step, however, the accumulator phase and the 5-bit resolution phase coincide resulting in no phase error. This pattern continues as the accumulator increments by 6 counts on the outer circle each time.

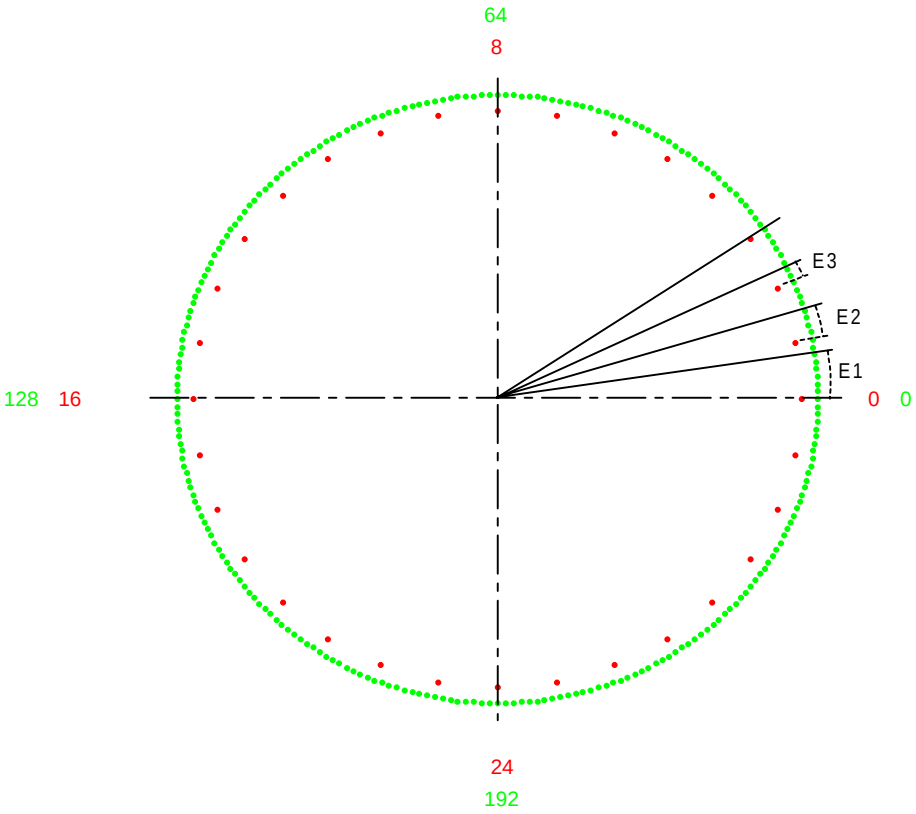


Figure 4.4. Phase Truncation Error and the Phase Wheel

Obviously, the phase errors introduced by truncating the accumulator will result in errors in amplitude during the phase-to-amplitude conversion process inherent in the DDS. It turns out that these errors are periodic. They are periodic because, regardless of the tuning word chosen, after a sufficient number of revolutions of the phase wheel, the accumulator phase and truncated phase will coincide. Since these amplitude errors are periodic in the time domain, they appear as line spectra (spurs) in the frequency domain and are what is known as **phase truncation spurs**.

It turns out that the magnitude and distribution of phase truncation spurs is dependent on three factors (Ref. [3]):

1. Accumulator size (**A** bits)
2. Phase word size (**P** bits); i.e., the number of bits of phase after truncation
3. Tuning word (**T**)

Phase Truncation Spur Magnitude

Certain tuning words yield no phase truncation spurs at all while others yield spurs with the maximum possible level. If the quantity, A-P, is 4 or more (usually the case for any practical DDS design), then the maximum spur level turns out to be very closely approximated by **–6.02P** dBc (i.e., 6.02P decibels below the level of the tuning word frequency). So, a 32-bit DDS with a 12-bit phase word will yield phase truncation spurs of no more than –72dBc regardless of the tuning word chosen.

Tuning words that yield the maximum spur level are those that satisfy the following:

$$GCD(T, 2^{(A-P)}) = 2^{(A-P-1)}$$

Where GCD(X, Y) is the Greatest Common Divisor of ***both*** X and Y. In order for this equation to be true, a tuning word bit pattern for the tuning word must be as shown in Figure 4.5 below.

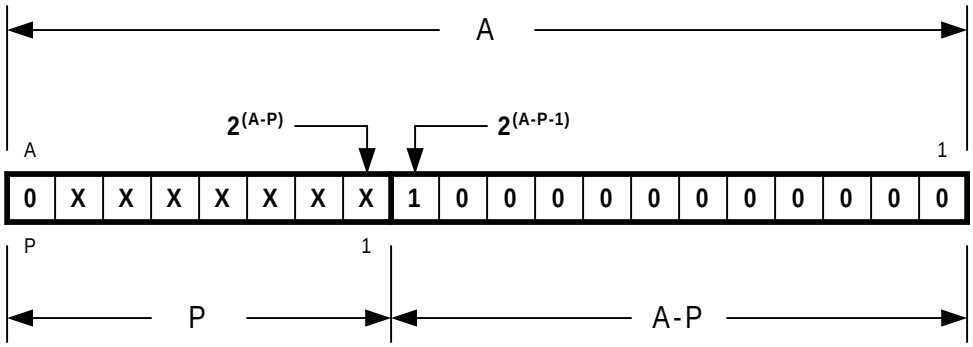


Figure 4.5. Tuning Word Patterns That Yield Maximum Spur Level

An **A**-bit word is shown, which corresponds to a phase accumulator with **A** bits of resolution. The upper **P** bits constitute the ***phase word*** (the bits that are to be used for conversion from phase to amplitude). The lower **A-P** bits are truncated, that is, ignored as far as phase resolution is concerned. The tuning word, **T**, is made up of the **A-1** least significant bits (the most significant bit of the tuning word must be a 0 to avoid the problem of aliasing). As shown in the above figure, any tuning word with a 1 in bit position $2^{(A-P-1)}$ and 0's in all less significant bit positions will yield the worst case phase truncation spur level (-6.02P dBc).

At the other extreme are tuning words that yield no phase truncation spurs. Such tuning words must satisfy,

$$GCD(T, 2^{(A-P)}) = 2^{(A-P)}$$

In order for this equation to be true, the tuning word bit pattern must be as shown in Figure 4.6 below.

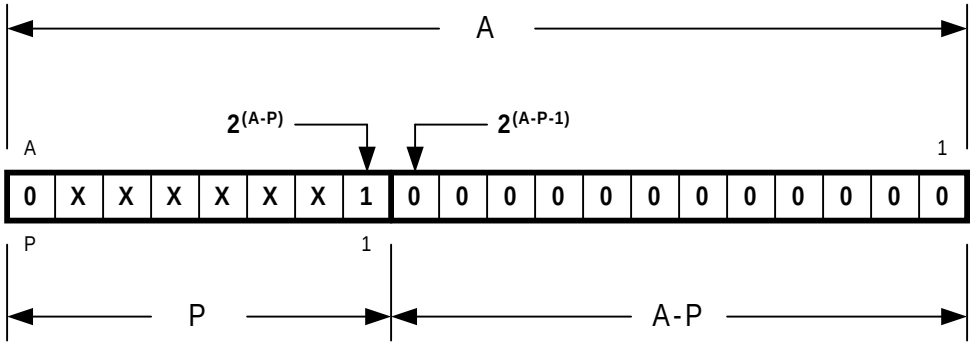


Figure 4.6. Tuning Word Patterns That Yield No Phase Truncation Spurs

Thus, tuning words that yield no phase truncation spurs are characterized by a 1 in bit position $2^{(A-P)}$ and 0's in all less significant bit positions. All other tuning word patterns that do not fit the two categories above will yield phase truncation spur levels between the two extremes.

Phase Truncation Spur Distribution

To precisely analyze the distribution of phase truncation spurs is quite complicated. A detailed analysis may be found in [3]. Rather than delve into the details of the analysis, a more intuitive presentation follows.

Remember, first of all, that the DDS core consists of an accumulator which recursively adds the tuning word value. Several iterations of this process are shown in Figure 4.7. Initially, the accumulator contains the value of the tuning word (in this case, an arbitrary binary number which has been assigned the variable, K). On each successive cycle of the DDS system clock, the tuning word is added to the previous contents of the accumulator. Remember, however, that the accumulator is modulo 2^A, so bits that would carry beyond the MSB are simply dropped. As the accumulator sequence proceeds the value of the accumulator will eventually return to the original tuning word value and the sequence will repeat. The number of steps (or clock cycles) required to accomplish this is known as the Grand Repitition Rate (**GRR**). The formula for determining the GRR is:

$$GRR = 2^A / GCD(T, 2^A)$$

For example, in case shown, A is 20 and T is 182,898 (base 10), which yields a GRR of 524,288. From this result it can be seen that over a half a million clock cycles are required before the accumulator begins to repeat its sequence. Although this may seem like a long repitition period, keep in mind that some DDS cores use 48-bit accumlators (A=48), which can yield enormous GRR values.

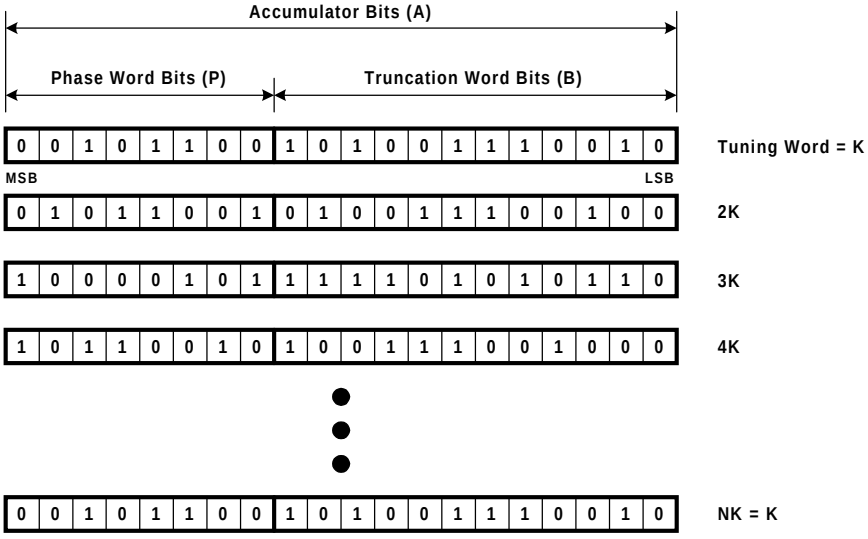


Figure 4.7. Accumulator Sequence

Refer, once again, to Figure 4.7. The P-bits of the phase word are passed along to the phase-to-amplitude conversion portion of the DDS, which is used to produce the output waveform. However, the B-bits of the truncation word are not passed along to the phase-to-amplitude converter. Therefore, if the full A bits of the accumulator represent the true phase, but only P-bits of the phase word are used for determining amplitude, then the output signal is essentially in error by the value of the truncation word. Thus, the output signal can be thought of as a composite of a full resolution signal (that which would be obtained with no phase truncation) and an error signal due to the B-bits of the truncation word.

The error signal, then, is a source of spurious noise. Since the error signal is defined by the truncation word, then analysis of the behavior of the truncation word should allow some insight into the nature of the error signal. Thus, we shall focus on only the truncation word and ignore the phase word.

If only the truncation bits are considered, it is possible to determine the period over which the truncation word repeats; i.e., the GRR of the truncation word. For example, for the conditions given in Figure 4.7, the value of A becomes 12 (the number of truncation bits). The truncation word behaves as a B-bit accumulator with an equivalent tuning word (**ETW**) given by,

$$\text{ETW} = \text{T modulus } 2^B$$

Where T is the original tuning word. The result of this operation is nothing more than the value of the truncation word portion of the original tuning word. For the given example the ETW is 2,674 (base 10). So, with A=12 and T=2674, the GRR is 2,048. Thus, every 2,048 clock cycles, the truncation word will repeat the pattern of its sequence. So, at this point, we know we have an error signal that is periodic over a time interval of 2,048 clock cycles.

But what is the behavior of the truncation word within this period? That question can be answered by noting that the "**capacity**" of the truncation word is 2^B. Dividing the capacity by the ETW determines the number of clock cycles required to cause the accumulator to overflow. The capacity of the truncation word is easily calculated because in the example given B is 12. This yields a trucation word capacity of $2^{12} = 4096$.

Before we divide by the ETW, however, notice that the MSB of the ETW is a 1. This implies an overflow period of less than 2 clock cycles, which, in turn, implies that the frequency produced would be an alias. So, we must adjust the ETW by subtracting it from the capacity of the truncation word (4096). So, the adjusted ETW is 1422 (4096 – 2674). If the MSB of the ETW had been a 0, the alias adjustment procedure would not have been necessary.

Now that we know the capacity of the truncation word and the properly adjusted ETW, we can determine the overflow period of the truncation word as:

$$\text{Capacity/ETW} = 2^B/1422 = 4096/1422 = 2.88045$$

This value is the average number of clock cycles required for the truncation word to overflow. Since we know that the GRR of the truncation word is 2048 clocks and that it takes ~2.88 clocks for the truncation word to overflow, then the number of overflows that occurs over the period of the GRR is:

Number of Overflows = GRR/(Capacity/ETW) = 2048/(4096/1422) = 711

With this information it is possible to visualize the behavior of the truncation word as shown in Figure 4.8 below.

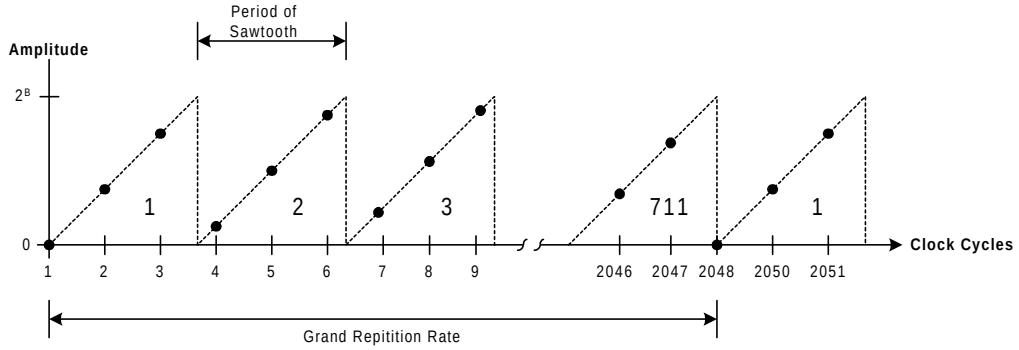


Figure 4.8. Behavior of the Truncation Word

Note that the truncation word accumulates up to a maximum value of 2^B. It has the shape of a sawtooth waveform with a period of 4096/1422 clock cycles. It should be apparent that the sawtooth shape results from the overflow characteristic of the accumulator. Also note that the complete sequence of truncation word values repeats after a period of 2048 clock cycles. Since the behavior of the truncation word is periodic in the time domain, then its fourier transform is periodic in the frequency domain. Also, the truncation word sequence is a real sequence, so the fourier transform may be represented by half as many frequency points as there are periodic time domain points (because the fourier transform of a real time domain sequence is symmetric about the origin in the frequency domain). Hence, there will be 1024 discrete frequencies associated with the behavior of the truncation word, and these frequencies constitute the truncation spurs.

Furthermore, the spectrum of the truncation word sequence will be related to that of a sawtooth waveform. The fundamental frequency of the sawtooth is F_s x (ETW/Capacity) or 0.3472 F_s for the example given. The spectrum of a sawtooth waveform is comprised of harmonics of its fundamental. Since we know that there are 1024 discrete frequencies associated with the truncation word sequence, then the spectrum consists of triangle waveform with 1024 frequencies spaced at intervals of $0.3472F_s$. This spans a frequency range of $355.5F_s$. This, of course, results in aliasing of the higher order harmonics into the Nyquist bandwidth, $F_s/2$. Figure 4.9 below illustrates this phenomenon.

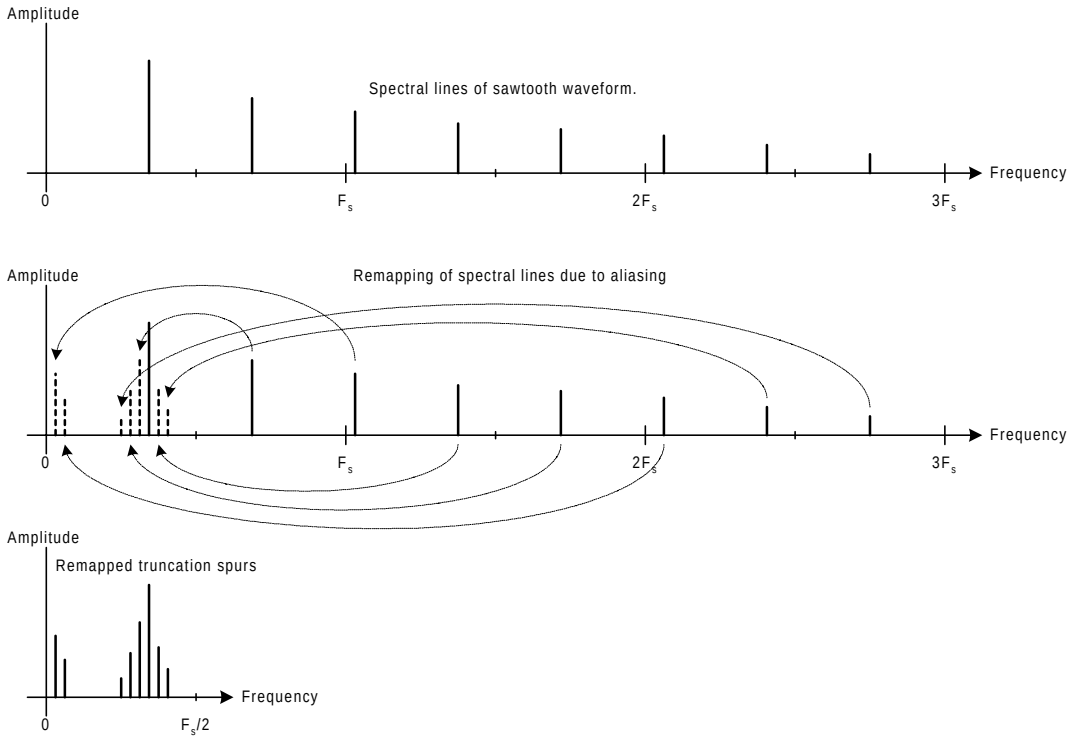


Figure 4.9: Spectrum of Truncation Word Sequence

The upper trace of Figure 4.9 shows the partial spectrum of the sawtooth waveform. The middle trace shows the remapping of the spectral lines due to aliasing. Note that aliasing causes spurs in frequency bands that are odd integer multiples of $F_s/2$ to map directly into the region of $F_s/2$. While spurs that occur in frequency bands that are even multiples of $F_s/2$ map as mirror images into the region of $F_s/2$. Such is the nature of the aliasing phenomenon. The bottom trace of the figure shows only the region $F_s/2$ (the Nyquist band) with the remapped spectral lines. This is the actual truncation spur spectrum produced by the DDS. Keep in mind however, that Figure 4.9 only displays the frequency range of 0 to $3F_s$. The full spectrum of the sawtooth waveform actually spans 355.5 F_s. Thus, there are many more truncation spurs present than are actually shown in the Figure 4.9 (the intent of Figure 4.9 is to demonstrate the concept rather than to be exhaustively accurate).

Phase Truncation Summary

In summary, truncation of the phase accumulator results in an error in the DDS output signal. This error signal is characterized by the behavior of the truncation word (the truncation word being the portion of the phase accumulator which contains the truncated bits). Furthermore, the truncation error signal causes discrete frequency spurs to appear in the DDS output and these spurs are referred to as phase truncation spurs.

The magnitude of the phase truncation spurs has an upper bound that is determined by the number of bits in the phase word (P). The value of that upper bound is –6.02P dBc and this upper bound occurs for a specific class of tuning words. Namely, those tuning words for which the truncated bits are all 0's except for the most significant truncated bit. However, a second class of tuning words results in no phase truncation spurs. These are characterized by all 0's in

the truncation word and a 1 in *at least* the LSB position of the phase word. All other classes of tuning words produce phase truncations spurs with a maximum magnitude *less than* –6.02P dBc.

The distribution of the truncation words is not as easily characterized as the maximum magnitude. However, it has been explained that the truncation word portion of the accumulator can be thought of as the source of an error phase signal. This error signal is of the form of a sawtooth waveform with a frequency of:

$F_s(ETW/2^B)$,

where F_s is the DDS system clock frequency, ETW is the equivalent tuning word represented by the truncated bits (after alias correction), and B is the number of truncation bits. The number of harmonics of this frequency which must be considered for the analysis for phase truncation spurs is given by:

$2^{B-1} / GCD(ETW, 2^B)$,

where GCD(x,y) is the Greatest Common Divisor of x and y. The result is a spectrum which spans many multiples of F_s. Therefore, a remapping of the harmonics of the sawtooth spectrum must be performed due to aliasing. The result of the remapping places all of the spurs of the sawtooth spectrum within the Nyquist band ($F_s/2$). This constitutes the distribution of the phase truncation spurs as produced by the DDS.

Additional DDS Spur Sources

The previous two sections addressed two of the sources of DDS spurs; DAC resolution and phase truncation. Additional sources of DDS spurs include:

1. DAC nonlinearity
2. Switching transients associated with the DAC
3. Clock feedthrough

DAC nonlinearity is a consequence of the inability to design a perfect DAC. There will always be an error associated with the expected DAC output level for a given input code and the actual output level. DAC manufacturers express this error as DNL (differential nonlinearity) and INL (integral nonlinearity). The net result of DNL and INL is that the relationship between the DAC's expected output and its actual output is not perfectly linear. This means that an input signal will be transformed through some nonlinear process before appearing at the output. If a perfect digital sine wave is fed into the DAC, the nonlinear process causes the output to contain the desired sine wave plus harmonics. Thus, a distorted sine wave is produced at the DAC output. This form of error is known as *harmonic distortion*. The result is harmonically related spurs in the output spectrum. The amplitude of the spurs is not readily predictable as it is a function of the DAC linearity. However, the location of such spurs is predictable, since they are harmonically related to the tuning word frequency of the DDS. For example, if the DDS is tuned to 100kHz, then the 2nd harmonic is at 200kHz, the 3rd at 300kHz, and so on. Generally, for a DDS output frequency of f_o, the n^{th} harmonic is at nf_o. Remember, however, that a DDS is a sampled system operating as some system sample rate, F_s. So, the Nyquist criteria are applicable. Thus, any harmonics greater than $\frac{1}{2}F_s$ will appear as aliases in the frequency range between 0 and $\frac{1}{2}F_s$ (also known as the first Nyquist zone). The 2nd Nyquist zone covers the

range from ½F_s to F_s. The 3rd Nyquist zone is from F_s to 1.5F_s, and so on. Frequencies in the ODD Nyquist zones map directly onto the 1st Nyquist zone, while frequencies in the EVEN Nyquist zones map in mirrored fashion onto the 1st Nyquist zone. This is shown pictorially in Figure 4.10.

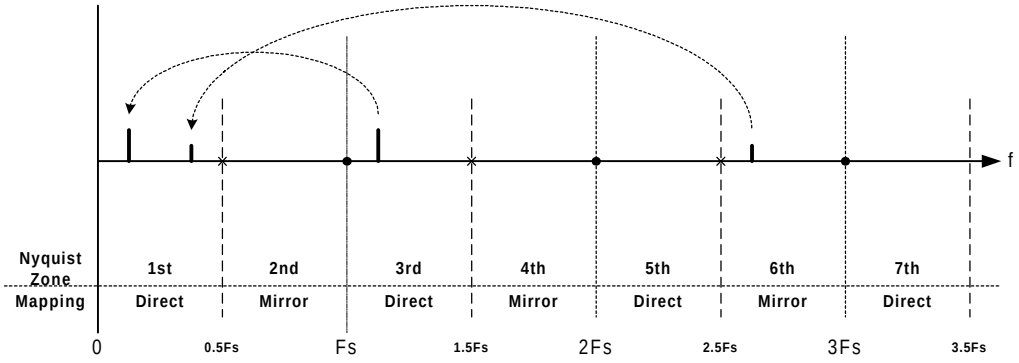


Figure 4.10. Nyquist Zones and Aliased Frequency Mapping

The procedure, then, for determining the aliased frequency of the N^{th} harmonic is as follows:

1. Let R be the remainder of the quotient $(Nf_o)/F_s$, where N is an integer.
2. Let $SPUR_N$ be the aliased frequency of the N^{th} harmonic spur.
3. Then $SPUR_N = R$ if $(R \leq ½F_s)$, otherwise $SPUR_N = F_s - R$.

The above algorithm provides a means of predicting the location of harmonic spurs that result from nonlinearities associated with a practical DAC. As mentioned earlier, the magnitude of the spurs is not predictable because it is directly related to the amount of non-linearity exhibited by a particular DAC (i.e., non-linearity is DAC dependent).

Another source of spurs are **switching transients** that arise within the internal physical architecture of the DAC. Non-symmetrical rising and falling switching characteristics such as unequal rise and fall time will also contribute to harmonic distortion. The amount of distortion is determined by the effective ac or dynamic transfer function. Transients can cause ringing on the rising and/or falling edges of the DAC output waveform. Ringing tends to occur at the natural resonant frequency of the circuit involved and may show up as spurs in the output spectrum.

Clock feedthrough is another source of DDS spurs. Many mixed signal designs include one or more high frequency clock circuits on chip. It is not uncommon for these clock signals to appear at the DAC output by means of capacitive or inductive coupling. Obviously, any coupling of a clock signal into the DAC output will result in a spectral line at the frequency of the interfering clock signal. Another possibility is that the clock signal is coupled to the DAC's sample clock. This causes the DAC output signal to be modulated by the clock signal. The result is spurs that are symmetric about the frequency of the output signal.

Proper layout and fabrication techniques are the only insurance against these forms of spurious contamination. The spectral location of clock feedthrough spurs is predictable since a device's internal clock frequencies are usually known. Therefore, clock feedthrough spurs are likely to be

found in the output spectrum coincident with their associated frequencies (or their aliases) or at an associated offset from the output frequency in the case of modulation.

Wideband Spur Performance

Wideband spurious performance is a measure of the spurious content of the DDS output spectrum over the entire Nyquist band. The worst-case wideband spurs are generally due to the DAC generated harmonics. Wideband spurious performance of a DDS system depends on the quality of both the DAC and the architecture of the DDS core. As discussed earlier, the DDS core is the source of phase truncation spurs. The spur level is bounded by the number of non-truncated phase bits and the spur distribution is a function of the tuning word. Generally speaking, phase truncation spurs will be arbitrarily distributed across the output spectrum and must be considered as part of the wideband spurious performance of the DDS system.

Narrowband Spur Performance

Narrowband spurious performance is a measure of the DDS output spectrum over a very narrow band (typically less than 1% of the system clock frequency) centered on the DDS output frequency. Narrowband spurious performance depends mostly on the purity of the DDS system clock. To a lesser degree, it depends on the distribution of spurs associated with phase truncation. The latter is only a factor, however, when phase truncation spurs happen to fall very near the DDS output frequency.

If the DDS system clock suffers from jitter, then the DDS will be clocked at non-uniform intervals. The result is a spreading of the spectral line at the DDS output frequency. The degree of spreading is proportional to the amount of jitter present. Narrowband performance is further affected when the DDS system clock is driven by a PLL (phase locked loop). The nature of a PLL is to continuously adjust the frequency and phase of the output clock signal to track a reference signal. This continuous adjustment exhibits itself as phase noise in the DDS output spectrum. The result is a further spreading of the spectral line associated with the output frequency of the DDS.

Predicting and Exploiting Spur "Sweet Spots" in a DDS' Tuning Range

In many DDS applications the output frequency need not be constrained to a single specific frequency. Rather, the designer is given the liberty to choose any frequency within a specified band that satisfies the design requirements of the system. Oftentimes, these applications specify fairly stringent spurious noise requirements, but only in a fairly narrow passband surrounding the fundamental output frequency of the DDS. In these applications, the output signal is usually bandpass filtered so that only a particular band around the fundamental output frequency is of critical importance. In these instances, the designer can select a DDS output frequency that lies within the desired bandwidth but yields minimal spurious noise within the passband.

As mentioned earlier, harmonic spurs (such as those due to DAC nonlinearity) fall a predictable locations in the output spectrum. Knowledge of the location of these spurs (and their aliases) can aid the designer in the choice of an optimal output frequency. Simply chose a fundamental frequency that yields harmonic spurs outside of the desired passband. This topic was detailed in the section entitled, “Additional DDS Spur Sources”.

Also, knowledge of the location of phase truncation spurs can prove helpful. Choosing the appropriate tuning word can result in minimal spurs in the passband of interest, with the larger phase truncation spurs appearing out of band. This topic was detailed in the section titled, "The Effect of Truncating the Phase Accumulator on Spurious Performance".

Using the above techniques, the designer can select the output frequency which results in minimum spurious noise within the desired passband. This may have the affect of increased out of band noise, but in many applications bandpass filters are employed to suppress the out of band signals. The net result is a successful implementation of a DDS system. Many times designers pass over a DDS solution because of the lack-luster spurious performance often associated with a DDS system. By employing the above techniques coupled with the improvements that have been made in DDS technology, the designer can now use a DDS in applications where only analog solutions would have been considered.

Jitter and Phase Noise Considerations in a DDS System

The maximum achievable spectral purity of a synthesized sinewave is ultimately related to the purity of the system clock used to drive the DDS. This is due to the fact that in a sampled system the time interval between samples is expected to be constant. Practical limitations, however, make perfectly uniform sampling intervals an impossibility. There is always some variability in the time between samples leading to deviations from the desired sampling interval. These deviations are referred to as **timing jitter**. There are two primary mechanisms that cause jitter the system clock. The first is thermal noise and the second is coupling noise.

Thermal noise is produced from the random motion of electrons in electric circuits. Any device possessing electrical resistance serves as a source for thermal noise. Since thermal noise is random, its frequency spectrum in infinite. In fact, in any given bandwidth, the amount of thermal noise power produced by a given resistance is constant. This fact leads to an expression for the noise voltage, V_{noise}, produced by a resistance, R, in a bandwidth, B. It is given by the equation:

$$V_{noise} = \sqrt{(4kTRB)}$$

Where V_{noise} is the RMS (root-mean-square) voltage, k is Boltzmann's constant (1.38×10^{-23} Joules/°K), T is absolute temperature in degrees Kelvin (°K), R is the resistance in ohms, and B is the bandwidth in hertz. So, in a 3000 Hz bandwidth at room temperature (300°K) a 50Ω resistor produces a noise voltage of 49.8nVrms. The important thing to note is that it makes no difference where the center frequency of the 3kHz bandwidth is located. The noise voltage of the room temperature 50Ω resistor is 49.8nVrms whether measured at 10kHz or 10MHz (as long as the bandwidth of the measurement is 3kHz).

The implication here is that whatever circuit is used to generate the system clock it will always exhibit some finite amount of timing jitter due to thermal noise. Thus, thermal noise is the limiting factor when it come to minimizing timing jitter.

The second source of timing jitter is coupled noise. Coupled noise can be in the form of locally coupled noise caused by crosstalk and/or ground loops within or adjacent to the immediate area of the circuit. It can also be introduced from sources far removed from the circuit. Interference

that is coupled into the circuit from the surrounding environment is known as EMI (electromagnetic interference). Sources of EMI may include nearby power lines, radio and TV transmitters, and electric motors, just to name a few.

The existence of jitter leads to the question: "How does timing jitter on the system clock of a DDS effect the spectrum of a synthesized sine wave?" This is best explained via Figure 4.11, which is a Mathcad simulation of a jittered sinusoid.

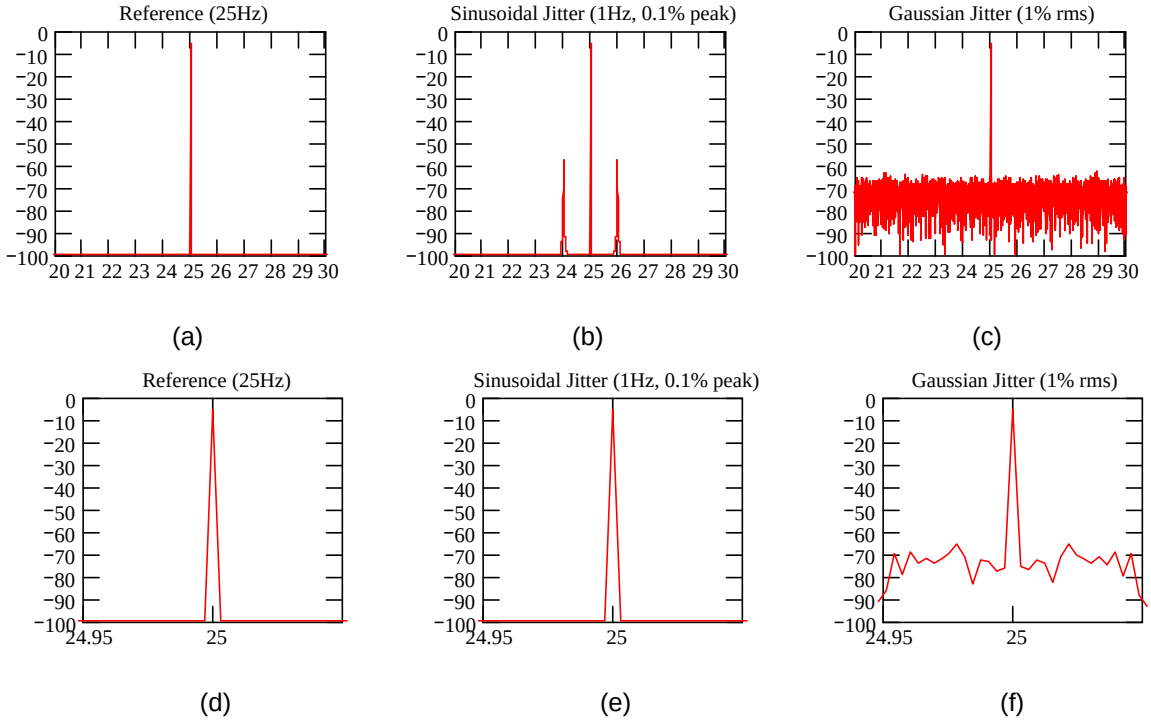


Figure 4.11. Effect of System Clock Jitter

Figure 4.11(a)-(c) span a 10Hz range centered on the 25Hz fundamental frequency, while Figure 4.11(d)-(e) is a "zoom in" around the fundamental frequency to show spectral detail near the fundamental.

Figure 4.11(a) and (d) show the spectrum of a pure sinusoid at a frequency of 25Hz. Note the single spectral line at 25Hz. This is the spectral signature of a pure sinusoid. The widening of the spectral line in Figure 4.11 (d) is a result of the finite resolution of the FFT used in the simulation.

Figure 4.11(b) and (e) show the same sinusoid but with sinusoidal timing jitter added. The jitter varies at a frequency of 1Hz and has a magnitude that is 0.1% of the period of the 25Hz fundamental. Since the period of the fundamental is 40ms, the magnitude of the jitter is 40µs peak. Thus, the sampling of the fundamental occurs at intervals that are not uniformly separated in time. Instead, the sampling instants have a timing error which causes the sampling points to occur around the ideal sampling points with a sinusoidal timing error. Thus, for the example given, the timing error oscillates around the ideal sampling points at a 1Hz rate with a peak deviation of 40µs. Note that sinusoidal jitter in the sampling clock causes modulation sidebands

to appear in the spectrum. Also, the spectral line is unchanged as can be seen by comparing Figure 4.11(d) and (e).

The frequency of the jitter can easily be determined by the separation of the sidebands from the fundamental (1Hz in this case). The magnitude of the jitter can determined by the relative amplitude of the sidebands. The following formula can be used to convert from dBc to the peak jitter magnitude:

$$\text{Peak Jitter Magnitude} = [10^{(\text{dBc}/20)}]/\pi$$

For the above case, where the jitter sidebands are at –50dBc, the peak jitter magnitude is found to be:

$$[10^{(-50/20)}]/\pi = 0.001 \text{ (or 0.1\%)}$$

This value is relative to the period of the fundamental. Thus, the absolute jitter magnitude is found by multiplying this result by the period of the fundamental (40ms). Thus, the peak jitter magnitude is 40μs (0.1% of 40ms).

Figure 4.11(c) and (f) show a pure sinusoid but with random timing jitter added. This implies that the actual sampling instants fluctuate around the ideal sampling time points in a random manner. The jitter in the example follows a Gaussian (or normal) distribution. The mean (μ) and standard deviation (σ) are 0 and 0.0004, respectively. The standard deviation of 0.0004 represents 1% of the fundamental period (or 0.4ms). The timing jitter is defined as Gaussian with a σ value of 0.0004. Thus, statistically, there is a 68% probability that the timing error of any given sampling instant is in error by no more than 0.4ms. Notice in Figure 4.11(c) that random jitter on the sampling clock results in an increase in the level of the noise floor. Furthermore, comparing Figure 4.11(f) to Figure 4.11(d), note that there is a broadening of the fundamental. The broadening of the fundamental is known by the term, **phase noise**.

Output Filtering Considerations

Fundamentally, a DDS is a sampled system. As such, the output spectrum of a DDS system is infinite. Although the device is "tuned" to a specific frequency, it is inferred that the tuned frequency lies within the Nyquist band ($0 \le f_o \le \frac{1}{2}F_s$). In actuality, the output spectrum consists of f_o and its alias frequencies as shown below in Figure 4.12.

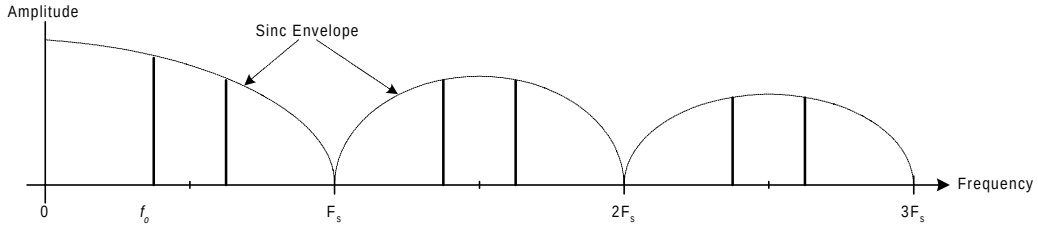


Figure 4.12. DDS Output Spectrum

The sinc (or sin[x]/x) envelope is a result of the zero-order-hold associated with the output circuit of the DDS (typically a DAC). The images of f_o continue indefinitely, but with ever

decreasing magnitude as a result of the sinc response. In the Figure 4.12, only the result of generating the fundamental frequency by means of the sampling process have been considered. Spurious noise due to harmonic distortion, phase truncation, and all other sources have been ignored for the sake for clarity.

In most applications, the aliases of the fundamental are not desired. Hence, the output section of the DDS is usually followed by a lowpass "antialiasing" filter. The frequency response of an ideal antialias filter would be unity over the Nyquist band ($0 \le f \le \frac{1}{2}F_s$) and 0 elsewhere (see Figure 4.13). However, such a filter is not physically realizable. The best one can hope for is a reasonably flat response over some percentage of the Nyquist band (say 90%) with rapidly increasing attenuation up to a frequency of $\frac{1}{2}F_s$, and sufficient attenuation for frequencies beyond $\frac{1}{2}F_s$. This, unfortunately, results in the sacrifice of some portion of the available output bandwidth in order to allow for the non-ideal response of the antialias filter.

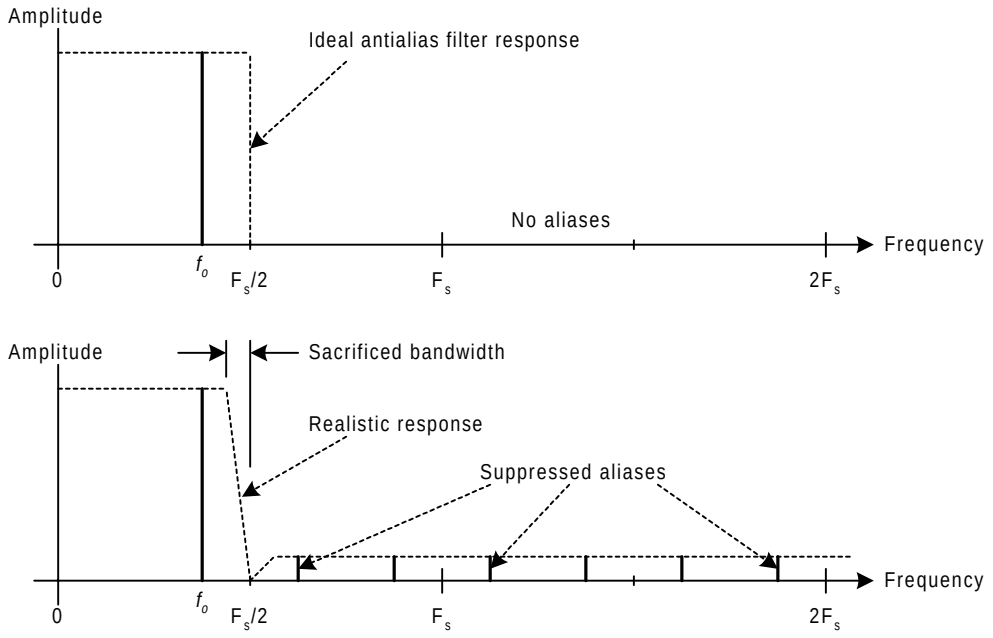


Figure 4.13. Antialias Filter

The antialias filter is a critical element in the design of a DDS system. The requirements which must be imposed on the filter design are very much dependent on the details of the DDS system. Before discussing the various types of DDS systems, it is beneficial to review some of the basic filter types in terms of their time domain and frequency domain characteristics.

First of all, it is important to clarify the relationship between the time and frequency domains as applied to filters. In the time domain, we are concerned with the behavior of the filter over time. For example, we can analyze a filter in the time domain by driving it with a pulse and observing the output on an oscilloscope. The oscilloscope displays the response of the filter to the input pulse in the time domain (see Figure 4.14).

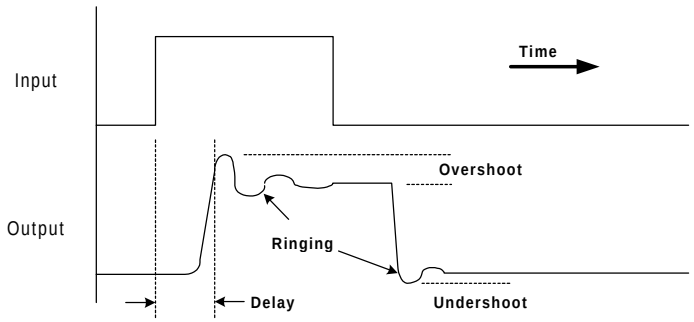


Figure 4.14. Time Domain Response

When dealing with filters (or any linear system, for that matter) there is a special case of time domain response that is fundamental in characterizing filter performance. This special case is know as **impulse response**. Impulse response is conceptually identical to the time domain figure above. The only difference is that the rectangular pulse is replaced by an ideal impulse (i.e., an infinitely large voltage spike of zero time duration). Obviously, the concept of an ideal impulse is theoretical in nature, but the response of a filter to such an input would constitute that filter's impulse response. The impulse response of a hypothetical filter is depicted below in Figure 4.15.

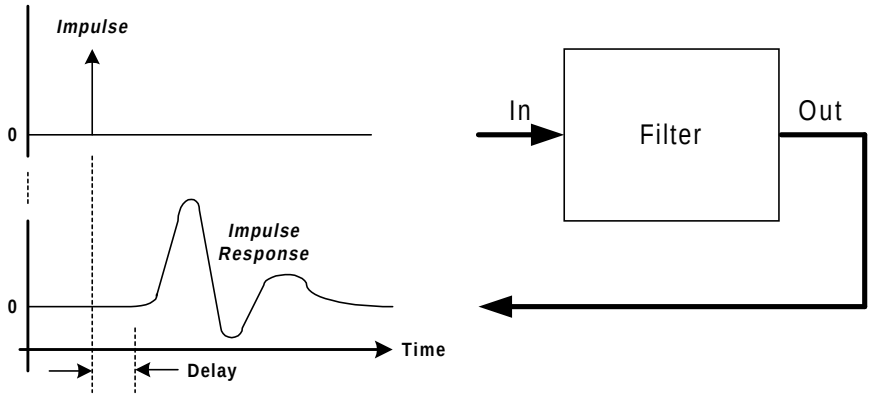


Figure 4.15. Impulse Response

Usually, when describing the behavior of a filter, a frequency domain point of view is chosen instead of a time domain point of view. In this case, the earlier oscilloscope analogy can not be used to observe the behavior of the filter. Instead, a spectrum analyzer must be employed, because it is capable of measuring magnitude vs. frequency (whereas an oscilloscope measures amplitude vs. time). A filter's frequency response is a measure of how much signal the filter will pass at a given frequency. A hypothetical lowpass filter response is shown in Figure 4.16. Typical filter parameters of interest are the cutoff frequency (f_c), the stopband frequency (f_s), the maximum passband attenuation (A_{max}) and the minimum stopband attenuation (A_{min}).

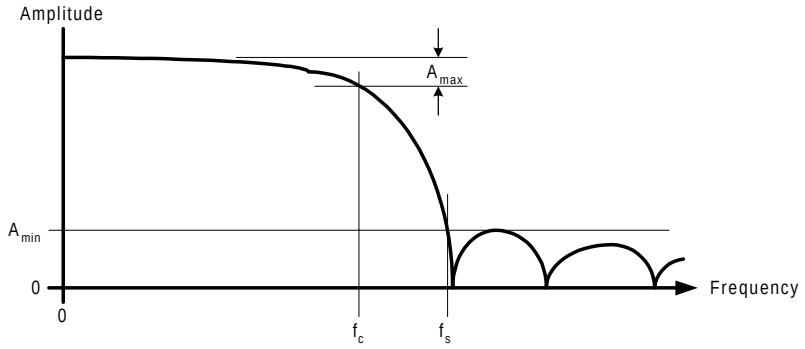


Figure 4.16. Frequency Response

Mathematically, there is a direct link between impulse response and frequency response; namely, the Fourier transform. If a filter's impulse response is known (that is, its time domain behavior), then the Fourier transform of the impulse response yields the filter's frequency response (its frequency domain behavior). Likewise, the Inverse Fourier transform of a filter's frequency response yields its impulse response. Thus, the Fourier transform (and its inverse) is the platform by which we can translate our viewpoint between the time and frequency domains.

There is an important reason for exploring the relationship between the time and frequency domains in regard to filters. Specifically, the choice of a particular filter type depends on whether an application requires a filter with certain time domain characteristics or a filter with certain frequency domain characteristics. One must realize that there exists a trade off between the desirable characteristics the two domains. Namely, a smooth time domain response and a sharp frequency domain response. Unfortunately, a filter that exhibits a sharp, well defined passband will necessarily have ringing and overshoot in its impulse response. Likewise, a filter with a smooth time domain characteristic will not yield a sharp transition between its passband and stopband.

So far, two significant aspects of filters have been presented; the time domain and the frequency domain response. Another important filter parameter is **group delay** (which is related to the time domain response). Group delay is a measure of the rate at which signals of different frequencies propagate through the filter. Generally, the group delay at one frequency is not the same as that at another frequency; that is, group delay is typically frequency dependent. This can cause a problem when a filter must carry a group of frequencies simultaneously in its passband. Since the different frequencies propagate at different rates the signals tend to spread out from one another in time. Which becomes a problem in wideband data communication applications where it is important that multi-frequency signals that are sent through a filter arrive at the output of the filter at the same time.

There are many classes of filters that exist in technical literature. However, for most applications the field can be narrowed to three basic filter families. Each is optimized for a particular characteristic in either the time or frequency domain. The three filter types are the Chebyshev, Gaussian, and Legendre families of responses. Filter applications that require fairly sharp frequency response characteristics are best served by the Chebyshev family of responses. However, it is assumed that ringing and overshoot in the time domain do not present a problem in such applications. Conversely, filter applications that require smooth time domain

characteristics (minimal overshoot and ringing and constant group delay) are best served by the Gaussian family of filter responses. In these applications it is assumed that sharp frequency response transitions are not required. For those applications that lie in between these two extremes, the Legendre filter family is a good choice. A brief description of the three filter families follows.

The Chebyshev Family of Responses

The Chebyshev family generally offers sharp frequency domain characteristics. As such, the time domain response is rather poor with significant overshoot and ringing and nonlinear group delay. This makes the Chebyshev family suitable for applications in which the frequency domain characteristics are the dominant area of concern, while the time domain characteristics are of little importance.

The Chebyshev family can be subdivided into four types of responses, each with its own special characteristics. The four types are the *Butterworth* response, the *Chebyshev* response, the *Inverse Chebyshev* response, and the *Cauer-Chebyshev* (also known as *elliptical*) response. Figure 4.17 shows the generic lowpass response of each of the Chebyshev filter types.

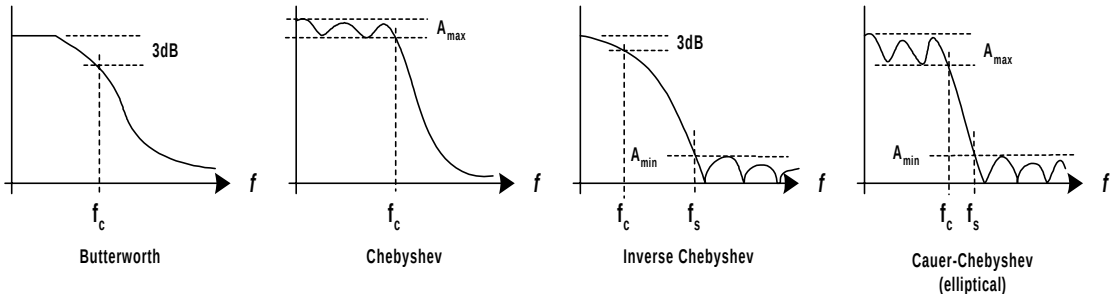


Figure 4.17. The Chebyshev Family of Responses

The Butterworth response is completely monotonic. The attenuation increases continuously as frequency increases; i.e., there are no ripples in the attenuation curve. Of the Chebyshev family of filters, the passband of the Butterworth response is the most flat. Its cutoff frequency is identified by the 3dB attentuation point. Attenuation continues to increase with frequency, but the rate of attenuation after cutoff is rather slow.

The Chebyshev response is characterized by attenuation ripples in the passband followed by monotonically increasing attenuation in the stopband. It has a much sharper passband to stopband transition than the Butterworth response. However, the cost for the faster stopband rolloff is ripples in the passband. The steepness of the stopband rolloff is directly proportional to the magnitude of the passband ripples; the larger the ripples, the steeper the rolloff.

The Inverse Chebyshev response is characterized by monotonically increasing attenuation in the passband with ripples in the stopband. Similar to the Chebyshev response, larger stopband ripples yields a steeper passband to stopband transition.

The Elliptical response offers the steepest passband to stopband transition of any of the filter types. The penalty, of course, is attenuation ripples. In this case, both in the passband and

stopband. For applications involving antialiasing filters, the elliptical is usually the filter of choice because of its steep transition region.

The Gaussian Family of Responses

The Gaussian family of responses are well suited to applications in which time domain characteristics are of primary concern. They offer smooth time domain characteristics with little to no ringing or overshoot. Furthermore, group delay is fairly constant. Since the time domain characteristics are so well behaved, it follows that the frequency domain response will not exhibit very sharp transitions. In fact, the frequency response is completely monotonic. The attenuation curve always maintains a negative slope with no peaking of the magnitude in either the passband or stopband.

The Gaussian family can be subdivided into three types of responses, each with its own special characteristics. They are the *Gaussian Magnitude* response, the *Bessel* response, and the *Equiripple Group Delay* response. Figure 4.18 shows the generic lowpass response of each of the Gaussian filter types. Although the magnitude responses of all three types seem to exhibit the same basic shape, each has its own special time domain characteristic for which it has been optimized, as expained below.

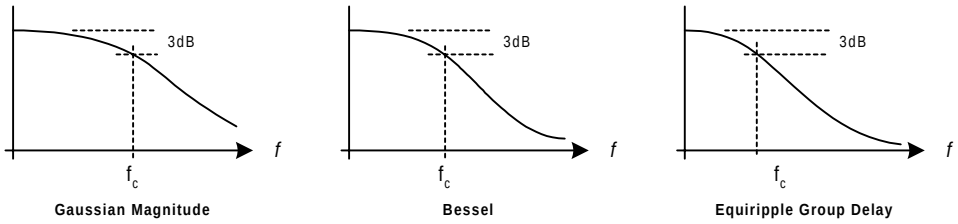


Figure 4.18. The Gaussian Family of Responses

The Gaussian Magnitude response is optimized to yield a response curve that most closely resembles a Gaussian distribution. The time domain characteristic offers nearly linear phase response with minimal overshoot or ringing. Group delay is not quite constant, but is dramatically better than that of the Chebyshev family.

The Bessel response is fully optimized for group delay. It offers maximally flat group delay in the passband. The Bessel response is to the time domain as the Butterworth response is to the frequency domain. This makes the Bessel the filter of choice where group delay is of primary concern. It offers nearly linear phase response with minimal overshoot or ringing.

The Equiripple Group Delay response is optimized to yield ripples in the group delay response that do not exceed a prescribed maximum in the passband (much like the magnitude response of the Chebyshev filter). Because the entire passband offers a certain maximum group delay, this filter well suited to wideband applications in which group delay must be controlled over the entire band of interest. Like the other Gaussian filters, the phase response is mostly linear with minimal overshoot or ringing.

The Legendre Family of Responses

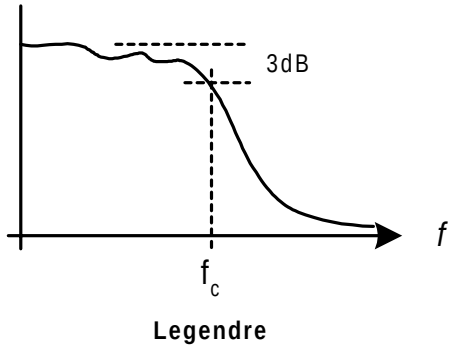


Figure 4.19. The Legendre Response

The Legendre filter family consists of a single type. Its passband response has slight ripples and is similar to a Chebyshev response with 0.1dB ripple. The stopband response monotonically decreases. The attenuation rate after the cutoff frequency is steeper than that of the Butterworth type, but not as steep as that of the Chebyshev type. Group delay is virtually constant over the first 25% of the passband, but shows ever increasing deviations as the cutoff frequency is approached.

References:

[1] Bellamy, J., 1991, Digital Telephony, John Wiley & Sons

[2] Gentile, K., 1998, "Signal Synthesis and Mixed Signal Technology", RF Design Magazine, Aug.

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

[4] Zverev, A., 1967, Handbook of Filter Synthesis, John Wiley & Sons

[5] High Speed Design Seminar, 1990 Analog Devices, Inc.

Section 5. Reference Clock Considerations

By Rick Cushing, Applications Engineer, Analog Devices Inc.

Direct Clocking of a DDS

The output signal quality of a direct digital synthesizer is dependent upon the signal quality of the reference clock that is driving the DDS. Important quality aspects of the clock source, such as frequency stability (in PPM), edge jitter (in ps or ns), and phase noise (in dBc/Hz) will be reflected in the DDS output. One quality, phase noise, is actually reduced according to: 20 LOG (Fout/Fclk). This means that a 10 MHz output signal will have 20 dB less phase noise than the 100 MHz reference clock that "created" it. The figure below illustrates how DDS processing is affected by phase noise and jitter of the input clock.

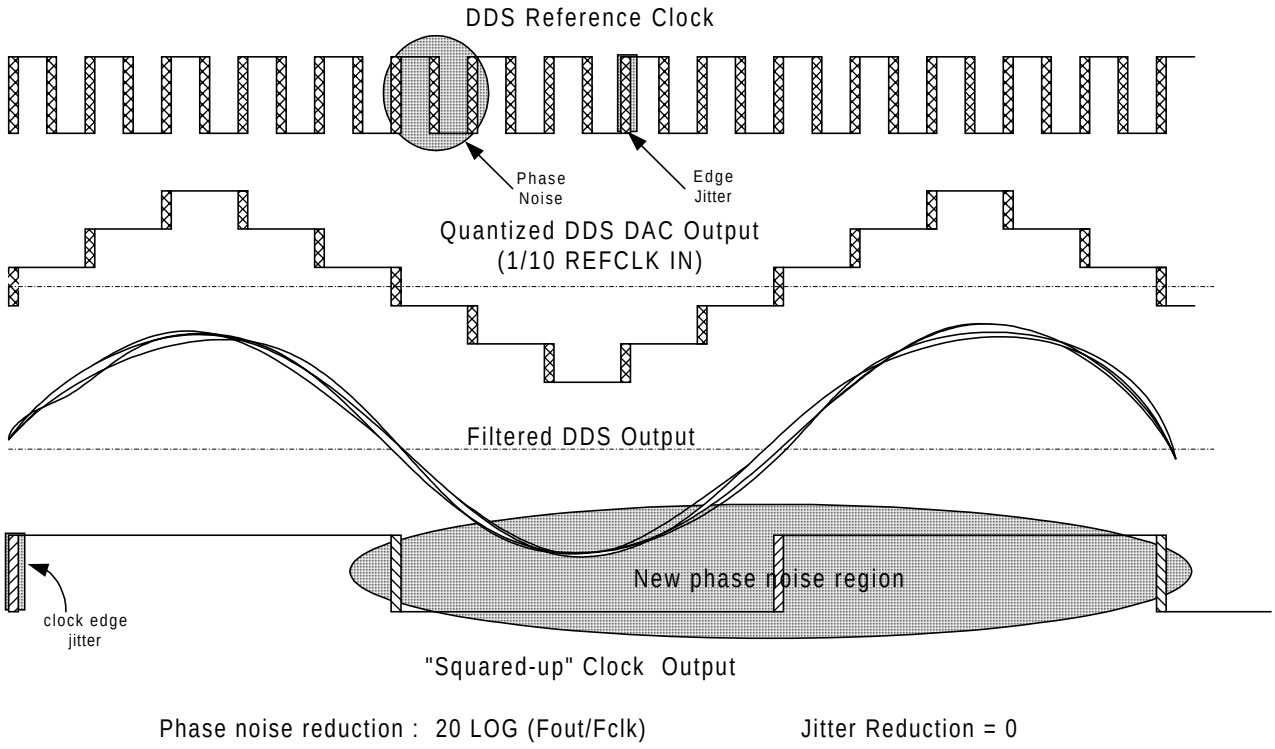


Figure 5-1. Reference clock edge uncertainty adversely affects DDS output signal quality

Figure 5-1 shows how phase noise, expressed in the time domain as period jitter with units of percent, is relative to the period of the waveform, and that absolute edge jitter is unaffected by changes in frequency or period. The "DDS Reference Clock" signal in Figure 5-1 shows that edge jitter is a much higher percentage of the total period than the same edge jitter in the "Squared-up Clock Output". This accounts for phase noise improvement through frequency division even though the same amount of edge jitter is present on both clock periods.

Reference clock edge jitter has nothing to do with the accuracy of the phase increment steps taken by the phase accumulator. These step sizes are fixed by the frequency "tuning" word and are mathematically manipulated with excellent precision regardless of the quality of the clock.

In order for the digital phase step to be properly positioned in the analog domain, two criteria must be met:

- Appropriate **amplitude** (this is the DAC's job)
- Appropriate **time** (the clock's job)

The Complete-DDS IC's from Analog Devices provide an appropriately accurate DAC to translate the digital phase steps to an analog voltage or current. But that is only half of the job. The remaining half involves accurate timing of these amplitude steps that constitute the output sine wave. This is where minimum clock edge jitter and low phase noise are required to support the precise capabilities of DDS.

The phase noise improvement of the DDS output relative to the input clock becomes more apparent in the frequency domain. Figure 5-2 is a screen-capture from a spectrum analyzer showing the phase noise of two different DDS reference clocks. The phase noise/jitter of 100 MHz DDS clock source 1 is much more pronounced than that of clock source 2.

Figure 5-3 shows the 10 MHz DDS output response to the two clock sources. Output 1 shows a 20 dB (10X improvement) in phase noise relative to clock 1. Output 2 shows less phase noise than clock 2, although 20 dB is not apparent since the noise floor of the instrument is limiting the measurement. Notice the presence of low level output "spurs" on the skirt of output 2. These spurious signals are due to the necessary truncation of phase bits in the DDS phase-to-amplitude stage and the algorithm used to perform the transformation. These spurious signals are also present in output 1 but the signal's excessive phase noise is masking their presence. This demonstrates why phase noise is important in maintaining good signal-to-noise ratio in radio and other noise-sensitive systems.

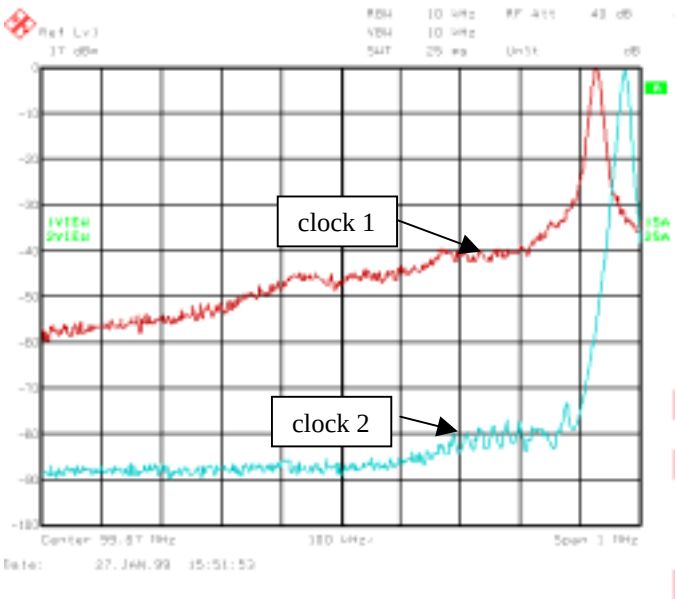


Figure 5-2: Good and poor clock phase noise

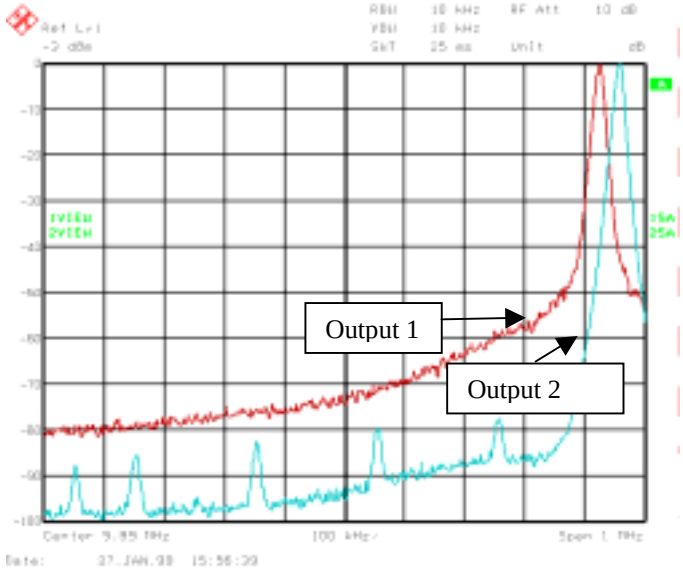


Figure 5-3: DDS output Response

There is a point at which the DDS can not mirror the quality of the input clock. For example, typical phase noise contribution of a DDS & DAC might be –130 dBc/Hz at a 1 kHz offset from the carrier. If the reference clock phase noise is better than –130 dBc/Hz then regardless of the reference oscillators good phase noise performance, the DDS & DAC output will never be better than –130 dBc/Hz at a 1 kHz offset. This DDS specification is listed as "Residual Phase Noise". One should not "over-design" with regard to the reference oscillator's phase noise specification. The DDS output phase noise performance will never exceed that of its inherent phase noise.

Overall DDS output phase noise is the sum of the phase noise of the reference clock source (after it has been enhanced by the frequency division quality of the DDS) and the residual phase noise of the DDS. For example: A reference clock oscillator has a phase noise of –110 dBc/Hz at a 1 kHz offset. The Fout/Fclk ratio is 1/10 and therefore the output phase noise reduction is –20dB. This reduction in phase noise makes the reference oscillator's phase noise at 10 MHz output equal to that of the residual phase noise of the DDS (which is given as –130 dBc/Hz at a 1 kHz offset). Adding –130dBc/Hz to –130 dBc/Hz gives a doubling of noise power and equals –127 dBc/Hz. Even if the reference clock phase noise was –200 dBc/Hz the overall DDS output phase noise will still be approximately –130 dBc.

Using an Internal Reference Clock Multiplier Circuit

Many Analog Devices DDS and digital modulator products have on-chip reference clock multiplier circuits. These multipliers, which can be engaged or bypassed, allow lower frequency clock oscillators to be used to clock the DDS at much higher frequencies. Programmable or fixed multiplier values from 4× to 20× are available. They are desirable because they can easily solve a high-speed clocking problem or allow synchronization of the DDS to a "master clock" of

another existing system clock. They permit simplified applications and reduce the cost of supplying a high frequency clock oscillator.

The REFCLK Multiplier feature is not the optimum solution for every application though. There is a tradeoff in terms of output signal quality whenever REFCLK frequency multiplication is involved. Multiplication will degrade reference oscillator phase noise within the PLL loop bandwidth by 20 LOG (Fout/Fclk), where Fout is the multiplied output frequency and Fclk is the PLL reference clock. For example, a 6X clock multiplier will degrade the input clock phase noise of a –110 dBc/Hz oscillator by 15.5 dB which results in a –94.5 dBc/Hz reference clock phase noise. Furthermore, the PLL loop filter characteristics may cause "peaking" of the phase noise response near cutoff. Figure 5-4 demonstrates typical DDS output phase noise degradation in the AD9851 device which has the entire loop filter on-chip. Other DDS devices with sections of the loop filter off-chip will generally not demonstrate peaking in the filter response.

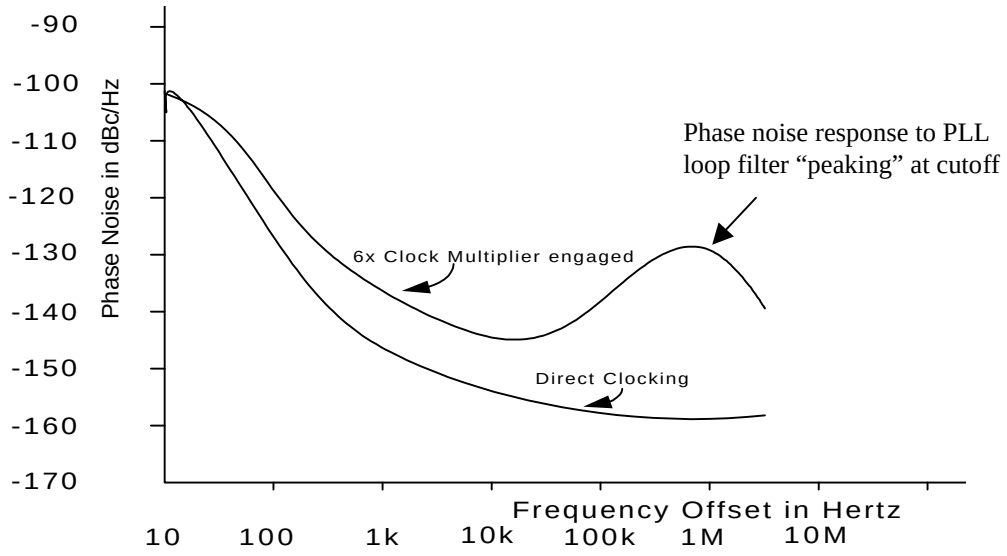


Figure 5-4. Typical DDS Phase Noise With and Without Clock Multiplier Function

DDS SFDR Performance

Use of reference clock multiplication also has an impact on SFDR (spurious-free dynamic range). Figure 5-5 shows two spectral plots of the same output frequency except output 1 has a 6× clock multiplier function engaged and output 2 is directly clocked. Close-in SFDR (+/- 1 MHz) shows SFDR of –68 dBc for the clock multiplied output and –78 dBc for the direct-clocked output. Also noticeable is the slightly elevated noise floor of output 1.

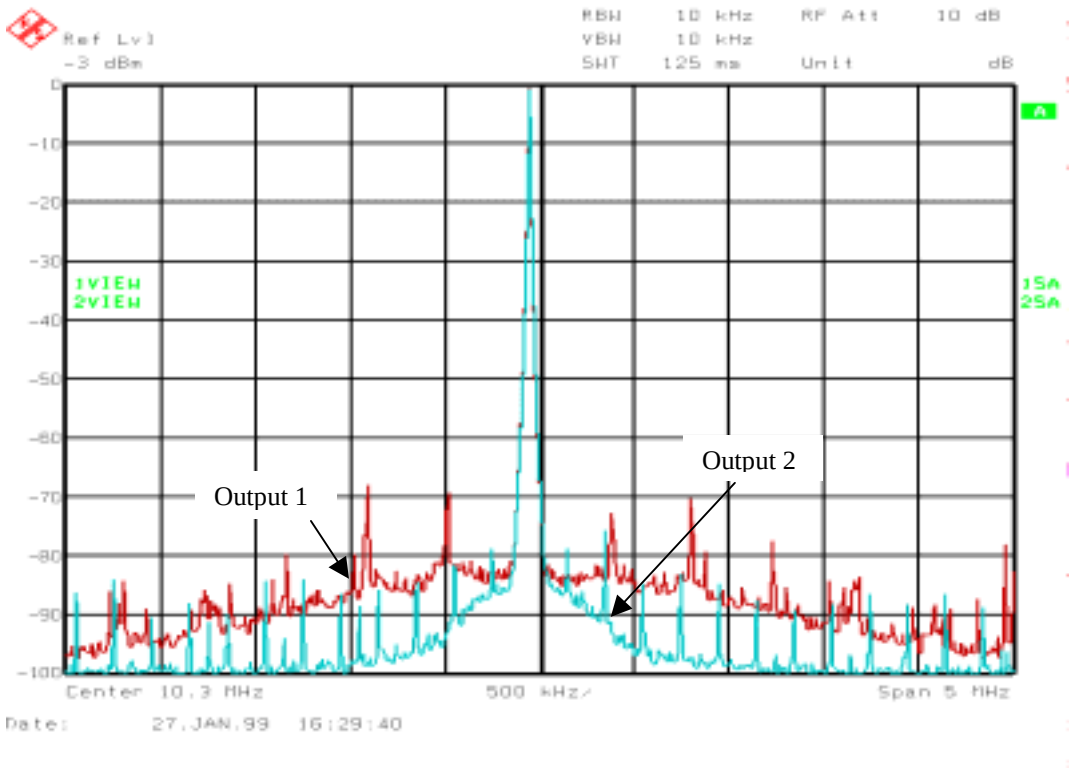


Figure 5-5. Spectral Plot of DDS Output With & Without Reference Clock Multiplication

Even considering the performance tradeoffs, the good performance, convenience and cost savings of an on-chip reference clock multiplier support its use for many, if not most, DDS applications. However, for the very best SFDR and phase noise performance, direct clocking of a DDS with a good quality clock oscillator (or sine source) is necessary.

Section 6. Interfacing to the DDS Output

By Rick Cushing, Applications Engineer, Analog Devices Inc.

Output Characteristics

High speed DDS IC's with integrated DAC's provide an output current as opposed to an output voltage. This current can be pumped into any resistive load, including a dead short, as long as the voltage developed at the DAC output pin (when referenced to ground) does not violate the DAC output compliance specification. Output compliance is simply the maximum voltage at the DAC output pin, both positive and negative, that is allowed for proper DAC functioning. Compliance voltages beyond the limits will cause moderate to drastic DAC output distortion. Normally, the outputs are terminated to ground through a resistor as in Figure 6-1. Users may terminate to any voltage that does not violate compliance specifications while the DAC is operating. The AD985X DDS DAC's will source current into a load according to the following equation: *Iout = 39.93/Rset* where Iout = Amps and Rset = ohms.

Voltage output DAC's for a DDS application are avoided due to internal I * R losses which would cause output voltage across a load to vary according to the load resistance. Current outputs will source or sink their rated current with little full-scale output variation to or from the load as long as output compliance is within limits. Current switched DACS generally exhibit better performance at higher clock rates. The DDS/DAC output resistance specification is the combined impedance (see Fig. 6-1 below) of the CMOS devices that comprise the switches and current source circuitry. The DAC output resistance is so high (usually >100k ohms) that its presence can be ignored and the load resistance, chosen by the user, essentially sets the DAC output impedance.

The output current of the AD985X DDS/DAC is unipolar. If ground is the termination point of the output load resistor, then the voltages developed across the resistor will range from 0 volts (zero-scale) to some positive or negative extreme (full-scale). In contrast, a bipolar current would develop a negative extreme voltage (zero-scale) and extend to some positive extreme voltage (full-scale)…the center point (mid-scale) between the two extremes is usually 0 volts.

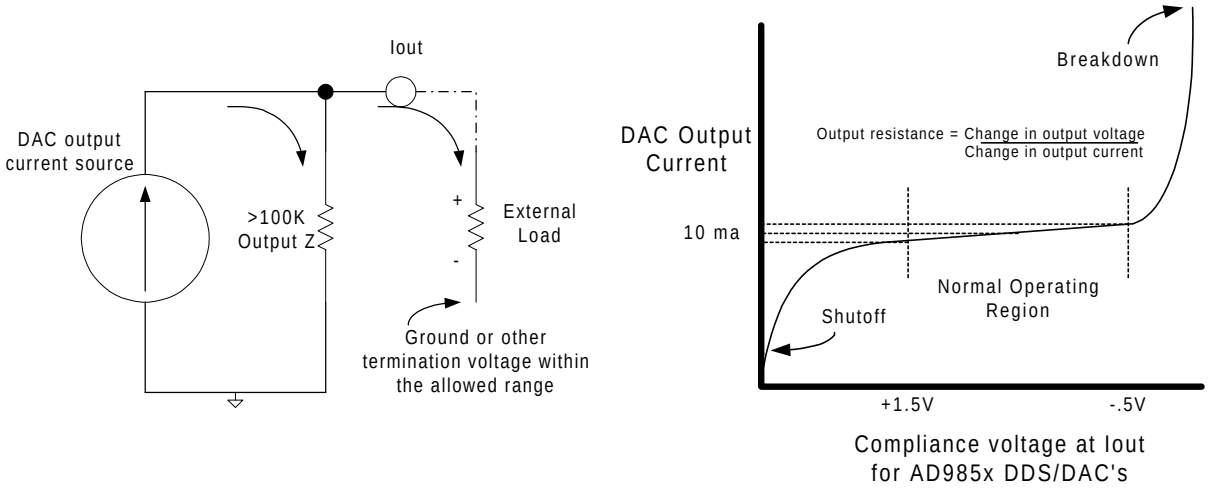


Figure 6-1. Normal Load Connection and Output Impedance

Why does unipolar current matter? First, the center point of the DDS output sine wave will be dc-offset from the load termination potential by one-half of the full-scale voltage. This may be an important consideration when applying this signal to a dc coupled amplifier since the dc component could cause amplifier clipping. Second, when AM modulating the output power envelope using the Rset resistor, the modulation envelope will be asymmetrical, looking more like a pulsed output (Figure 6-2B) than a symmetrically modulated carrier (Figure 6-2A). For more information regarding AM modulation of a DDS see Analog Devices application note AN423, available on the Analog Devices website.

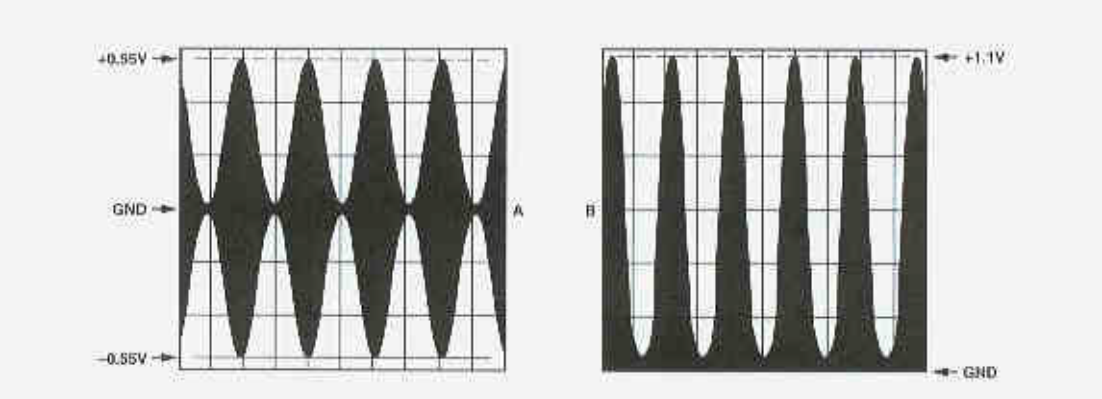


Figure 6-2. A-Symmetrical AM Modulated RF Envelope; B- Asymmetrical AM Modulation Produced from a Unipolar Current Output by Modulating the Rset Resistor

Transformer vs Single-ended Output Coupling

The DDS DAC output of the AD985X series is actually composed of two outputs, 180 degrees out of phase with each other (the true and complement). These two signals can be combined in a center-tapped RF transformer to produce the symmetrical waveform seen in Figure 6-2A. In the beginning of this section the equation for Iout was given. This equation actually defines the *sum*

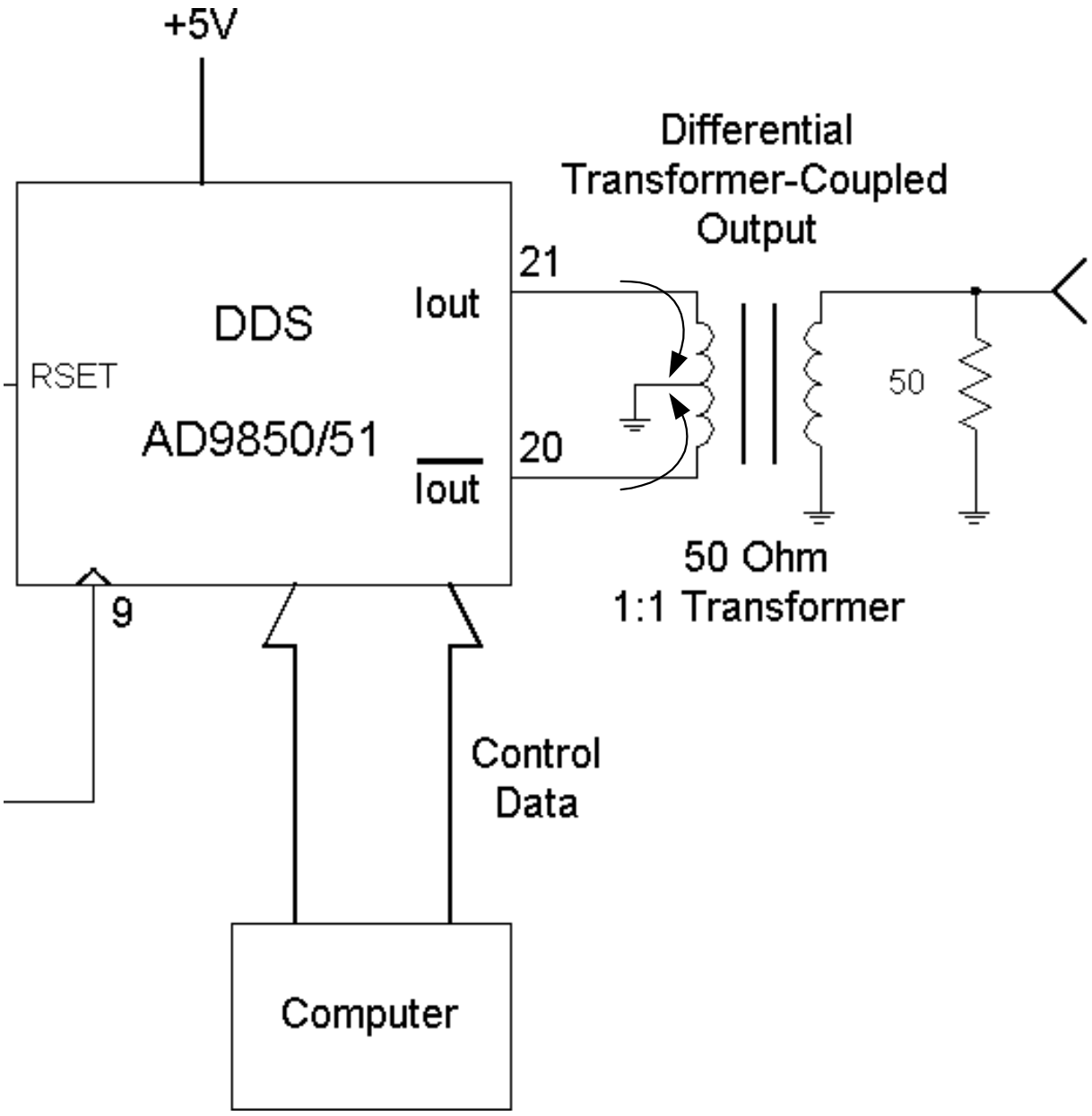


Figure 6-3. Combining Complementary Outputs to Achieve Symmetrical Output Envelope

of the two currents available from the Iout and IoutB outputs. For example, if Rset is set for10 ma full-scale, then if one output is at 2 ma, the other must be at 8 ma…if one is at 0 ma, the other must be at 10 ma, etc. By combining these two complementary currents in a transformer as in Figure 6-3, the output envelope becomes symmetrical and the dc offset is lost.

Transformer coupling is also beneficial in coupling the DAC current-outputs to reactive inputs, such as LC filters (Figure 6-4A). The low impedance pathway to ground through the transformer center tap is far better than taking the reactive pathway through an LC filter that is terminated only at the filter output (Figure 6-4C). Without a transformer, the next best method is to apply the DAC current output to a LC filter that is doubly-terminated, as shown in Figure 6-4B. The arrows in Figure 6-3 show current flow in the broadband 1:1 transformer primary and how the unipolar current of two complementary outputs can be used to simulate a bipolar current. The 50-ohm load resistance at the transformer secondary is reflected to the transformer center-tapped primary where it looks like a 25-ohm load for *each* output. Different turns-ratios will allow different loads to be used without violating the DDS DAC output compliance specification. The voltages developed at pins 20 and 21 (of an AD985X DDS) in Figures 6-3 and 6-4A will no longer be unipolar as they would be if each pin were driving a resistive load to ground (Figure 6-4B & C). Instead, the voltages will be bipolar and symmetrical around the voltage present at the center-tap - ground in this instance. This transformation from unipolar to bipolar voltage is to be expected as the magnetic fields of the center-tapped primary build and collapse. Users should pay attention to the *negative* compliance voltage as well as the *positive* compliance voltage when configuring the outputs for transformer coupling.

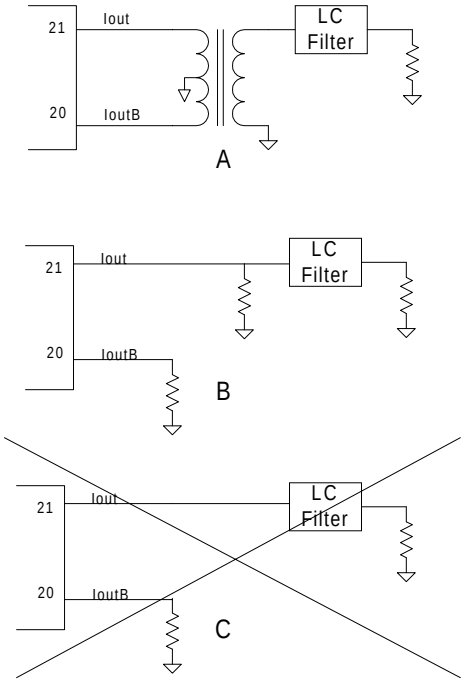


Figure 6-4: Coupling Reactive Loads to the DDS DAC Output

Another benefit to transformer coupling is the phenomenon of *common mode rejection*. If the DDS DAC outputs (Iout and IoutB) contain signals that are common or identical to each other…such as clock feedthrough, ac power supply components, or other spurious signals, then these signals can be reduced or eliminated from the output spectrum by transformer coupling. If identical signals are presented to the two transformer primary inputs as in Figure 6-4A, then their opposing fields will cancel each other to some degree. The degree of cancellation is dependent upon the transformer winding matching as well as the matching of the two "identical " signals

Output Power Considerations

Combining the two complementary outputs in a transformer does not offer any power gain. In the AD985X series of DDS IC's, the only way to increase output power is to set the output current to a higher value by adjusting Rset. Up to 20 ma maximum output current is commonly available; however, harmonic distortion of the output may also increase slightly. Use of a transformer (Figure 6-4A) does permit more efficient transfer of power to a load by eliminating the need for an input termination resistor (Figure 6-4B) which dissipates power that should have been transferred to the output termination resistor.

Output power into a 50-Ohm load for a 20 ma full-scale output sine wave is 2.5 mW or +4 dBm. Output power into the same load for a 10 ma full-scale output sine wave is .625 mW or -2 dBm. Output power is determined using the equation $P = E^2/R$, where E is the *RMS* voltage developed across the load resistance, R. P, of course, is measured in Watts. The units of dBm are arrived at using *10 * LOG (P)* where P is expressed in milliWatts. As a reminder, the term dBm expresses an absolute relationship between the power level of one milliWatt and some other power level. The term dB expresses the power level of some arbitrary reference to another power level and therefore, it is a relative measurement.

DDS/DAC Output Termination

Regardless of what output termination scheme is chosen, experience has shown that optimum spurious and harmonic suppression are achieved when both Iout and IoutB outputs are terminated equally. Failure to do so may not be noticed at lower frequencies, but at higher output frequencies, where every dB of SFDR (spurious-free dynamic range) counts, this practice will give a cleaner output spectrum, lower spurs and higher SFDR. This practice is especially applicable to situations where only one of the outputs is utilized.

Section 7. DDS as a Clock Generator

By Rick Cushing, Applications Engineer, Analog Devices, Inc.

Clock Generator Defined

A clock generator should produce a precisely timed logic pulse train with very low edge jitter and a fixed duty cycle. The logic output levels should be compatible with the device(s) that it will be "clocking". Precise timing implies a very high-Q oscillator; low edge jitter implies high noise immunity. While these attributes are relatively easy to accommodate for a single frequency, for example, a crystal clock oscillator; how can a designer accommodate the need for multiple clock frequencies that need to be changed frequently or rapidly and that have no integer relationship with each other? This is where the DDS shines! With a single precise pulse train that times the *assembly* of new sine wave *samples*, the DDS can output 2^{N-1} discrete frequencies (where N is the DDS resolution in bits). These frequencies range from dc to one-half the input clock frequency at intervals of $1/2^N$.

The DDS Clock Generator

The DDS output is a sampled sine wave containing many extraneous frequency components that will create jitter if used "as is". The amount of jitter resulting from an unfiltered sampled sine wave is equal to 1 input clock cycle. If a clock cycle is 5.7ns (175 MHz) then that much jitter will be observed from an accumulation of adjacent cycles of the DDS output signal. Figure 1 shows an unfiltered sampled sine wave from a DDS being clocked at 175 MHz. Only about 3 samples per cycle are being synthesized; however, the cycle-to-cycle samples are different as is evident by the change in voltage levels of the samples as they progress from left to right. This waveform represents about 56 MHz. When this signal is routed to a comparator with a fixed zero-crossing threshold, the 1 clock period jitter becomes visible with the scope in the infinite persistence mode. Incidentally, the jitter magnitude is the same if only the MSB of the 10-bit input code to the DAC were to be examined.

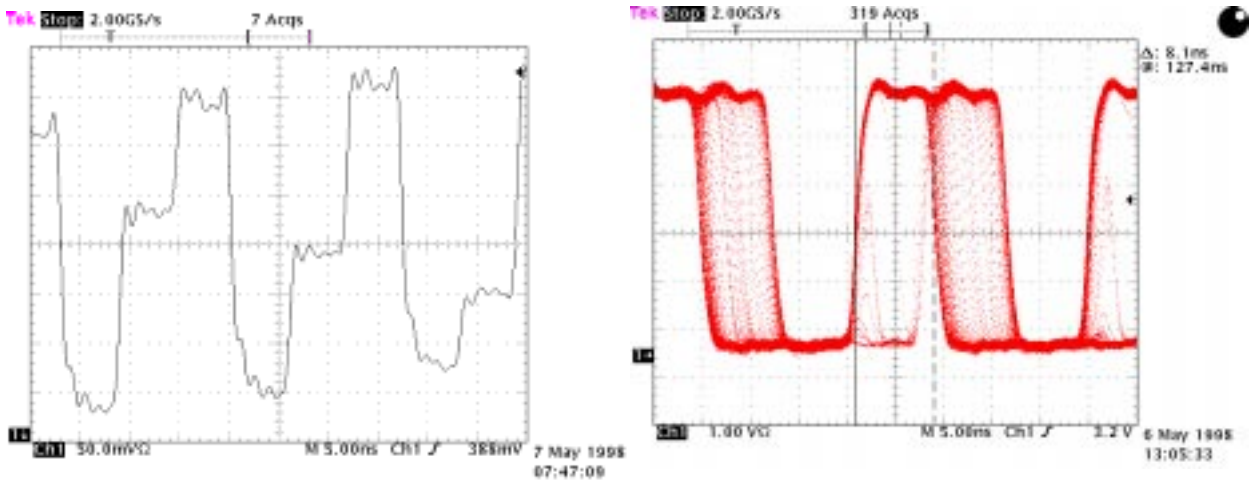


Figure 7-1. "Raw" DAC Output and Corresponding 1 Clock-period Jitter from a Comparator

Six nanoseconds of clock jitter are unsuitable for most clock applications. By low-pass or band-pass filtering the DDS output signal, many of the extraneous signals can be removed from the

DDS output and a nearly pure sine wave is extracted at the filter's output. When the filtered signal is presented to the comparator, edge jitter of the "squared-up" logic output signal reduces from 1 clock period to approximately 250 picoseconds peak-to-peak (including the jitter of the measurement instrument). Filtering can reduce the jitter to a certain level and thereafter, further filtering is ineffective due to the inherent jitter associated with the comparator being used. Effective filtering can be achieved inexpensively with a low pass filter that reduces spurious components to a level at least -50 dB (preferably more) relative to the fundamental signal. For the AD985X DDS products, the on-chip comparator has an inherent edge jitter of approximately 80 picoseconds peak-to-peak. This indicates that with better filtering the clock signal jitter can be reduced even further. Figure 7-2 shows the effect of filtering the sampled DAC output signal seen in Figure 7-1 with a 7th order elliptic low pass filter with a cutoff (-3dB) frequency of approximately 65 MHz.

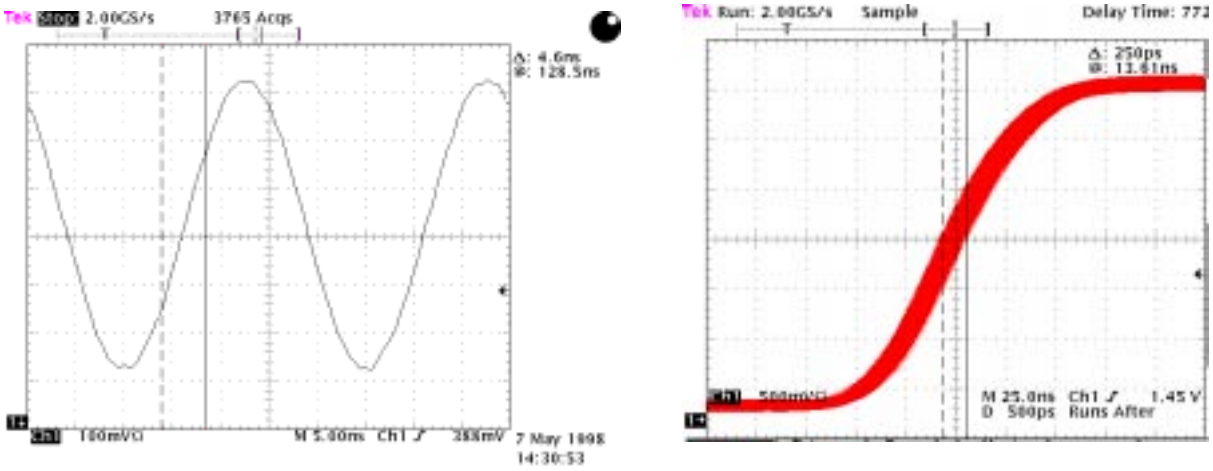


Figure 7-2. Filtered DAC Output and Corresponding Comparator Edge Jitter

Figure 7-3 below shows a frequency domain view of the DDS/DAC output before and after filtering with a simple 65 MHz 7th order elliptic low pass filter. The low pass filter does nothing to remove *aliased* harmonics of the fundamental that fall within the legitimate passband of the DDS output. For this reason, band-pass filtering would be a better choice when only a narrow segment of the DDS passband is needed.

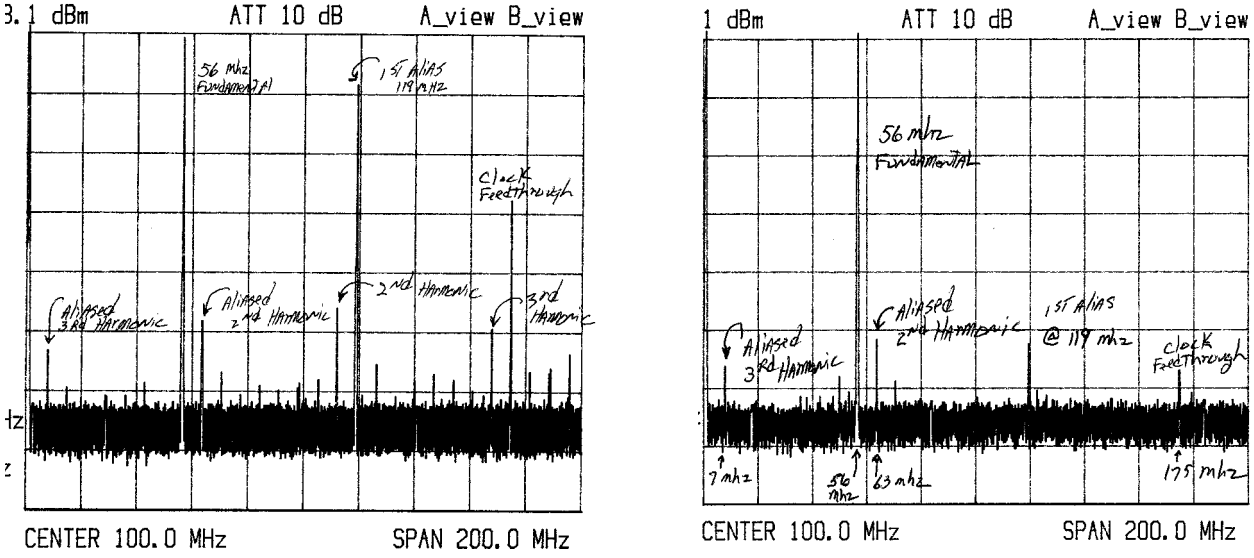


Figure 7-3: DDS/DAC Output Spectrum Before and After 65 MHz Elliptic LPF

Shown in Figure 7-4 below is the schematic diagram of the elliptic low pass filter used in a typical clock generator application. The input and output impedance of the filter has been designed for 200-Ohms to allow the 10 ma current output of the DDS/DAC to develop a 1-volt p-p signal at the output of the filter. The large output signal increases the signal slew rate and overdrive at the comparator switching threshold which reduces jitter caused by internal input noise. The 200-Ohm impedance also makes the filter more susceptible to component tolerance error, output impedance mismatch and complicates filter examination with customary 50-Ohm instruments. Figure 5 shows the swept frequency response of the filter using the tracking generator of a spectrum analyzer.

A 50% duty cycle at the comparator output is maintained by the averaging circuit composed of R4, R5 and C1. This circuit simply combines the unfiltered Iout and IoutB complementary DAC signals which should be of equal p-p amplitude and provides low pass filtering. The result is a dc voltage that equals the center point of the sampled sine wave. This voltage is used as the comparator threshold at the inverting input to the comparator. This circuit tracks amplitude variations and compensates the dc level at the comparator threshold input to maintain the 50% output duty cycle.

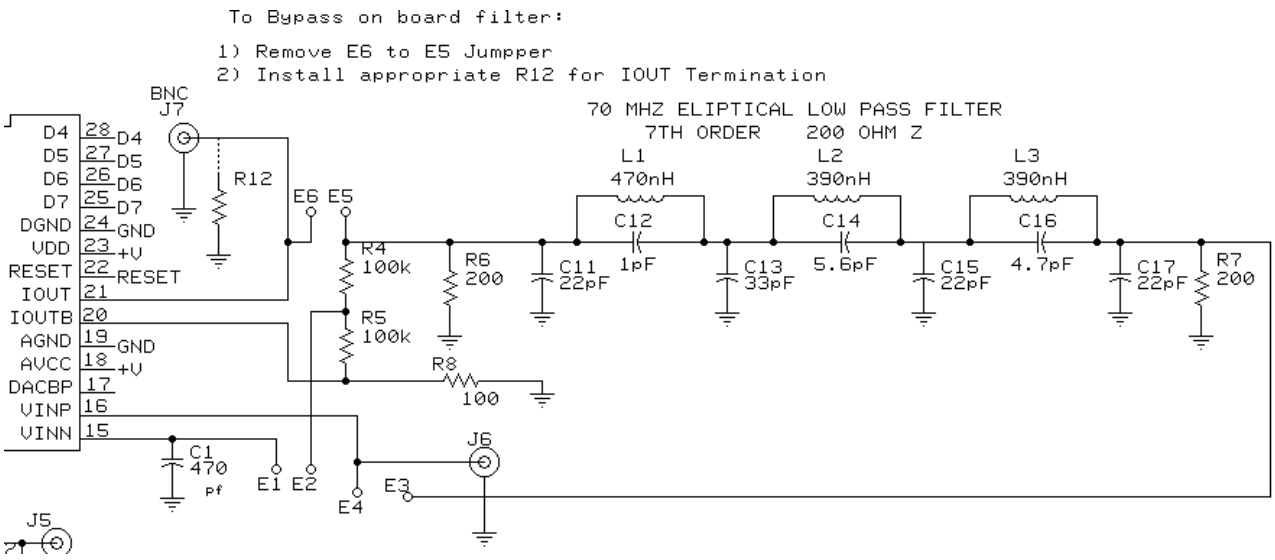


Figure 7-4. Elliptical Filter Design for Clock Generator Application

The circuit shown in Figure 7-4 is the lowpass filter and threshold voltage averaging circuit for the DDS Clock Generator application. Operation requires the following connections to be made. E5 to E6, E3 to E4, E1 to E2.

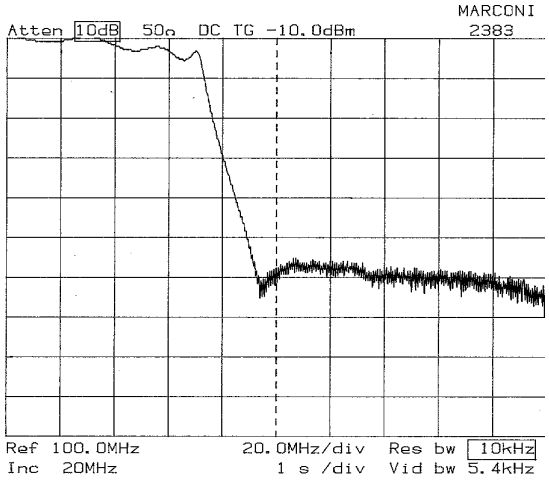


Figure 7-5. Frequency Sweep Plot of Lowpass Filter

The frequency sweep of the above filter shows the approximately –60dB stopband beginning only 25 MHz beyond the cutoff frequency approximately 70 MHz passband. (Vertical scale = 10 dB per division).

Section 8. Replacing or Integrating PLL's with DDS solutions

By Rick Cushing, Applications Engineer, Analog Devices, Inc.

DDS vs Standard PLL

PLL (phase-locked loop) frequency synthesizers are long-time favorites with designers who need stable, *programmable* high & low frequencies, and high quality signal sources or clocks. They are well understood, widely available, and inexpensive. What can a DDS do that a PLL can't?

- Extremely fast frequency changes make a DDS *thousands* of times more agile than a PLL. This makes DDS a natural choice for frequency-hopping and spread-spectrum.
- Frequency resolution is extraordinary! Up to one-millionth of a Hertz
- Fundamental output frequency span > 40 octaves (.000001 Hz to 150 MHz)
- Effortless ultra high-speed digital phase modulation (PSK) and FSK
- Perfect, exactly repeatable synchronization of multiple DDS's (allowing quadrature and other phase offset relationships to be easily accomplished)

Applications requiring any of the above traits should evaluate DDS as a possible solution. As an example, consider *dielectrophoresis* This phenomenon is utilized in micro-biology studies to separate, move and rotate individual cells or bacteria in a polarized medium using non-uniform traveling fields, typically under a microscope. The traveling waves are emitted from micro-electrodes that are excited by synchronized signals from two DDS's. Rotation and movement of particles is accomplished by connecting the synchronized signals of relative differing phases to successive electrodes (0°, 90°, 180° , 270°, etc.) which in turn generate the traveling field in which the particles move. Differing particles are affected differently by various wavelength signals, and as such, it is desirable to generate signals over a wide frequency range. DDS, by virtue of its extremely wide output frequency span, phase offset capability and precise synchronization, is an ideal vehicle to generate the synchronized signals from 1 kHz to 50 MHz typically used in this technique.

One major difference between a PLL and a DDS is the PLL's ability to lock its output to the input phase of a reference clock. A standard PLL can easily lock its VCO to a 10 MHz input signal and provided a phase locked 20 MHz output signal. The DDS can get extremely close to the 20 MHz output frequency but requires an internal clock speed that is at least twice that of the output frequency. A DDS with a 6× multiplier will synchronize with the 10 MHz master clock and internally clock at a 60 MHz rate that will (with 32-bit resolution) output a 19.9999999954 MHz or 20.000000009 MHz signal, but exactly 20 MHz can not be achieved. In fact, the *only* time the DDS output is an *exact integer division* of its system clocking frequency is when the division factor is a power of two, 2^N. Only then will the input clock and the output frequency be "phase-locked" or synchronized.

The above PLL example dramatically reverses itself when attempting to construct a PLL to output exactly 19.9999999954 MHz for a 10 MHz input signal! Imagine trying to digitally

increment or decrement a PLL in sub-Hz steps. So it can be seen that there are optimum applications for DDS's just as there are for PLL's.

A DDS can be equipped with a tunable reference clock oscillator that will allow it to perform like the VCO in a PLL. Exact frequency tuning is accomplished by first setting the tuning word *close* to the desired output frequency. The DDS reference clock frequency is tuned (without altering the frequency tuning word) until the output frequency exactly matches the desired frequency. This requires the DDS reference oscillator to be somewhat tunable while retaining high-Q characteristics such as low phase noise and frequency stability (such as in a VCXO). The 6× clock multiplier (of an AD9851 DDS) comes in handy because changes in the reference oscillator frequency will be multiplied by 6, giving considerably more tuning range to a VCXO (voltage controlled crystal oscillator). Figure 8-1 shows a partial block diagram of a practical system.

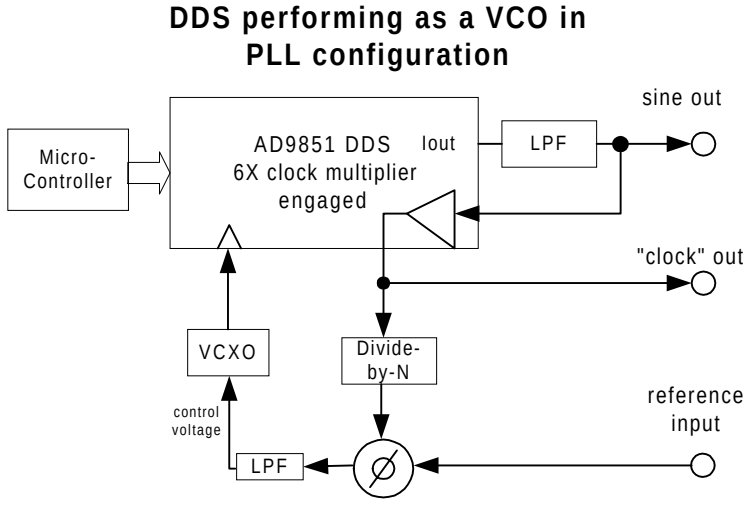


Figure 8-1. DDS Combined with PLL

The system in Figure 8-1 has very limited applications and is limited to output frequencies of approximately 40% of the system clock. The primary DDS advantage over a standard VCO in this configuration is its thirty-octave range of operation (.04 Hz to 70 MHz). Most of the other desirable DDS traits are lost in this configuration due to the presence of the "divide-by-N" stage and filters that take time to settle.

Integrating DDS with PLL's for Higher Output Frequencies

Another interesting application that combines the extreme frequency resolution trait of a DDS with the high frequency attribute of a PLL is seen in Figure 8-2. Here, the DDS acts as a fractional "Divide-by-N" stage within the feedback loop of a PLL. This gives the PLL sub-Hertz resolution at output frequencies up to the system clock limit of the DDS – typically 50 to 300 MHz. Any DDS spurs within the PLL bandwidth will be multiplied (gained-up) by 20 Log (Fout/Fref) .

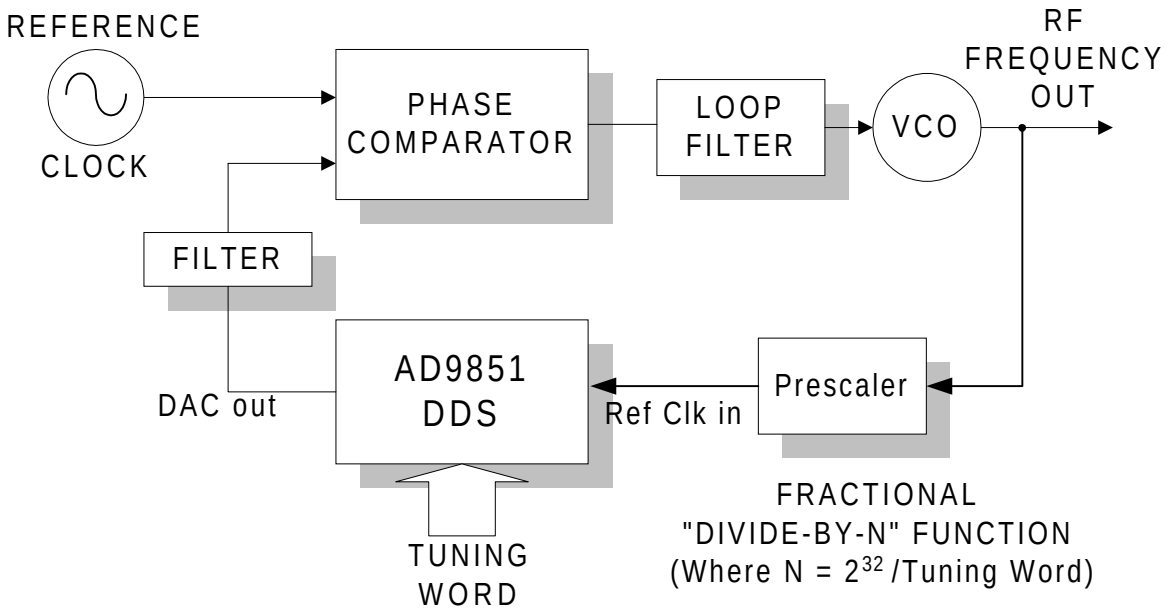


Figure 8-2. Fractional "Divide-by-N" Allows Sub-Hertz Frequency Resolution at VHF

Figure 8-3 below shows the DDS performing the local oscillator function in an analog mixer within the PLL loop. This gives the PLL greatly increased frequency resolution at the VCO output frequency while retaining the high signal quality traits of the fixed PLL reference oscillator. The VCO could be operating at HF, VHF, UHF or microwave frequencies. The "divide-by-N" stage could be a fixed divider or a course, programmable divider that selects a particular frequency range while the DDS provides the fine frequency resolution within that range. DDS spurs will be reduced in the "divide-by-N" stage but augmented by the same amount in the frequency multiplication process. Therefore, any DDS spurs within the PLL loop bandwidth will be passed along *unchanged* to the output. Spur reduction and augmentation follow the standard processing gain or loss of 20Log Fout/Fref. Simply stated, flaws (spurs, jitter, phase noise) in the reference signal that are within the loop passband of a PLL will be multiplied along with the frequency. Greater PLL multiplication factors result in greater reference signal degradation at the output. Practical output limitations are imposed by the need to adequately filter the mixer output.

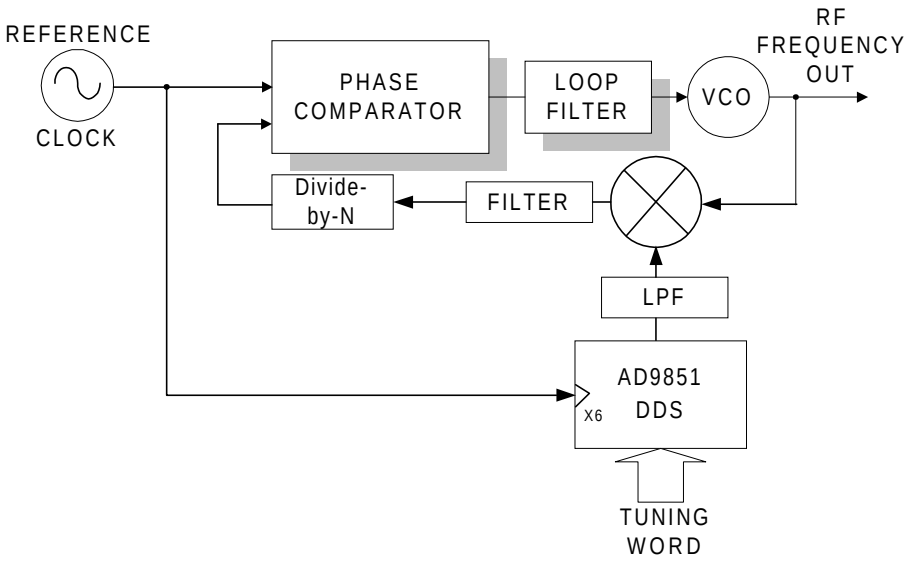


Figure 8-3. DDS as Local Oscillator in Mixer/PLL at UHF or Microwave Output Frequencies

Practical Application of a DDS Driving a PLL at 900 MHz

Figure 8-4 shows a 14.0 MHz output signal of an AD9851 DDS that is being input as a reference to a National Semiconductor LMX1501A, 900 MHz PLL evaluation board. Figure 8-5 shows the impact of PLL frequency multiplication on the phase noise and spurs of the reference signal after multiplication to 896 MHz by the LMX1501A. In this instance, the 14.0 MHz signal from the DDS was first divided by a factor of 64 in the LMX1501A, yielding a phase noise improvement of 36 dB. The resulting signal was then multiplied by a factor of 4096, yielding a phase noise degradation of 72 dB. Overall phase noise is theoretically degraded by approximately 36 dB within the PLL loop bandwidth. Spurs are located beyond the PLL loop bandwidth and are only slightly augmented.

Note: The LMX1501A evaluation board was being operated at 3.1 volts and the DDS reference input signal is approximately 10 dB below the recommended input level. These two factors were both seen to have an adverse impact on the overall phase noise of the PLL output. The examples shown below are meant to demonstrate the impact of PLL multiplication on reference input phase noise and spur levels and do not represent an optimized system.

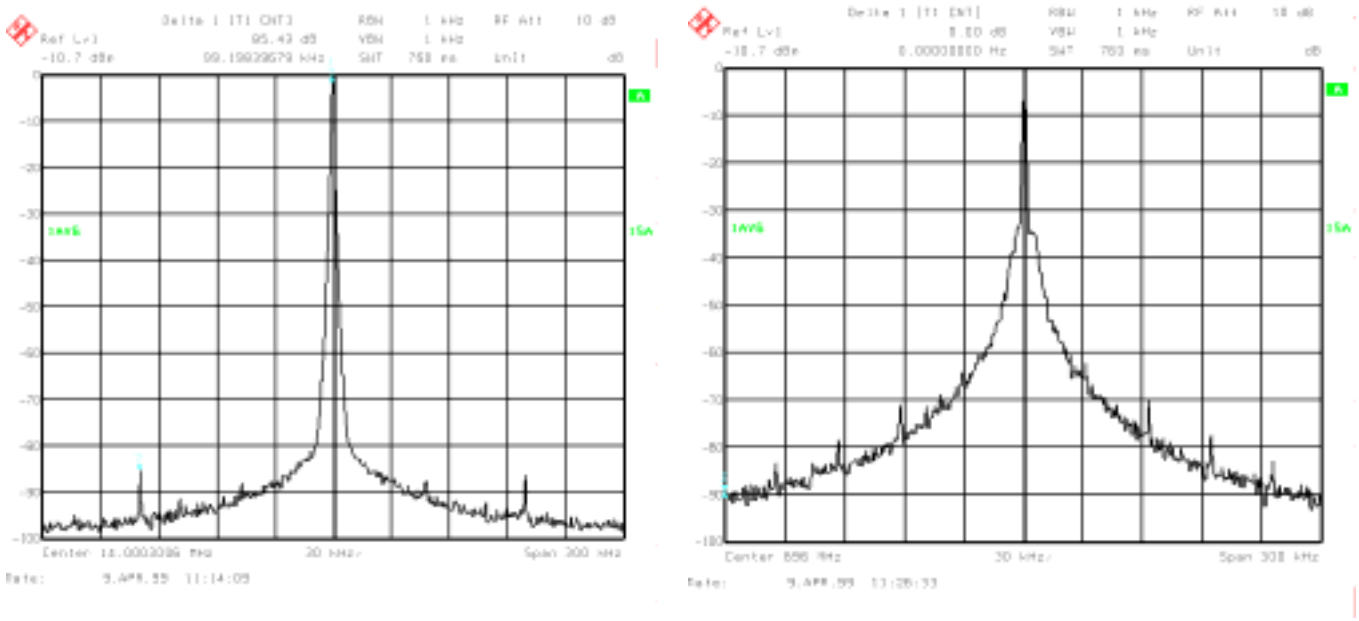


Figure 8-4. 14 MHz DDS Reference Signal to PLL

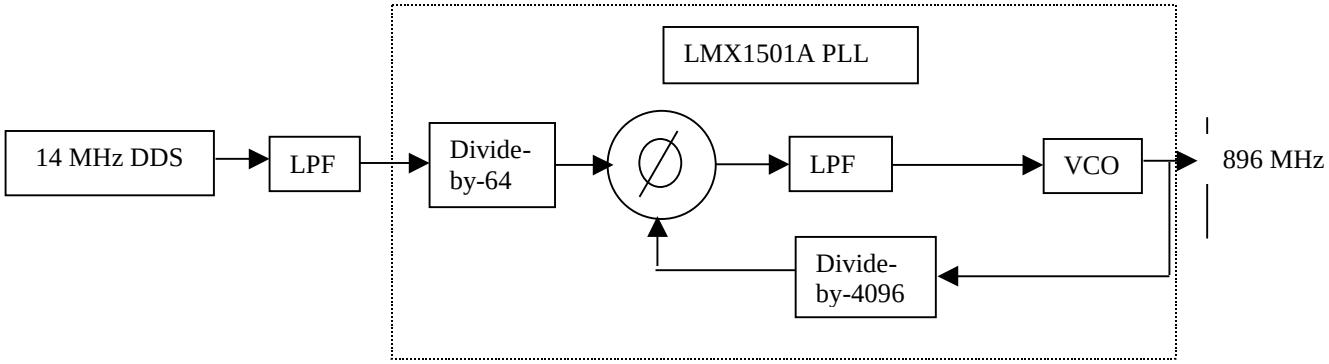


Figure 8-5: 896 MHz PLL output

Figures 8-6 and 8-7, show a DDS output with spurs and the PLL's response, both within and outside of the loop bandwidth. The same PLL conditions exist as explained above. The PLL loop bandwidth at this particular divide-by-N appears to extend to approximately 80 kHz as seen by the PLL's diminishing response to spurs beyond that cutoff frequency.

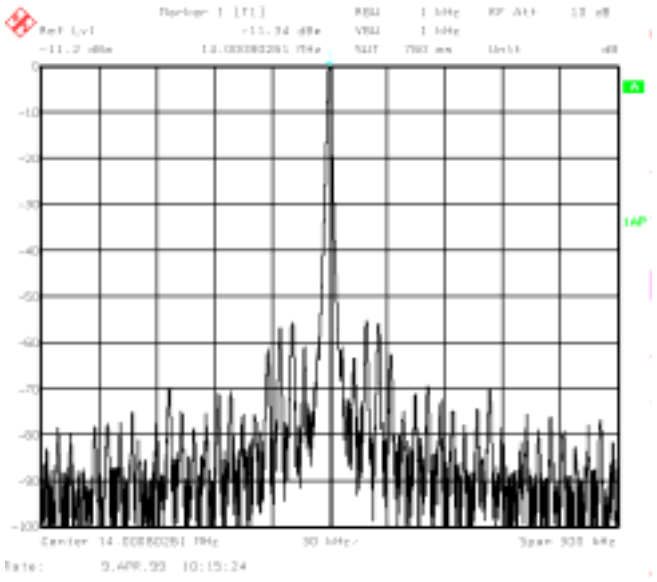


Figure 8-6. 14 MHz DDS Reference Signal with Spurs

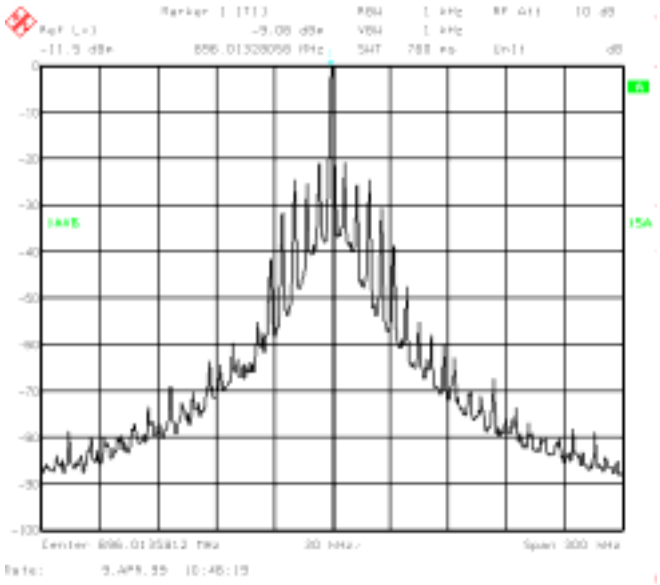
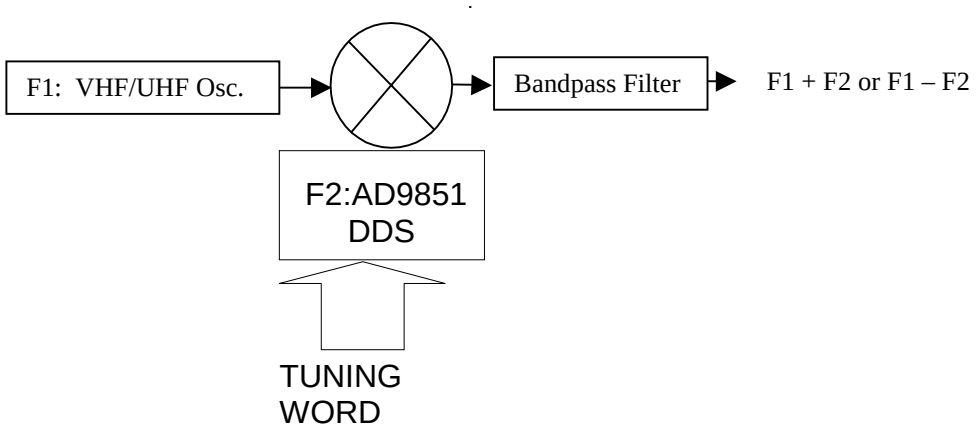


Figure 8-7.896 MHz LMX1501A PLL Output

Comparing Figure 8-5 with Figure 8-7, designers can see how important it is to keep spur levels at the PLL reference inputs as low as possible or to keep spurs out of the PLL loop bandwidth. A seemingly insignificant reference spur within the loop bandwidth can become a huge component in the VCO output spectrum. Another strategy to control spurs, such as in Figure 8-3, is to place the known spur source ahead of a divide-by-N stage to reduce spurs levels by 20 Log (Fout/Fin). If the divide-by-N stage had been placed at the mixer input instead of the output, then any DDS spurs within the loop bandwidth would have been passed undiminished and subjected to loop gain augmentation.

The need for low output phase noise is an equally important consideration in any noise sensitive system and unfortunately, this is a PLL weakness that requires elaborate avoidance measures to achieve good performance. The inherent phase noise improvement (of the reference signal relative to the output) in DDS is a quality that makes its use nearly ideal as a PLL replacement at lower output frequencies. And, at VHF and higher output frequencies, the DDS can function as a LO in a VHF/UHF mixer (see Figure 8-8) that avoids the multiplication pitfalls of the PLL and maintains the high quality attributes of the DDS signal. A practical output limitation is imposed by the need to adequately filter the mixer output.

Figure 8-8. VHF/UHF Output Maintains Phase Noise and Spur Performance of Low Frequency DDS Output and Avoids PLL Entirely.



Section 9. Basic Digital Modulator Theory

By Ken Gentile, Systems Engineer, Analog Devices, Inc.

To understand digital modulators it may be insightful to first review some of the basic concepts of signals. With a basic understanding of signals, the concept of baseband and bandpass signals can be addressed. Which then leads to the concept of modulation in continuous time (the analog world). Once continuous time modulation is understood, the step to digital modulation is relatively simple.

Signals

The variety of signal types and classes is broad and a discussion of all of them is not necessary to understand the concepts of digital modulation. However, one class of signals is very important: **periodic complex exponentials**. Periodic complex exponentials have the form:

$$x(t) = \beta(t)e^{j\omega t}$$

where $\beta(t)$ is a function of time and may be either real or complex. It should be noted that $\beta(t)$ is not restricted to a function of time; it may be a constant. Also, ω is the radian frequency of the periodic signal. The natural frequency, f, is related to the radian frequency by $\omega = 2\pi f$. $\beta(t)$ is known as the **envelope** of the signal. A plot of x(t) is shown in Figure 9.1, where $\beta(t) = \sin(2\pi f_a t)$, f_a = 1MHz, $\omega = 2\pi f_c t$, and f_c = 10MHz. Since x(t) is complex, only the real part of x(t) is plotted (the dashed lines indicate the envelope produced by $\beta(t)$).

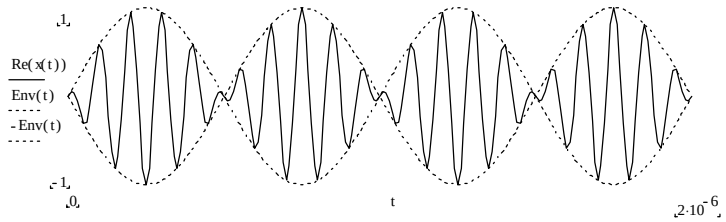


Figure 9.1. Periodic Complex Exponential

An alternative form of x(t) can be obtained by invoking Euler's identity:

$$x(t) = \beta(t)[cos(\omega t) + jsin(\omega t)]$$

This form tends to be a little more intuitive because its complex nature is clearly indicated by its real and imaginary components; a quality not so obvious in the complex exponential form.

A special subset of periodic complex exponentials are **sinusoidal signals**. A sinusoidal signal has the form:

$$x(t) = Acos(\omega t)$$

Again, using Euler's identity, x(t) can be written as:

$$x(t) = \tfrac{1}{2}A(e^{j\omega t} + e^{-j\omega t})$$

Note from the above equation that a sinusoidal signal contains both positive and negative frequency components (each contributing half of A). This is an important fact to remember. We are generally not accustomed to negative frequency, but it is nontheless mathematically valid. Figure 9.2 is a frequency vs. magnitude plot of a sinusoidal signal.

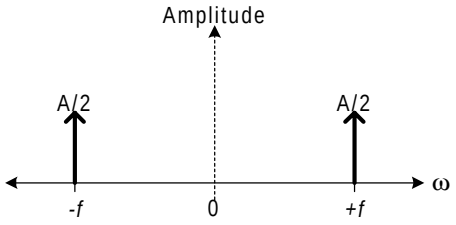


Figure 9.2. Positive and Negative Frequency

Interestingly, a sinusoidal signal can also be represented by extracting the real part of a periodic complex exponential:

$$\Re e\{Ae^{j\omega t}\} = \Re e\{A\cos(\omega t) + jA\sin(\omega t)\} = A\cos(\omega t)$$

Baseband Signals

A baseband signal is one that has a frequency spectrum which begins at 0Hz (DC) extending up to some maximum frequency. Though a baseband signal includes 0Hz, the magnitude at 0Hz may be 0 (i.e., no DC component). Furthermore, though a baseband signal usually extends up to some maximum frequency, an upper frequency limit is not a requirement. A baseband signal may extend to infinity. Most of the time, however, baseband signals are bandlimited and have some upper frequency bound, f_{max}. A bandlimited baseband signal can be represented graphically as a plot of signal amplitude vs. frequency. This is commonly referred to as a **spectrum**, or spectral plot. Figure 9.3 shows an example of a baseband spectrum. It's maximum magnitude, A, occurs at DC and its upper frequency bound is f_{max}.

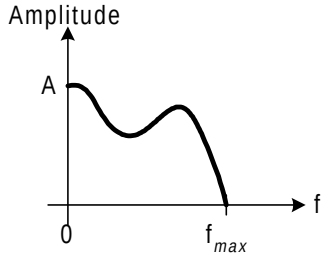


Figure 9.3. Single-Sided Bandlimited Baseband Spectrum

Note that only the positive portion of the frequency axis is shown. This type of spectrum is known as a single-sided spectrum. A more useful representation of a spectrum is one which includes the negative portion of the frequency axis is included. Such a spectrum is known as a double-sided spectrum. Figure 9.4 shows a double-sided representation of the spectrum of Figure 9.3.

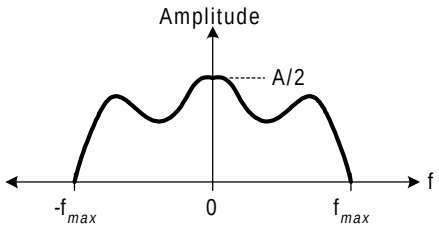


Figure 9.4. Double-Sided Bandlimited Baseband Spectrum

Note that the signal amplitude is only 50% of the amplitude shown in the single-sided spectrum. This is because the negative frequency components are now being accounted for. In the single-sided spectrum, the energy of the negative frequency components are simply added to the positive frequency components yielding an amplitude that is twice that of the double-sided spectrum. Note, also, that the righthand side of the spectrum (+f) is mirrored about the 0Hz line creating the lefthand portion of the spectrum (-f). Baseband spectra like the one above (which contain horizontal symmetry) represent **real** baseband signals.

There are also baseband signals that lack horizontal symmetry about the f = 0 line. These are **complex** baseband signals. An example of a complex baseband spectrum is shown below in Figure 9.5.

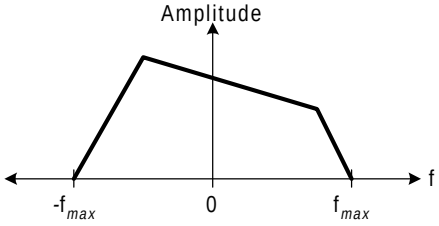


Figure 9.5. Complex Baseband Spectrum

Note that the left side and right side of the spectrum are not mirror images. This lack of symmetry is the earmark of a complex spectrum. A complex baseband spectrum, since it is complex, can not be expressed as a real signal. It can, however, be expressed as the complex sum of two real signals a(t) and b(t) as follows:

$$x(t) = a(t) + jb(t)$$

It turns out that it is not possible to propagate (transmit) a complex baseband spectrum in the real world. Only real signals can be propagated. However, a complex baseband signal can be transformed into a *real* bandpass signal through a process known as frequency translation or modulation (see the next section). It is interesting to note that modulation can transform a complex baseband signal (which can not be transmitted) into a bandpass signal (which is a real and can be transmitted). This concept is fundamental to all forms of signal transmission in use today (e.g., radio, television, etc.).

Bandpass Signals

A bandpass signal can be thought of as a bandlimited baseband signal centered on some frequency, f_c and it's negative, $-f_c$. Recall that a baseband signal is centered on f = 0. Bandpass signals, on the other hand, are centered on some non-zero frequency, $\pm f_c$, such that $|f_c| > 2f_{max}$. The value, $2f_{max}$, is the **bandwidth** (BW) of the bandpass signal. This is shown pictorially in Figure 9.6. It should be noted that there are two types of bandpass signals; those with a symmetric baseband spectrum and those with a nonsymmetric (or *quadrature*) baseband spectrum (depicted in Figure 9.6(a) and (b), respectively).

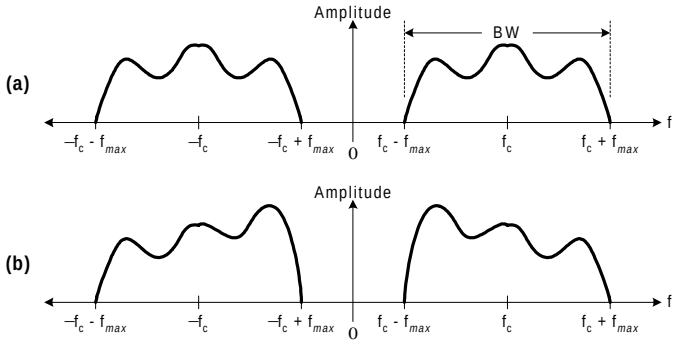


Figure 9.6. Bandpass Spectra

Mathematically, a bandpass signal can be represented in one of two forms. For case (a) we have:

$$x(t) = g(t)cos(\omega_c t)$$

where g(t) is the baseband signal and ω_c is radian frequency (radian frequency is related to natural frequency by the relationship $\omega = 2\pi f$). Note that multiplication of the baseband signal by $\cos(\omega_c t)$ translates the baseband signal so that it is centered on $\pm f_c$.

Alternatively, two baseband signals, $g_1(t)$ and $g_2(t)$ can be combined in quadrature fashion to produce the spectrum shown in case (b). Mathematically, this is represented by:

$$x(t) = g_1(t)cos(\omega_c t) + g_2(t)sin(\omega_c t)$$

Again, the important thing to note is that the bandpass signal is centered on $\pm f_c$. Furthermore, it should be mentioned that for quadrature bandpass signals $g_1(t)$ and $g_2(t)$ do not necessarily have to be different baseband signals. There is nothing to restrict $g_1(t)$ from being equal to $g_2(t)$. In which case, the same baseband signal is combined in quadrature fashion to create a bandpass signal.

Modulation

The concept of bandpass signals leads directly to the concept of modulation. In fact, the translation of a spectrum from one center frequency to another is, by definition, *modulation*. The bandpass equations in the previous section indicate that multiplication of a signal, g(t), by a sinusoid (of frequency ω_c) is all that is necessary to perform the modulation function. The only difference between the concept of a bandpass signal and modulation, is that with modulation it is

not necessary to restrict g(t) to a baseband signal. g(t) can, in fact, be another bandpass signal. In which case, the bandpass signal, g(t), is translated in frequency by $\pm f_c$. It is this very property of modulation which enables the process of **demodulation**. Demodulation is accomplished by multiplying a bandpass signal centered on f_c by $cos(\omega_c)$. This shifts the bandpass signal which was centered on f_c to a baseband signal centered on 0Hz and a bandpass signal centered on $2f_c$. All that is required to complete the transformation from bandpass to baseband is to filter out the $2f_c$-centered component of the spectrum.

Figure 9.7 below shows a functional block diagram of the two basic modulation structures. Figure 9.7(a) demonstrates sinusoidal modulation, while (b) demonstrates quadrature modulation. There are, in fact, variations on these two themes which produce specialized forms of modulation. These include present- and suppressed-carrier modulation as well as double- and single-sideband modulation.

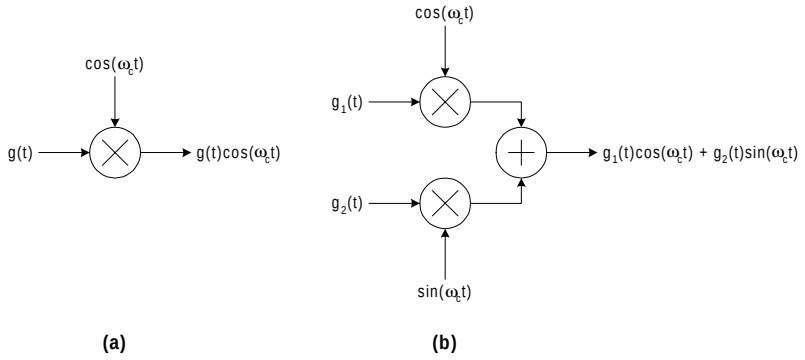


Figure 9.7. Basic Modulation Structures

The preceding sections set the stage for an understanding of **digital modulation**. The reader should be familiar with the basics of sampled systems in order to fully understand digital modulation. This chapter is written with the assumption that the reader has fundamental knowledge of sampled systems, especially with respect to the implications of the Nyquist theorem.

Digital modulation is the discrete-time counterpart to the continuous-time modulation concepts discussed above. Instead of dealing with an analog waveform, x(t), we are dealing with instantaneous samples of an analog waveform, x(n). It is implied that n, an integer index, bears a one-to-one correspondence with sampling time instants. That is, if T represents the time interval between successive samples, then nT represents the time instants at which samples are taken. The similarity between continuous- and discrete-time signals becomes obvious when their forms are written together. For example, consider a sinusoidal signal:

$$x(t) = A\cos(\omega t) \quad \textit{continuous-time}$$
$$x(n) = A\cos(\omega nT) \quad \textit{discrete-time}$$

The main difference in the discrete-time signal is that there are certain restrictions on ω and T as a result of the Nyquist theorem. Specifically, T must be less than π/ω (remember $\omega = 2\pi f$ and T is the sampling period). Since x(n) is a series of instantaneous samples of x(t), then x(n) can be

represented as a series of numbers, where each number is the instantaneous value of x(t) at the instants, nT.

The importance of this idea is paramount in understanding digital modulators. In the analog world modulation is achieved by multiplying together continuous-time waveforms using specialized analog circuits. However, in the digital world it is possible to perform modulation by simply manipulating sequences of numbers. A purely numeric operation.

The modulation structures for continuous-time signals can be adapted for digital modulators. This is shown in Figure 9.8.

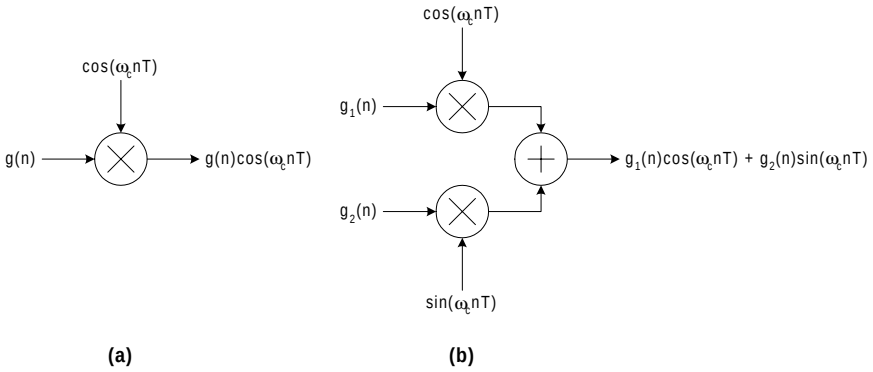


Figure 9.8. Basic Digital Modulation Structures

Here, g(n), $g_1(n)$, $g_2(n)$, $\sin(\omega_c nT)$ and $\cos(\omega_c nT)$ are sequences of numbers. The multipliers and adders are logic elements (digital multipliers and adders). Their complexity is a function of the number of bits used to represent the samples of the input waveforms. This is not much of an issue in theory. However, when implemented in hardware, the number of circuit elements can grow very quickly. For example, if the digital waveforms are represented by 8-bit numbers, then multipliers and adders capable of handling 8-bit words are required. If, on the other hand, the digital waveforms are represented by IEEE double-precision floating point numbers (64-bits), then the multipliers and adders become very large structures.

It is in the environment of digital modulators that DDS technology becomes very attractive. This is because a DDS directly generates the series of numbers that represent samples of a sine and/or cosine waveform. DDS-based digital modulator structures are shown in Figure 9.9.

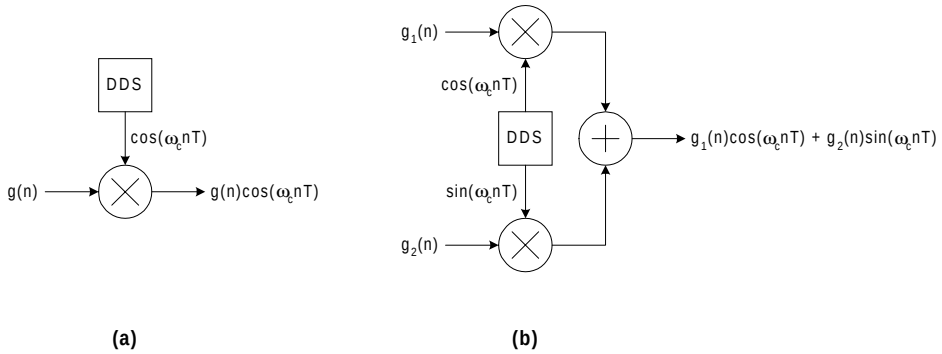


Figure 9.9. Basic DDS Modulation Structures

System Architecture and Requirements

The basic DDS modulation structures described in the previous section are very simplistic. There are a number of elements not shown that are required to actually make a digital modulator work. The most critical element is a clock source. A DDS can only generate samples if it is driven by a sample clock. Thus, without a system clock source a digital modulator is completely nonfunctional.

Also, since the digital modulator shown above amounts to nothing more than a "number cruncher", its output (a continuous stream of numbers) is of little use in a real application. As with any sampled system, there is the prevailing requirement that the digital number stream be converted back to a continuous-time waveform. This will require the second major element of a digital modulator, a digital-to-analog converter (**DAC**).

A more complete DDS modulator is shown in Figure 9.10. In order to keep things simple, only the sinusoidal modulator form is shown. The extension to a quadrature modulator structure is trivial.

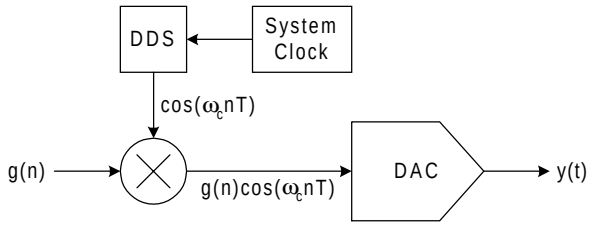


Figure 9.10. DDS Modulator

At first glance, the DDS modulator appears to be quite simple. However, there is a subtle requirement that makes digital modulation a bit more difficult to implement. The requirement is that g(n) must consist of the samples of a signal sampled at the same frequency as the DDS sample rate. Otherwise, the multiplier stage is multiplying values that have been sampled at completely different instants.

To help illustrate the point, suppose

$$g(n) = cos[2\pi(1kHz)nT_1]$$

where T_1 is 0.00025 (0.25ms). Thus, g(n) can be described as a 1kHz signal sampled at 4kHz ($1/T_1$). Suppose, also, that the DDS output is

$$DDS = cos[2\pi(3kHz)nT_2]$$

where T_2 is 0.0001 (0.1ms). This means that the DDS output is a 3kHz signal sampled at 10kHz ($1/T_2$). In the DDS modulator, the value of n (the sample index) for the multiplier is the same for both of the inputs as well as the output. So, for a specific value of n, say n=10 (i.e., the 10^{th} sample) the time index for the DDS is nT_2, which is 0.001(1ms). Clearly, $nT_1 \neq nT_2$ (2.5ms ≠ 1ms). Thus, at time index n=10, DDS time is 1ms while g(n) time is 2.5ms. The bottom line is

that the product, g(n)*cos*(ω_cnT), which is the output of the multiplier, is not at all what it is expected to be, because the time reference for g(n) is not the same as that of *cos*(ω_cnT).

The "equal sample rate" requirement is the primary design consideration in a digital modulator. If, in a DDS modulator system, the source of the g(n) signal operates at a sample rate other than the DDS clock, then steps must be taken to correct the sample rate discrepancy. The DDS modulator design then becomes an exercise in multirate digital signal processing (**DSP**). Multirate DSP requires an understanding of the techniques involved in what is known as interpolation and decimation. However, interpolation and decimation require some basic knowledge of digital filters. These topics are covered in the following sections.

Digital Filters

Digital filters are the discrete-time counterpart to continuous-time analog filters. Some groundwork was laid on analog filters in Chapter 4. The analog filter material presented in Chapter 4 was purposely restricted to lowpass filters, because the topic was antialias filtering (which is inherently a lowpass application). However, analog filters can be designed with highpass, bandpass, and bandstop characteristics, as well.

The same is true for digital filters. Unlike an analog filter, however, a digital filter is not made up of physical components (capacitors, resistors, inductors, etc.). It is a logic device that simply performs as a number cruncher. Subsequently, a digital filter does not suffer from the component drift problems that can plague analog filters. Once the digital filter is designed, its operation is predictable and reproducible. The second feature is particularly attractive when two signal paths require identical filtering.

There are two basic classes of digital filters; one is the **FIR** (Finite Impulse Response) filter and the other is the **IIR** (Infinite Impulse Response) filter. From the point of view of filter design, the FIR is the simpler of the two to work. However, from the point of view of hardware requirements, the IIR has an advantage. It usually requires much less circuitry than an FIR with the same basic response. Unfortunately, the IIR has the potential to become unstable under certain conditions. This property often excludes it from being designed into systems for which the input signal is not well defined.

FIR Filters

Fundamentally, an FIR is a very simply structure. It is nothing more than a chain of delay, multiply, and add stages. Each stage consists of an input and output data path and a fixed coefficient (a number which serves as one of the multiplicands in the multiplier section). A single stage is often referred to as a **tap**. Figure 9.11 below shows a simple 2-tap FIR (the input signal to the FIR filter is considered to be one of the taps).

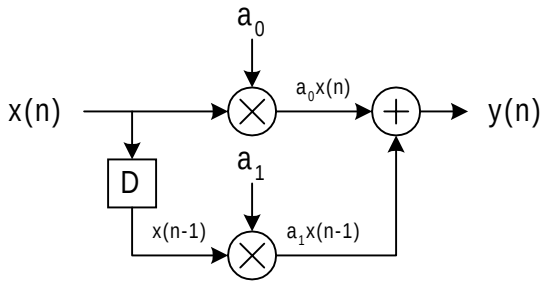


Figure 9.11. Simple FIR Filter

In this simple filter we have input, x(n), which is a number sequence that represents the sampled values of an input signal. The input signal is fed to two places. First, it is fed to a multiplier which multiplies the current sample by coefficient, a_0 (a_0 is simply a number). The product, $a_0x(n)$ is fed to one input of the output adder. The second place that the input signal is routed to is a delay stage (the "D" box). This stage simply delays x(n) by one sample. Note that the output of the delay stage is labeled, x(n-1). That is, it is the value of the previous sample. To better understand this concept consider the general expression, x(n-k), where k is an integer that represents time relative to the present. If k=0, then we have x(n), which is the value of x(n) at the present time. If k=1, then we have x(n-1), which is the the value of x(n) one sample earlier in time. If k=2, then we have x(n-2), which is the value of x(n) two samples earlier in time, etc. Returning to the figure, the output of the delay stage feeds a multiplier with coefficient a_1. So, the output of this multiplier is $a_1x(n-1)$, which is the value of x(n) one sample earlier in time multiplied by a_1. This product is fed to the other input of the output adder. Thus, the output of the FIR filter, y(n), can be represented as:

$$y(n) = a_0x(n) + a_1x(n-1)$$

This means, that at any given instant, the output of this FIR is nothing more than the sum of the current sample of x(n) and the previous sample of x(n) each multiplied by some constant value (a_0 and a_1, respectively). Does this somehow relate back to the concept of filtering a signal? Yes, it does, but in order to see the connection we must make a short jump to the world of the z-transform.

The z-transform is related to the Fourier transform in that, just as the Fourier transform allows us to exchange the time domain for the frequency domain (and vice-versa), the z-transform does exactly the same for sampled signals. In other words, the z-transform is merely a special case of the Fourier transform in that it is specifically restricted to discrete time systems (i.e., sampled systems).

By applying the z-transform to the previous equation, it can be shown that the transfer function, H(z), is given as:

$$H(z) = a_0z^0 + a_1z^{-1} = a_0 + a_1z^{-1}$$

Where $z = e^{j\omega}$, $\omega = 2\pi f/F_s$ and F_s is the sample frequency. Notice that transfer function, through the use of the z-transform, results in the conversion of x(n-1) to z^{-1}. In fact, this can be generally extended to the case, x(n-k), which is converted to z^{-k} (k is the number of unit delays).

Now let's apply some numbers to our example to make things a little more visual. Suppose we employ a sample rate of 10kHz (F_s=10kHz) and let a_0=a_1=0.5. Now, compute H(z) for different frequencies (f) and plot the magnitude of H(z) as a function of frequency:

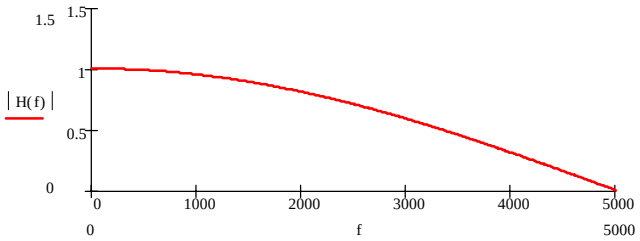


Figure 9.12. FIR Frequency Response for $a_0 = a_1 = 0.5$

Clearly, this constitutes a filter with a lowpass response. Note that the frequency axis only extends to 5kHz (½F_s) because in a sampled system the region from ½F_s to F_s is a mirror image of the region 0 to F_s (a consequence of the Nyquist limit). So, from the plot of the transfer function above, we see that at an input frequency of 3kHz this particular FIR filter allows only about 60% of the input signal to pass through. Thus, if the input sample sequence {x(n)} represents the samples of a 3kHz sinewave, then the output sample sequence {y(n)} represents the samples of a 3kHz sinewave also. However, the output signal exhibits a 40% reduction in amplitude with respect to the input signal.

The simple FIR can be extended to provide any number of taps. Figure 9.13 shows an example of how to extend the simple FIR to an N-tap FIR.

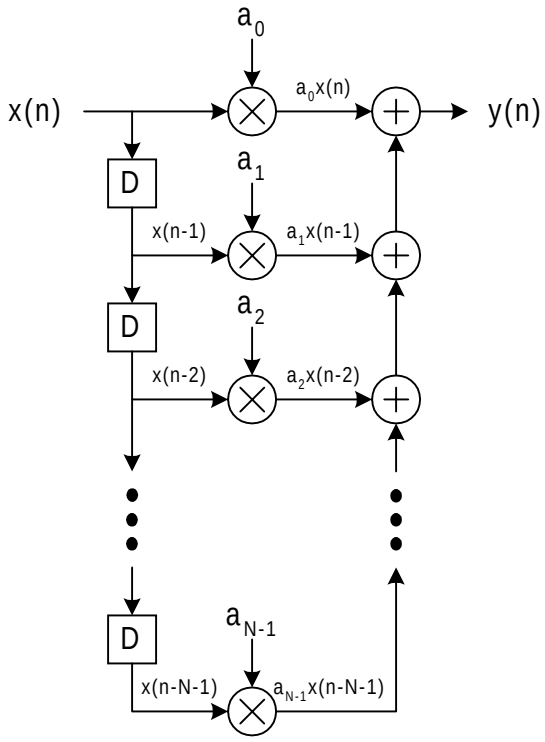


Figure 9.13. An N-tap FIR Filter

It should be mentioned that it is possible to have coefficients that are equal to 0 for a particular type of filter response. In which case, the need for the multiply and add stage for that particular coefficient is superfluous. All that is required is the unit delay stage. Also, for a coefficient of 1, a multiply is not needed. The value of the delayed sample can be passed on directly to the adder for that stage. Similarly, for a coefficient of –1, the multiply operation can be replace by a simple negation. Thus, when an FIR is implemented in hardware, coefficients that are 0, 1, or –1 help to reduce the circuit complexity by eliminating the multiplier for that stage.

An analysis of the behavior of the FIR is easy to visualize if y(n) is initially at 0 and the x(n) sequence is chosen to be 1,0,0,0,0,0… From the diagram, it should be apparent that, with each new sample instant, the single leading 1 in x(n) will propagate down the delay chain. Thus, on the first sample instant $y(n)=a_0$. On the second sample instant $y(n)=a_1$. On the third sample instant $y(n)=a_2$, etc. This process continues through the N-th sample, at which time the output of each delay stage is 0 and remains at 0 for all further sample instants. This means that y(n) becomes 0 after the N-th sample as well, and remains at 0 thereafter.

This leads to two interesting observations. First, the input of a single 1 for x(n) constitutes an impulse. So, the output of the FIR for this input constitutes the impulse response of the FIR (which is directly related to its frequency response). Notice that the impulse response only exists for N samples. Hence the name, ***Finite*** Impulse Response filter. Furthermore, the impulse response also demonstrates that an input to the FIR will require exactly N samples to propagate through the entire filter before its effect is no longer present at the output.

It should be apparent that increasing N will increase the overall delay through the FIR. This can be a problem in systems that are not delay tolerant. However, there is a distinct advantage to increasing N; it increases the sharpness of the filter response.

Examination of the above figure leads to an equation for the output, y(n), in terms of the input, x(n) for an FIR of arbitrary length, N. The result is:

$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2) + \ldots + a_{N-1}x(n-N-1)$$

Application of the z-transform leads to the transfer function, H(z), of an N-tap FIR filter.

$$H(z) = a_0 + a_1z^{-1} + a_2z^{-2} + \ldots + a_{N-1}z^{-(N-1)}$$

Earlier, it was mentioned that increasing the number of taps in an FIR increases the sharpness of the filter response. This is shown in Figure 9.14, which is a plot of a 10-tap FIR. All coefficients (a_0– a_9) are equal to 0.1. Notice how the rolloff of the filter is much sharper than the 2-tap simple FIR shown earlier. The first null appears at 1kHz for the 10-tap FIR, instead of at 5kHz for the 2-tap FIR.

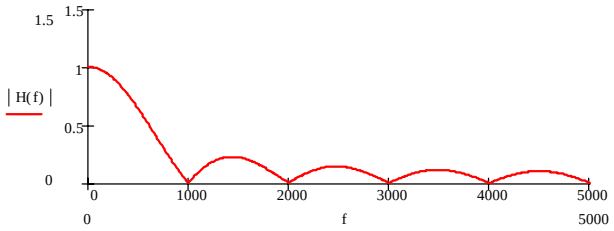


Figure 9.14. 10-Tap FIR Frequency Response for $a_{0-9} = 0.1$

It should be mentioned here, that the response of the FIR is completely determined by the coefficient values. By choosing the appropriate coefficients it is possible to design filters with just about any response; lowpass, highpass, bandpass, bandreject, allpass, and more.

IIR Filters

With an understanding of FIR filters, the jump to IIR's is fairly simple. The difference between an IIR and an FIR is feedback. An IIR filter has a feedback section with additional delay, multiply, and add stages that tap the output signal, y(n). A simple IIR structure is shown in Figure 9.15.

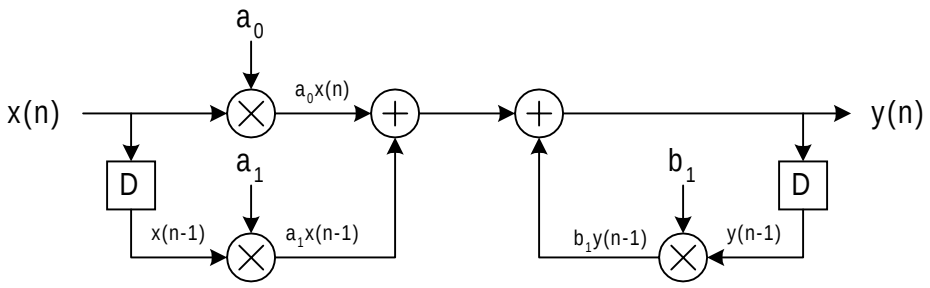


Figure 9.15. Simple IIR Filter

Notice that the left side is an exact copy of the simple 2-tap FIR. This portion of the IIR is often referred to as the **feedforward** section, because delayed samples of the input sequence are fed forward to the adder block. The right hand portion is a **feedback** section. The feedback is a delayed and scaled version of the output signal, y(n). Note that the feedback is summed with the output of the feedforward section. The existence of feedback in an IIR filter makes a dramatic difference in the behavior of the filter. As with any feedback system, stability becomes a significant issue. Improper choice of the coefficients or unanticipated behavior of the input signal can cause instability in the IIR. The result can be oscillation or severe clipping of the output signal. The stability issue may be enough to exclude the use of an IIR in certain applications.

The impulse response exercise that was done for the FIR is also a helpful tool for examining the IIR. Again, let y(n) be initially at 0 and x(n) be the sequence: 1,0,0,0…. After 2 sample instants, the left side of the IIR will have generated two impulses of height a_0 and a_1 (exactly like the simple FIR). After generating two impulses, the left side will output nothing but 0's. The right side is a different story. Every new sample instant modifies the value of y(n) by the previous value of y(n), recursively. The result is that y(n) continues to output values indefinitely, even though the input signal is no longer present. So, a single impulse at the input results in an infinitely long sequence of impulses at the output. Hence the name, ***Infinite*** Impulse Response filter.

It should be pointed out that *infinite* is an ideal concept. In a practical application, an IIR can only be implemented with a finite amount of numeric resolution. Especially, in fixed point applications, where the data path may be restricted to, say, 16-bit words. With finite numeric resolution, values very near to 0 get truncated to a value of 0. This leads to IIR performance that deviates from the ideal. This is because an ideal IIR would continue to output ever decreasing values, gradually approaching 0. However, because of the finite resolution issue, there comes a point at which small values are assigned a value of 0, eventually terminating the gradually decreasing signal to 0. This, of course, marks the end of the "infinite" impulse response.

The structure of the simple IIR leads to an expression for y(n) as:

$$y(n) = a_0x(n) + a_1x(n-1) + b_1y(n-1)$$

This shows, that at any given instant, the output of the IIR is a function of both the input signal, x(n), and the output signal, y(n). By applying the z-transform to this equation, it can be shown that the frequency response, H(z), may be expressed as:

$$H(z) = (a_0 + a_1 z^{-1}) / (1 - b_1 z^{-1})$$

Again, $z = e^{j\omega}$, $\omega = 2\pi f/F_s$ and F_s is the sample frequency. As an example, let F_s=10kHz, $a_0 = a_1 = 0.1$, and $b_1 = 0.85$. Computing H(z) for different frequencies (f) and plotting the magnitude of H(z) as a function of frequency yields:

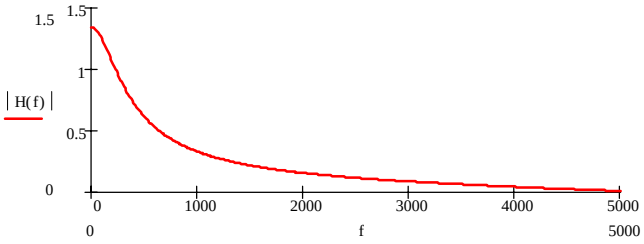


Figure 9.16. IIR Frequency Response for $a_0 = a_1 = 0.1$ and $b_1 = 0.85$

As with the FIR, it is a simple matter to expand the simple IIR to a multi-tap IIR. In this case, however, the left side of the IIR and the right side are not restricted to have the same number of delay taps. Thus, the number of "a" and "b" coefficients may not be the same. Also, as with the case for the FIR, certain filter responses may result in coefficients equal to 0, 1 or -1. Again, this leads to a circuit simplification when the IIR is implemented in hardware. Figure 9.17 shows how the simple IIR may be expanded to a multi-tap IIR. Note that the number of "a" taps (N) is not the same as the number of "b" taps (M). This accounts for the possibility a different number of coefficients in the feedforward ("a" coefficients) and feedback ("b" coefficients) sections.

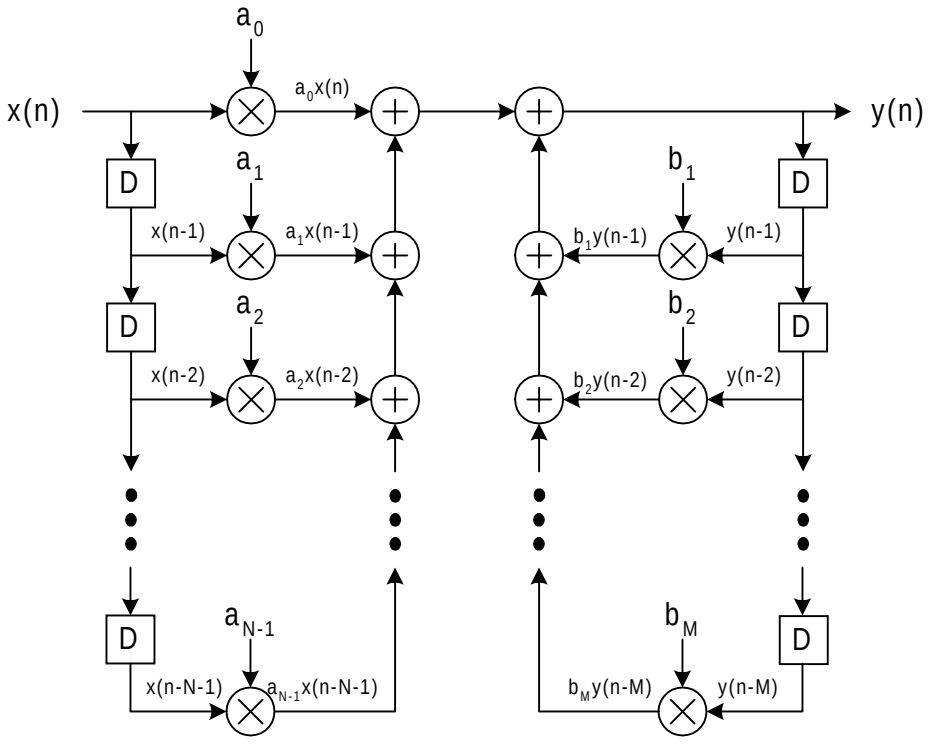


Figure 9.17. A Multi-tap IIR Filter

Examination of Figure 9.17 leads to an equation for the output, y(n), in terms of the input, x(n) for a multi-tap IIR. The result is:

$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2) + \ldots + a_{N-1}x(n-N-1) + b_1y(n-1) + b_2y(n-2) + \ldots + b_My(n-M)$$

Application of the z-transform leads to the general transfer function, H(z), of a multi-tap IIR filter with N feedforward coefficients and M feedback coefficients.

$$H(z) = (a_0 + a_1z^{-1} + a_2z^{-2} + \ldots + a_{N-1}z^{-(N-1)}) / (1 - b_1z^{-1} - b_2z^{-2} - \ldots - b_Mz^{-M})$$

Multirate DSP

Multirate DSP is the process of converting data sampled at one rate (F_{s1}) to data sampled at another rate (F_{s2}). If $F_{s1} > F_{s2}$, the process is called **decimation**. If $F_{s1} < F_{s2}$, the process is called **interpolation**.

The need for multirate DSP becomes apparent if one considers the following example. Suppose that there are 1000 samples of a 1kHz sinewave stored in memory, and that these samples were acquired using a 10kHz sample rate. This implies that the time duration of the entire sample set spans 100ms (1000 samples at 10000 samples/second). If this same sample set is now clocked out of memory at a 100kHz rate, it will require only 10ms to exhaust the data. Thus, the 1000 samples of data clocked at a 100kHz rate now look like a 10kHz sinewave instead of the original 1kHz sinewave. Obviously, if the sample rate is changed but the data left unchanged, the result is undesireable. Clearly, if the original 10kHz sampled data set represents a 1kHz signal, then it would be desireable to have the 100kHz sampled output represent a 1kHz signal, as well. In order for this to happen, the original data must somehow be modified. This is the task of multirate DSP.

Interpolation

The aforementioned problem is a case for interpolation. The function of an interpolator is to take data that was sampled at one rate and change it to new data sampled at a higher rate. The data must be modified in such a way that when it is sampled at a higher rate the original signal is preserved. A pictorial representation of the interpolation process is shown in Figure 9.18.

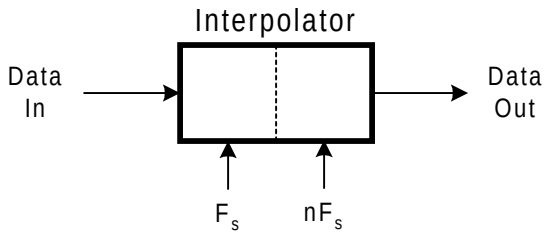


Figure 9.18. A Basic Interpolator

Notice that the interpolator has two parts. An input section that samples at a rate of F_s and an output section that samples at a rate of nF_s, where *n* is a positve integer greater than 1 (non-

integer multirate DSP is covered later). The structure of the basic interpolator indicates that for every input sample there will be *n* output samples. This begs the question: What must be done to the original data so that, when it is sampled at the higher rate, the original signal is preserved?

One might reason intuitively that, if *n*-1 zeroes are inserted between each of the input samples (*zero-stuffing*), then the output data would have the desired characteristics. After all, adding nothing (0) to something shouldn't change it. This turns out to be a pretty good start to the interpolation process, but it's not the complete picture.

The reason becomes clear if the process is examined in the frequency domain. Consider the case of interpolation by 3 ($n = 3$), as shown in Figure 9.19.

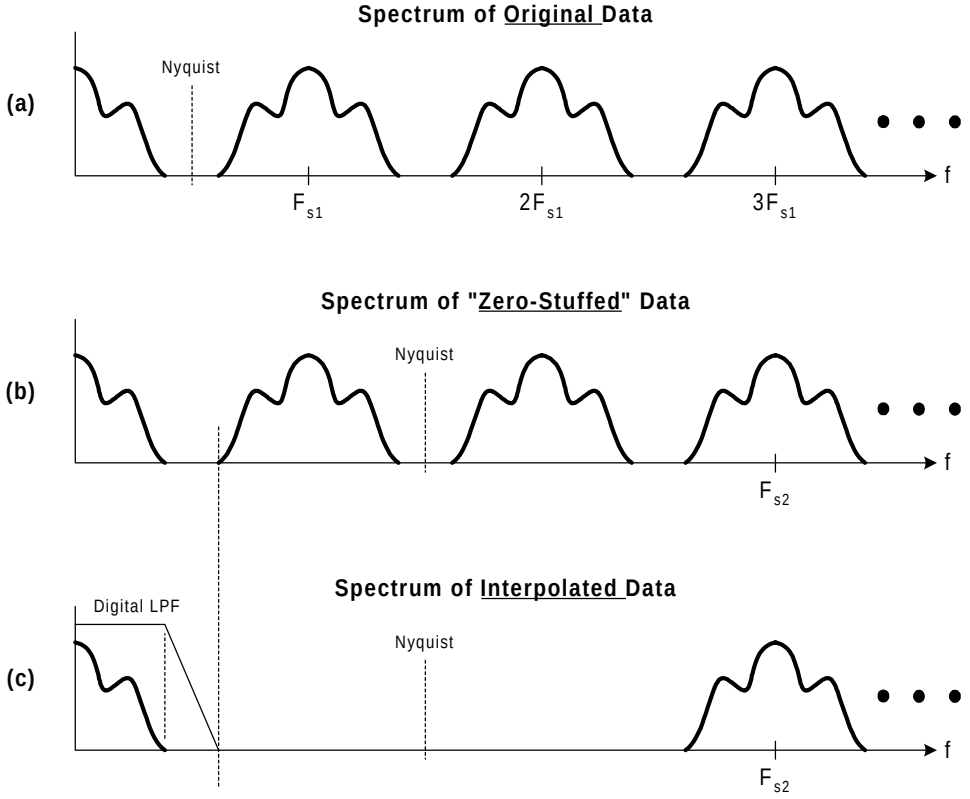


Figure 9.19. Frequency Domain View of Interpolation

The original data is sampled at a rate of F_{s1}. The spectrum of the original data is shown in Figure 9.19 (a). The Nyquist frequency for the original data is indicated at $\frac{1}{2}F_{s1}$.

With an interpolation rate of 3 the output sample rate, F_{s2}, is equal to $3F_{s1}$ (as is indicated by comparing Figure 9.19 (a) and (c)). The interpolation rate of 3 implies that 2 zeroes be stuffed between each input sample. The spectrum of the zero-stuffed data is shown in Figure 9.19 (b). Note that zero-stuffing retains the spectrum of the original data, but the sample rate has been increased a factor of 3 relative to the original sample rate. The new sample rate also results in a new Nyquist frequency at $\frac{1}{2}F_{s2}$, as shown.

It seems as though the interpolation process may be considered complete after zero-stuffing, since the two spectra match. But such is not the case. Here is why. The original data was sampled at a rate of F_{s1}. Keep in mind, however, that Nyquist requires the bandwidth of the

sampled signal to be less than $\frac{1}{2}F_{s1}$ (which it is). Furthermore, all of the information carried by the original data resides in the baseband spectrum at the far left side of Figure 9.19 (a). The spectral images centered at integer multiples of F_{s1} are a byproduct of the sampling process and carry no additional information.

Upon examination of Figure 9.19 (b) it is apparent that the Nyquist zone (the region from 0 to $\frac{1}{2}F_{s2}$) contains more than the single-sided spectrum at the far left of Figure 9.19 (a). In fact, it contains one of the images of the original sampled spectrum. Therein lies the problem: The Nyquist zone of the 3x sampled spectrum contains a different group of signals than the Nyquist zone of the original spectrum. So, something must be done to ensure that, after interpolating, the Nyquist zone of the interpolated spectrum contains exactly the same signals as the original spectrum.

Figure 9.19 (c) shows the solution. If the zero-stuffed spectrum is passed through a lowpass filter having the response shown in Figure 9.19 (c), then the result is exactly what is required to reproduce the baseband spectrum of the original data for a 3x sample rate. Furthermore, the filter can be employed as a lowpass FIR filter. Thus, the entire interpolation process can be done in the digital domain.

Decimation

The function of a decimator is to take data that was sampled at one rate and change it to new data sampled at a lower rate. The data must be modified in such a way that when it is sampled at the lower rate the original signal is preserved. A pictorial representation of the decimation process is shown in Figure 9.20 below.

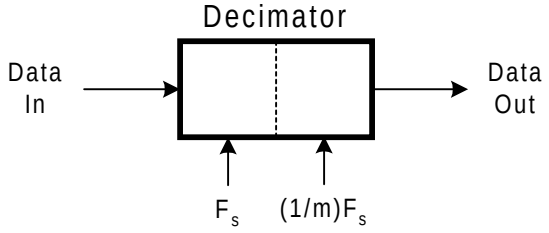


Figure 9.20. A Basic Decimator

Notice that the decimator has two parts. An input section that samples at a rate of F_s and an output section that samples at a rate of $(1/m)F_s$, where *m* is a positve integer greater than 1 (non-integer multirate DSP is covered later). The structure of the basic decimator indicates that for every *m* input samples there will be 1 output sample. This begs a similar question to that for interpolation: What must be done to the original data so that, when it is sampled at the lower rate, the original signal is preserved?

One might reason intuitively that, if every *m*-th sample of the input signal is picked off (ignoring the rest), then the output data would have the desired characteristics. After all, this constitutes a sparsely sampled version of the original data sequence. So, doesn't this reflect the same information as the original data? The short answer is, "No". The reason is imbedded in the ramifications of the Nyquist criteria in regard to the original data. The complete answer becomes apparent if the process is examined in the frequency domain, which is the topic of Figure 9.20.

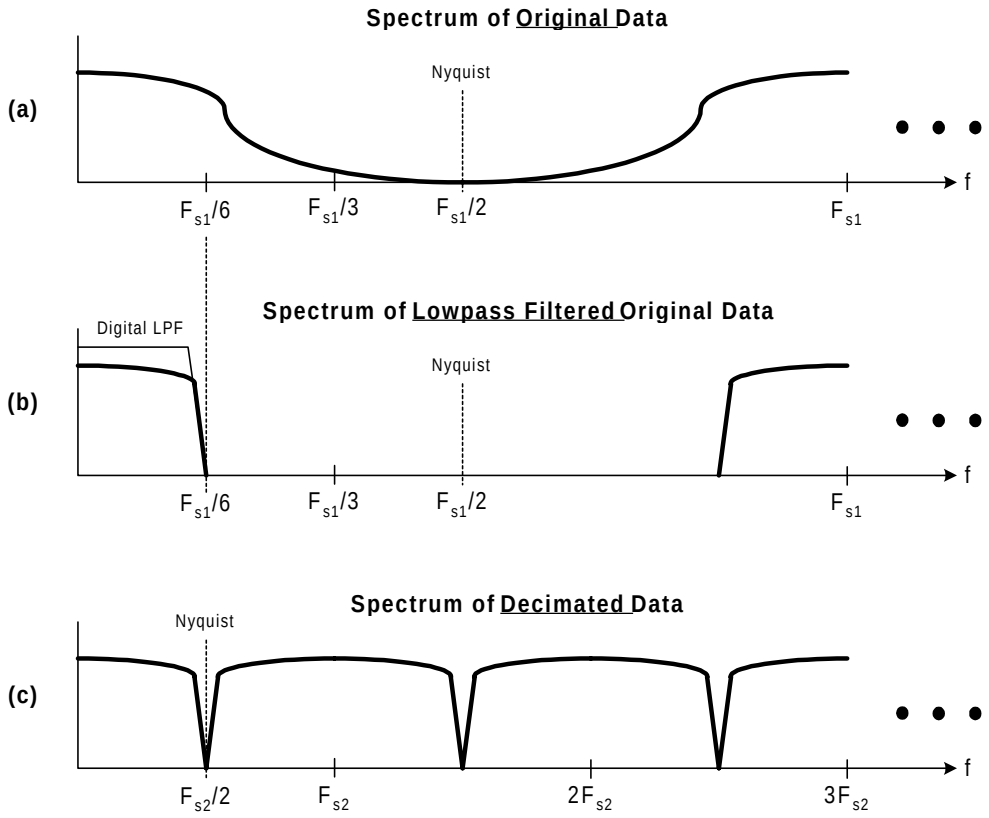


Figure 9.21. Frequency Domain View of Decimation

The matter at hand is that the decimation process is expected to translate the spectral information in Figure 9.21 (a) such that it can be properly contained in Figure 9.21 (c). The problem is that the Nyquist region of Figure 9.21 (a) is three times wider than the Nyquist region of Figure 9.21 (c). This is a direct result of the difference in sample rates. There is no way that the complete spectral content of the Nyquist region of Figure 9.21 (a) can be placed in the Nyquist region of Figure 9.21 (c). This brings up the cardinal rule of decimation: *The bandwidth of the data prior to decimation must be confined to the Nyquist bandwidth of the lower sample rate*. So, for decimation by a factor of *m*, the original data must reside in a bandwidth given by $F_s/(2m)$, where F_s is the rate at which the original data was sampled. Thus, if the original data contains valid information in the portion of the spectrum beyond $F_s/(2m)$, decimation is not possible. Such would be the case in this example if the portion of the orginal data spectrum beyond $F_{s1}/6$ in Figure 9.21 (a) was part of the actual data (as opposed to noise or other interfering signals).

Assuming that the spectrum of the original data meets the decimation bandwidth requirement, then the first step in the decimation process is to lowpass filter the original data. As with the interpolation process, this filter can be a digital FIR filter. The second step is to pick off every *m*-th sample using the lower output sample rate. The result is the spectrum of Figure 9.21 (c).

Rational n/m Rate Conversion

The interpolation and decimation processes described above allow only integer rate changes. It is often desirable to achieve a rational rate change. This is easily accomplished by cascading two (or more) interpolate/decimate stages. Using the same convention as in the previous sections, interpolation by *n* and decimation by *m*, rational *n/m* rate conversion can be accomplished as shown in Figure 9.22 below. It should be pointed out that **it is imperative that the interpolation process precede the decimation process** in a multirate converter. Otherwise, the bandwidth of the original data must be confined to $F_s/(2m)$, where F_s is the sample rate of the original data.

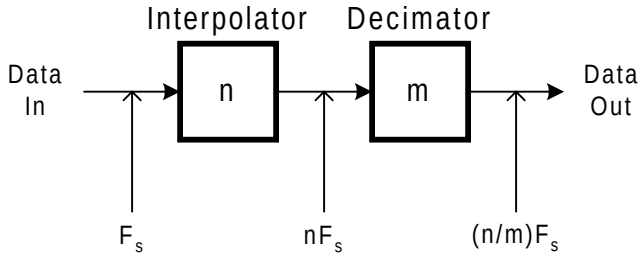


Figure 9.22. A Basic *n/m* Rate Converter

Multirate Digital Filters

It has been shown previously that interpolators and decimators both require the use of lowpass filters. These can be readily designed using FIR design techniques. However, there are two classes of digital filters worth mentioning that are particularly suited for multirate DSP: The *polyphase FIR* filter and the *Cascaded Integrator-Comb* (**CIC**) filter.

Polyphase FIR

The polyphase FIR is particularly suited to interpolation applications. Recall that the interpolation process begins with the insertion of *n*-1 zeroes between each input sample. The zero-stuffed data is then clocked out at the higher output rate. Next, the zero-stuffed data, which is now sampled at a higher rate, is passed through a lowpass FIR to complete the process of interpolation.

However, rather than performing this multi-step process, it is possible to do it all in one step by designing an FIR as follows. First, design the FIR with the appropriate frequency response, knowing that the FIR is to be clocked at the higher sample rate (nF_s). This will yield an FIR with a certain number of taps, T. Now add *n*-1 "delay" stages to each tap section of the FIR. A delay-only stage is the equivalent of a tap with a coefficient of 0. Thus, there are *n*-1 delay and multiply-by-0 operations interlaced between each of the original FIR taps. This arrangement maintains the desired frequency response characteristic while simultaneously making provision for zero-stuffing (a consequence of the additional *n*-1 multiply-by-0 stages). The result is an FIR with *n*T taps, but (*n*-1)T of the taps are multiply-by-0 stages.

It turns out that an FIR with this particular arrangement of coefficients can be implemented very efficiently in hardware by employing the approprate architecture. The special structure so created is a polyphase FIR. In operation, the polyphase FIR accepts each input sample *n* times because it is operating at the higher sample rate. However, *n*-1 of the samples are converted to zeros due to the additional delay-only stages. Thus, we get the zero-stuffing and filtering operation all in one package.

Cascaded Integrator-Comb (CIC)

The CIC filter is combination of a comb filter and an integrator. Before moving directly into the operation of the CIC filter, a presentation of the basic operation of a comb filter and an integrator ensues. The comb filter is a type of FIR consisting of a delay and add stages. The integrator can be thought of as a type of IIR, but without a feedforward section. A simple comb and integrator block diagram are shown in Figure 9.23.

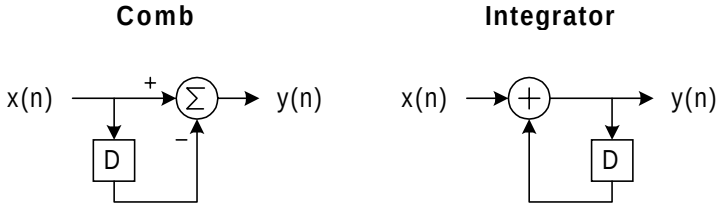


Figure 9.23. The Basic Comb and Integrator

Inspection of the comb and integrator architecture reveals an interesting feature: No multiply operations are required. In the integrator there is an implied multiply-by-one in the feedback path. In the comb, there is an implied multiply-by-negative-one in the feedforward path, but this can be implemented by simple negation, instead. The absence of multipliers offers a tremendous reduction in circuit complexity and, thereby, a significant savings in hardware compared to the standard FIR or IIR architectures. This feature is what makes a CIC filter so attractive.

In terms of frequency response, the comb filter acts as a notch filter. For the simple comb shown above, two notches appear; one at DC and the other at F_s (the sample rate). However, the comb can be slightly modified by cascading delay blocks together. This changes the number of notches in the comb response. In fact, K delay blocks produces K+1 equally spaced notches. Of the K+1 notches, one occurs at DC and another at F_s. The remainder (if any) are equally spaced between DC and F_s.

The integrator, on the other hand, acts as a lowpass filter. The response characteristics of each appears in Figure 9.24 below with frequency normalized to F_s. The comb response shown is that for K=1 (a single delay block).

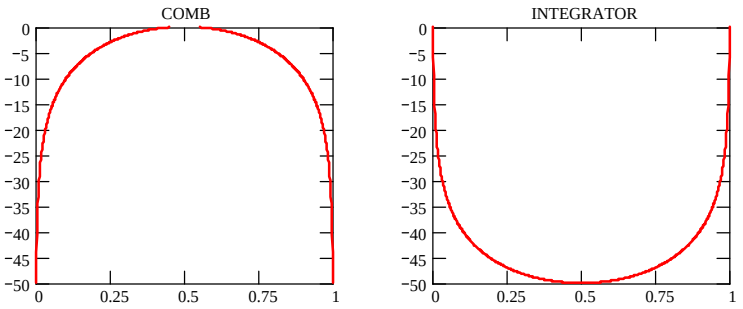


Figure 9.24. Frequency Response of the Comb and Integrator

The CIC filter is constructed by cascading the integrator and comb sections together. In order to perform integer sample rate conversion, the integrator is operated at one rate while the comb is operated at an integer multiple of the integrator rate. The beauty of the CIC filter is that it can function as either an interpolator or a decimator, depending on how the comb and integrator are connected. A simple CIC interpolator and decimator block diagram is shown in Figure 9.25. Note that the integrator and comb sections are operated at different sample rates for both the interpolator and decimator CIC filters.

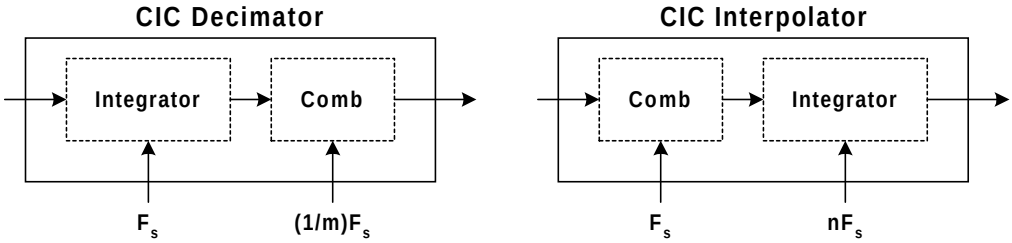


Figure 9.25. CIC Decimator and Interpolator

In the case of the interpolating version of the CIC, a slight modification to the basic architecture of its integrator stage is required which is not apparent in the simple block diagram above. Specifically, the integrator must have the ability to do zero-stuffing at its input in order to affect the increase in sample rate. For every sample provided by the comb, the integrator must insert n-1 zeroes.

The frequency response of the basic CIC filter is shown in Figure 9.26 below for an interpolation (or decimation) factor of 2; that is, n = 2 (or m = 2) as specified in the previous figure.

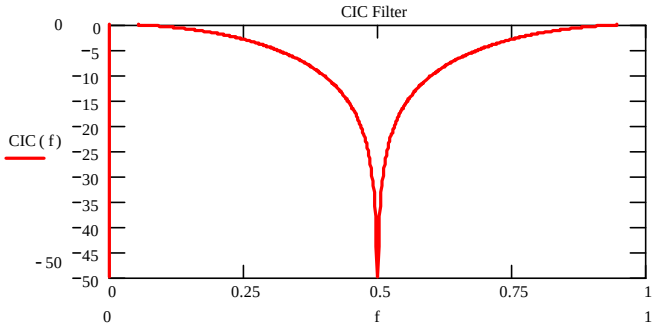


Figure 9.26. Frequency Response of the Basic CIC Filter

Inspection of the CIC frequency response bears out one of its potential problems: attenuation distortion. Recall that the plot above depicts a CIC with a data rate change factor of 2. The f=0.5 point on the horizontal access marks the frequency of the lower sample rate signal. The Nyquist frequency of the lower data rate signal is, therefore, at the f=0.25 point. Note that there is approximately 5dB of loss across what is considered the passband of the low sample rate signal. This lack of flatness in the passband can pose a serious problem in certain data communication applications and might actually preclude the use of the CIC filter for this very reason.

There are ways to overcome the attenuation problem, however. One method is to precede the CIC with an inverse filter that compensates for the attenuation of the CIC over the Nyquist bandwidth. A second method is to ensure further bandlimiting of the low sample rate signal so that its bandwidth is confined to the far lefthand portion of the CIC response where it is nearly flat.

It should be pointed out that the response curve of Figure 9.26 is for a *basic* CIC response. A CIC filter can be altered to change its frequency response characteristics. There are two methods by which this can be accomplished.

One method is to cascade multiple integrator stages together and multiple comb stages together. An example of a basic CIC interpolator modified to incorporate a triple cascade is shown in Figure 9.27.

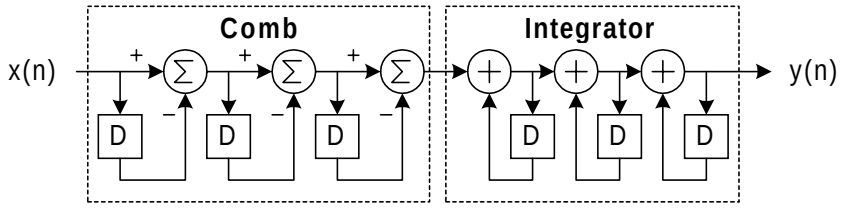


Figure 9.27. Triple Cascade CIC Interpolator

The second method is to add multiple delays in the comb section. An example of a basic CIC with one additional comb delay is shown in Figure 9.28.

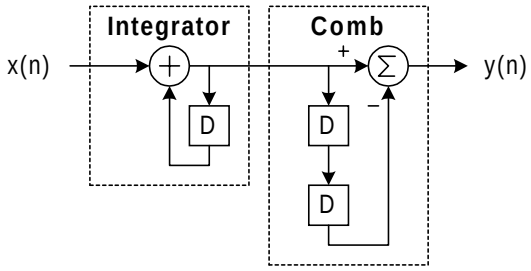


Figure 9.28. Double Delay CIC Decimator

The effect of each of these options is shown in Figure 9.29 along with the basic CIC response for comparison. In each case, a rate change factor of 2 is employed.

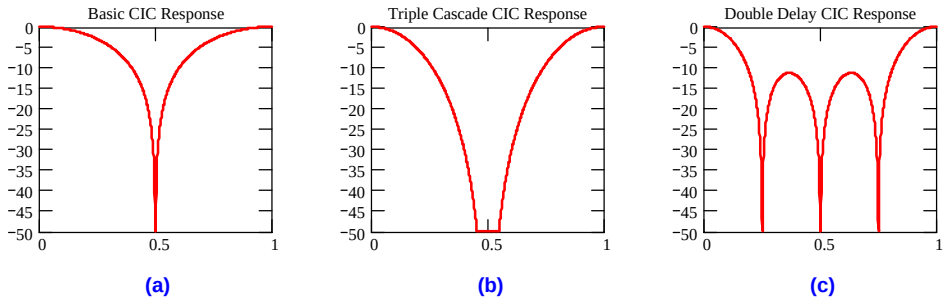


Figure 9.29. Comparison of Modified CIC Filter Responses

Note that cascading of sections (b) steepens the rate of attenuation. This results in more loss in the passband, which must be considered in applications where flatness in the passband is an issue. On the other hand, adding delays to the comb stage results in an increase in the number of zeroes in the transfer function. This is indicated in (c) by the additional null points in the frequency response, as well as increased passband attenuation. In the case of cascaded delays, consideration must be given to both the increased attenuation in the passband as well as reduced attenuation in the stopband. In addition, a combination of both methods can be employed allowing for a variety of possible responses.

Clock and Input Data Synchronization Considerations

In digital modulator applications it is important to maintain the proper timing relationship between the data source and the modulator. Figure 9.30 below shows a simple system block diagram of a digital modulator. The primary source of timing for the modulator is the clock which drives the DDS. This establishes the sample rate of the SIN and COS carrier signals of the modulator. Any samples propagating through the data pathway to the input of the modulator must occur at the same rate at which the carrier signal is sampled. It is important that samples arriving at the modulator input do so in a one-to-one correspondence with the samples of the carrier. Otherwise, the signal processing within the modulator is not carried out properly.

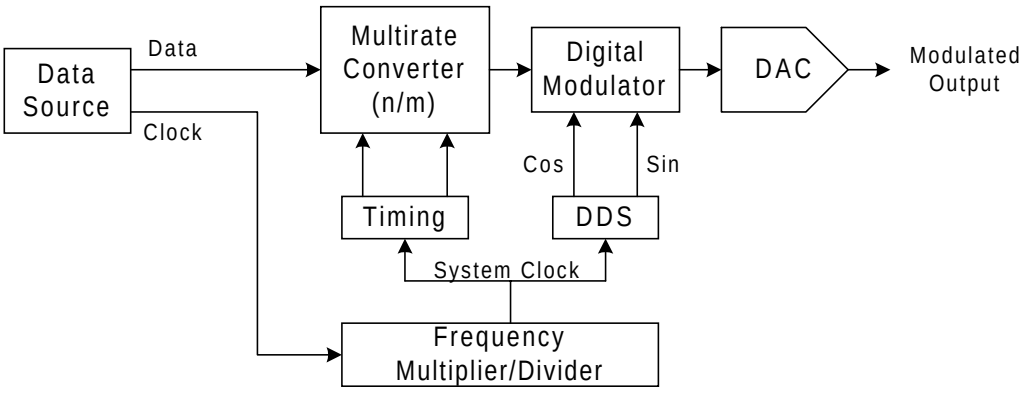


Figure 9.30. Generic Digital Modulator Block Diagram

Since the multirate converter must provide a rational rate conversion (i.e., the input and output rates of the converter must be expressible as a proper fraction), then the original data rate must

be rationally related to the system clock. That is, the system clock must operate at a factor of *n/m* times the data clock (or an integer multiple of *n/m*). In simpler modulator systems, the rate converter is a direct integer interpolator or decimator. In such instances, the clock multiplier is a simple integer multiplier or divider. Thus, the data source and system clock rates are related by an integer ratio instead of by a fractional ratio.

There are basically two categories of digital modulators when it comes to timing and synchronization requirements; **burst mode** modulators and **continuous mode** modulators. A burst mode modulator transmits data in packets; that is, groups of bits are transmitted as a block. During the time interval between bursts, the transmitter is idle. A continuous mode modulator, on the other hand, transmits a constant stream of data with no breaks in transmission.

It should quickly become apparent that the timing requirements of a burst mode modulator are much less stringent than those of a continuous mode modulator. The primary reason is that a burst mode modulator is only required to be synchronized with the data source over the duration of a data burst. This can be accomplished with a gating signal that synchronizes the modulator with the beginning of the burst. During the burst interval, the system clock can regulate the timing. As long as the system clock does not drift significantly relative to the data clock during the burst interal, the system operates properly. Naturally, as the data burst length increases the timing requirements on the system clock become more demanding.

In a continuous mode modulator the system clock must be synchronized with the data source at all times. Otherwise, the system clock will eventually drift enough to miss a bit at the input. This, of course, will cause an error in the data transmission process. Thus, an absolutely robust method must be implemented in order to ensure that the system clock and data clock remain continuously synchronized. Otherwise, transmission errors are all but guaranteed. Needless to say, great consideration must be given to the system timing requirements of any digital modulator application, burst or continuous, in order to make sure that the system meets its specified bit error rate (**BER**) requirements.

Data Encoding Methodologies and DDS Implementations

There is a wide array of methods by which data may be encoded before being modulated onto a carrier. In this section, a variety of data encoding schemes are addressed and their implementation in a DDS system. It is in this section that a sampling of Analog Devices' DDS products will be showcased to demonstrate the relative ease by which many applications may be realized. It should be understood that this section is by no means an exhaustive list of data encoding schemes. However, the concepts presented here may be extrapolated to cover a broad range of data encoding and modulation formats.

FSK Encoding

Frequency shift keying (**FSK**) is one of the simplest forms of data encoding. Binary 1's and 0's are represented as two different frequencies, f_0 and f_1, respectively. This encoding scheme is easily implemented in a DDS. All that is required is to arrange to change the DDS frequency tuning word so that f_0 and f_1 are generated in sympathy with the pattern of 1's and 0's to be transmitted. In the case of ADI's **AD9852**, **AD9853**, and **AD9856** the process is greatly simplified. In these devices the user programs the two required tuning words into the device

before transmission. Then a dedicated pin on the device is used to select the appropriate tuning word. In this way, when the dedicated pin is driven with a logic 0 level, f_0 is transmitted and when it is driven with a logic 1 level, f_1 is transmitted. A simple block diagram is shown in Figure 9.31 that demonstrates the implementation of FSK encoding. The concept is representative of the DDS implementation used in ADI's synthesizer product line.

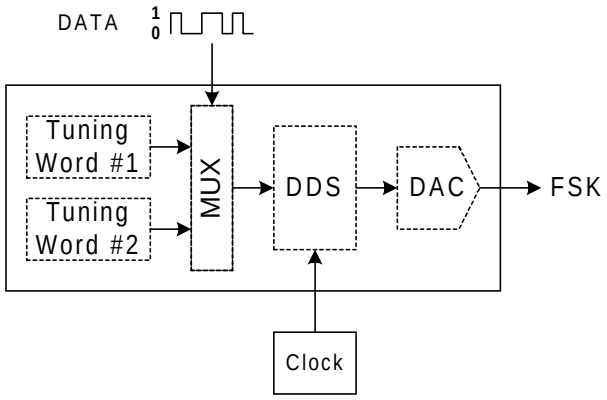


Figure 9.31. A DDS-based FSK Encoder

In some applications, the rapid transition between frequencies in FSK creates a problem. The reason is that the rapid transition from one frequency to another creates spurious components that can interfere with adjacent channels in a multi-channel environment. To alleviate the problem a method known as **ramped FSK** is employed. Rather than switching immediately between frequencies, ramped FSK offers a gradual transition from one frequency to the other. This significantly reduces the spurious signals associated with standard FSK. Ramped FSK can also be implemented using DDS techniques. In fact, the **AD9852** has a built in Ramped FSK feature which offers the user the ability to program the ramp rate. A model of a DDS Ramped FSK encoder is shown in Figure 9.32.

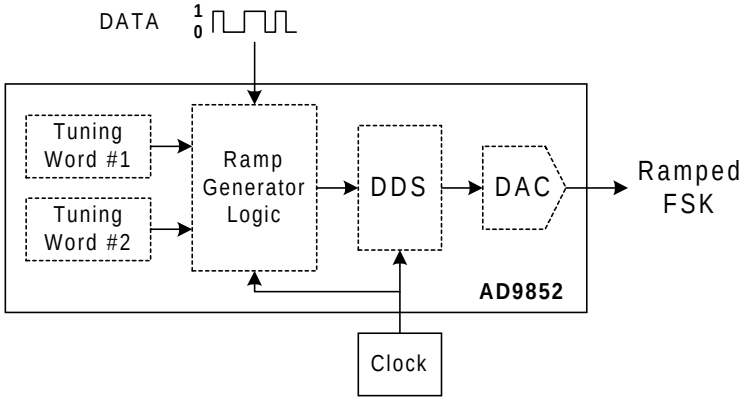


Figure 9.32. A DDS-based Ramped FSK Encoder

A variant of FSK is **MFSK** (multi-frequency FSK). In MFSK, 2^B frequencies are used (B>1). Furthermore, the data stream is grouped into packets of B bits. The binary number represented by any one B-bit word is mapped to one of the 2^B possible frequencies. For example, if B=3, then there would be 2^3, or 8, possible combinations of values when grouping 3 bits at a time. Each combination would correspond to 1 of 8 possible output frequencies, f_0 through f_7.

PSK Encoding

Phase shift keying (**PSK**) is another simple form of data encoding. In PSK the frequency of the carrier remains constant. However, binary 1's and 0's are used to phase shift the carrier by a certain angle. A common method for modulating phase on a carrier is to employ a quadrature modulator. When PSK is implemented with a quadrature modulator it is referred to as **QPSK** (quadrature PSK).

The most common form of PSK is **BPSK** (binary PSK). BPSK encodes 0° phase shift for a logic 1 input and a 180° phase shift for a logic 0 input. Of course, this method may be extended to encoding groups of B-bits and mapping them to 2^B possible angles in the range of 0° to 360° (B>1). This is similar to MFSK, but with phase as the variable instead of frequency. To distinguish between variants of PSK, the binary range is used as a prefix to the PSK label. For example, B=3 is referred to as 8PSK, while B=4 is referred to as 16PSK. These may also be referred to as 8QPSK and 16QPSK depending on the implementation.

Since PSK is encoded as a certain phase shift of the carrier, then decoding a PSK transmission requires knowledge of the absolute phase of the carrier. This is called **coherent** detection. The receiver must have access to the transmitter's carrier in order to decode the transmission. A workaround for this problem is differential PSK, or **DPSK** and can be extented to **DQPSK**. In this scheme, the change in phase of the carrier is dependent on the value of the previously sent bit (or symbol). Thus, the receiver need only identify the relative phase of the first symbol. All subsequent symbols can then be decoded based on phase changes relative to the first. This method of detection is referred to as **noncoherent** detection.

PSK encoding is easily implemented with ADI's synthesizer products. Most of the devices have a separate input register (a *phase register*) that can be loaded with a phase value. This value is directly added to the phase of the carrier without changing its frequency. Modulating the contents of this register modulates the phase of the carrier, thus generating a PSK output signal. This method is somewhat limited in terms of data rate, because of the time required to program the phase register. However, other devices in the synthesizer family, like the **AD9853**, eliminate the need to program a phase register. Instead, the user feeds a serial data stream to a dedicated input pin on the device. The device automatically parses the data into 2-bit symbols and then modulates the carrier as required to generate QPSK or DQPSK.

QAM Encoding

Quadrature amplitude modulation (**QAM**) is an encoding scheme in which both the amplitude and phase of the carrier are used to convey information. By its name, QAM implies the use of a quadrature modulator. In QAM the input data is parsed into groups of B-bits called symbols. Each symbol has 2^B possible states. Each state can be represented as a combination of a specific phase value and amplitude value. The number of states is usually used as a prefix to the QAM label to identify the number of bits encoded per symbol. For example, B=4 is referred to as 16QAM. Systems are presently in operation that use B=8 for 256QAM.

The assignment of the possible amplitude and phase values in a QAM system is generally optimized in such a way to maximize the probability of accurate detection at the receiver. The

tool most often used to display the amplitude and phase relationship in a QAM system is the **constellation diagram** or **I-Q diagram**. A typical 16QAM constellation is shown in Figure 9.33.

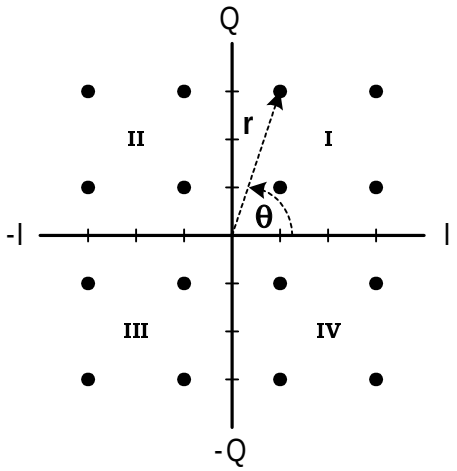


Figure 9.33. 16QAM Constellation

Each dot in Figure 9.33 represents a particular symbol (4-bits). The constellation of Figure 9.33 uses I and Q values of ±1 and ±3 to locate a dot. For example, in Quadrant I, the dots are represented by the (I,Q) pairs: (1,1), (1,3), (3,1) and (3,3). In Quadrant II, the dots are mapped as (-1,1), (-1,3), (-3,1) and (-3,3). Quadrants III and IV are mapped in similar fashion.

Note that each dot can also be represented by a vector starting at the origin and extending to a specific dot. Thus, an alternative means of interpreting a dot is by amplitude, r, and phase, θ. It is the r and θ values that define the way in which the carrier signal is modulated in a QAM signal. In this instance, the carrier can assume 1 of 3 amplitude values combined with 1 of 12 phase values.

To reiterate, each amplitude and phase combination represents one of 16 possible 4-bit symbols. Furthermore, the combination of a particular phase and amplitude value defines exactly one of the 16 dots. Additionally, each dot is associated with a specific 4-bit pattern, or symbol.

It is a simple matter to extend this concept to symbols with more bits. For example, 256QAM encodes 8-bit symbols. This yields a rather dense constellation of 256 dots. Obviously, the denser the constellation, the more resolution that is required in both phase and amplitude. This implies a greater chance that the receiver produces a decoding error. Thus, a higher QAM number means a less tolerant system to noise.

The available signal to noise ratio (SNR) of a data transmission link has a direct impact on the BER of the system. The available SNR and required BER place restrictions on the type of encoding scheme that can be used. This becomes apparent as the density of the QAM encoding scheme increases. For example, some data transmission systems have a restriction on the amount transmit power that can be employed over the link. This sets an upper bound on the SNR of the system. It can be shown that SNR and BER follow an inverse exponential

relationship. That is, BER increases exponentially as SNR decreases. The density of the QAM encoding scheme amplifies this exponential relationship. Thus, in a power limited data link, there comes a point at which increasing QAM density yields a BER that is not acceptable.

As with the other encoding schemes, a differential variant is possible. This leads to differential QAM (**DQAM**). Standard QAM requires coherent detection at the receiver, which is not always pssible. DQAM solves this problem by encoding symbols in a manner that is dependent on the preceding symbol. This frees the detector from requiring a reference signal that is in absolute phase with the transmitter's carrier.

The **AD9853** is capable of direct implementation of the 16QAM and D16QAM encoding. The **AD9856**, on the other hand, can be operated in any QAM mode. However, the **AD9856** is a general purpose quadrature modulator that accepts 12-bit 2's complement numbers as I and Q input values. Thus, the burden is on the user to parse the input data stream and convert it to bandlimited I and Q data before passing it to the **AD9856**.

FM Modulation

Frequency modulation (**FM**) is accomplished by varying the frequency of a carrier in sympathy with another signal, the message. In a DDS system, FM is implemented by rapidly updating the frequency tuning word in a manner prescribed by the amplitude of the message signal. Thus, performing FM modulation with a DDS requires extra hardware to sample the message signal, compute the appropriate frequency tuning word from the sample, and update the DDS.

A broadcast quality FM transmitter using the **AD9850** DDS is described in an application note (AN-543) from Analog Devices. The application note is included in the Appendix.

Quadrature Up-Conversion

Figure 9.30 shows an example of a generic modulator. A quadrature up-converter is a specific subset of the generic modulator and is shown in Figure 9.34.

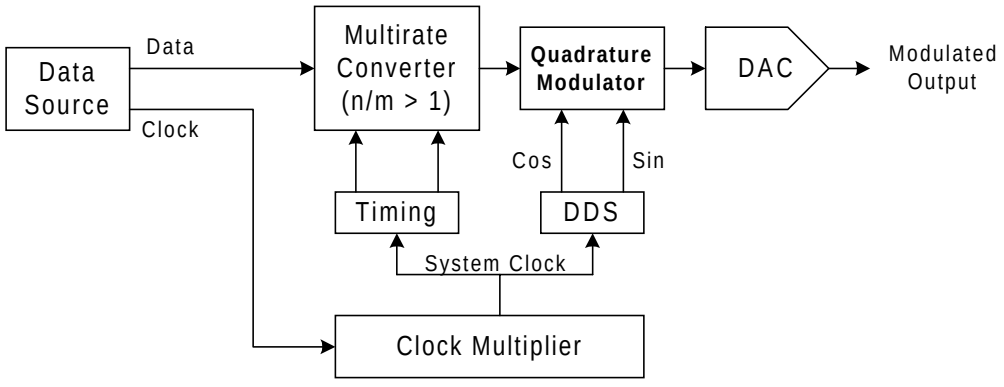


Figure 9.34. Quadrature Up-Converter

In an up-converter, the incoming data is intended to be broadcast on a carrier which is several times higher in frequency than the data rate. This automatically implies that the multirate converter ratio, n/m, is always greater than 1. Similarly, the system clock is generated by a

frequency multiplier since the data rate is less than the carrier frequency, which must be less than the DDS sample rate. Also, as its name implies, the modulator portion of the up-converter is of the quadrature variety.

ADI offers an integrated quadrature up-converter, the **AD9856**. The **AD9856** can be operated with a system clock rate up to 200MHz. The clock multiplier, multirate converter, DDS, digital quadrature modulator, and DAC are all integrated on a single chip. A simplified block diagram of the **AD9856** appears in Figure 9.35.

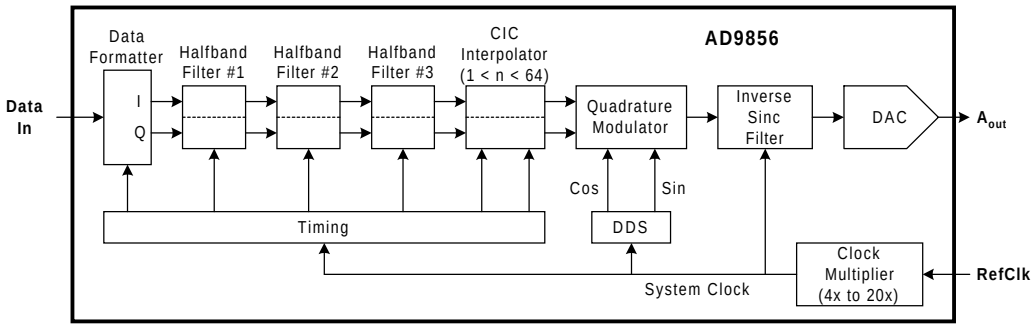


Figure 9.35. **AD9856** 200MHz Quadrature Digital Up-Converter

The halfband filters are polyphase FIR filters with an interpolation rate of 2 (n=2). Halfband Filter #3 can be bypassed, which yields a cumulative interpolation rate of either 4 or 8 via the halfband filters. The CIC interpolator offers a programmable range of 2 to 63. Thus, the overall interpolation rate can be as low as 8 or as high as 504. The input data consists of 12-bit 2's complement words that are delivered to the device as alternating I and Q values. The device also sports an Inverse Sinc Filter which can be bypassed, if so desired. The Inverse Sinc Filter is an FIR filter that provides pre-emphasis to compensate for the sinc rolloff characteristic associated with the DAC.

References:

[1] Gibson, J. D., 1993, Principles of Digital and Analog Communications, Prentice-Hall, Inc.

[2] Ifeachor, E. C. and Jervis, B. W., 1996, Digital Signal Processing: A Practical Approach, Addison-Wesley Publishing Co.

[3] Lathi, B. P., 1989, Modern Digital and Analog Communication Systems, Oxford University Press

[4] Oppenheim, A. V. and Willsky, A. S., 1983, Signals and Systems, Prentice-Hall, Inc.

[5] Proakis, J. G. and Manolakis, D. G., 1996, Digital Signal Processing: Principles, Algorithms, and Applications, Prentice-Hall, Inc.

Section 10. Using DDS Images as Primary Output Signals in VHF/UHF Applications

By Rick Cushing, Applications Engineer, Analog Devices, Inc.

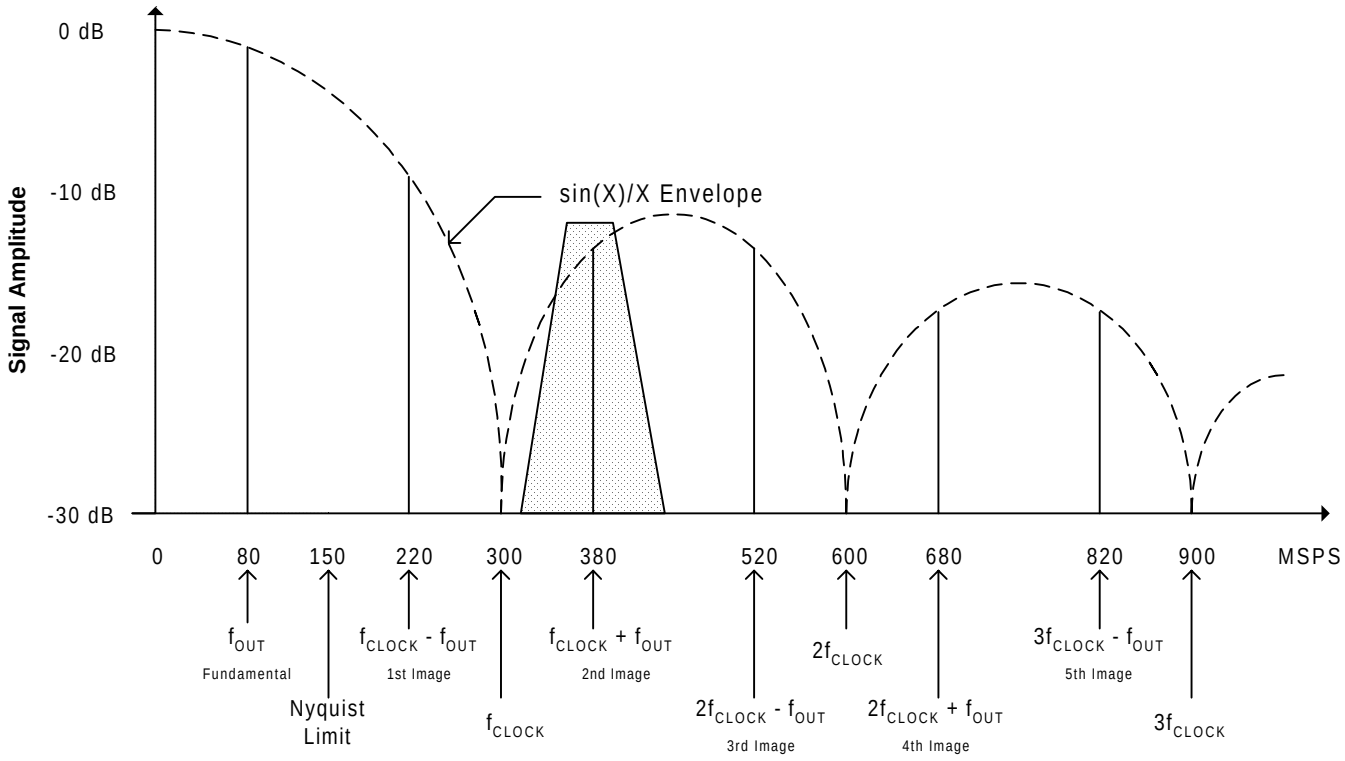


Figure 10-1. Frequency Representation of the DDS/DAC Output Spectrum

Some Properties of the DDS/DAC Output Signal

The DDS/DAC sampled output has many important properties that can be exploited. First, it is a "real" signal, that is, it is composed of both positive and negative frequencies. These signals are sym-metrical or "mirrored" about 0 Hz. Secondly, the sampled output implies that the positive and negative frequencies are replicated at locations separated by the sample frequency, both above and below 0 Hz to infinity (theoretically). These replications of the fundamental signal are called images

Using the example in Figure 10-1, one can conclude that if Fout = 80 MHz then there must also be a negative frequency component at –80 MHz. Furthermore, since the sample frequency is 300 MHz, these *two* signals should be replicated at integer multiples of the sample frequency, both above and below 0 Hz. Figure 10-1 shows only the positive frequencies, but one can see that if a –80 MHz component existed, it should be replicated at 220 MHz (-80 + 300 = 220), 520 MHz (220 +300 = 520), 820 MHz (520 +300 = 820), etc. The same is true for the fundamental signal located at +80 MHz which will have replica's of itself located at 380, 680, 980 MHz, etc. The same thing is going on below 0Hz as well. Theoretically, the amplitude of the images of a

sampled signal are the same as the fundamental and that the images continue to infinity. In reality, the DAC output amplitude follows the relationship established by sin(x)/x seen in Figure 10-1 above.

So what's different about the *negative frequency image* at 220 MHz and the *positive frequency image* at 380 MHz in Figure 10-1? A change in frequency of the fundamental output will cause the positive image to track the change whereas the negative image will change frequency in the opposite direction! A positive phase shift of the fundamental will result in a corresponding phase shift of the positive image and a negative phase shift for the negative image. These attributes have been termed "*spectral inversion*" and are traced back to the mirror image property of a "real" signal. If images are to be used as primary output signals, then the effects of spectral inversion should be accounted for.

Phase Noise

Phase noise of images will degrade since phase noise is referenced to the "carrier" power (dBc). It stands to reason that a reduction in carrier level (as experienced in utilizing images) while maintaining the same noise power in the signal "skirt" will result in a degraded phase noise measurement.

Image SFDR & SNR

The disadvantage of using images as primary output signals is basically the decrease in signal to noise ratio and SFDR (spurious-free dynamic range). The image amplitude as well as the fundamental amplitude are all subject to sin(x)/x amplitude variations with frequency. The value of x is the ratio of the frequency of interest to the sample frequency multiplied by π. When calculating sine, use radians instead of degrees. Unfortunately, spurious signals in the DDS/DAC output spectrum seem to get more numerous and larger the further one goes from the Nyquist limit! (see Figure 10-2A). Even given this constraint, it is possible to locate "sweet" spots in the sea-of-noise where the images reside, Figure 10-2B.

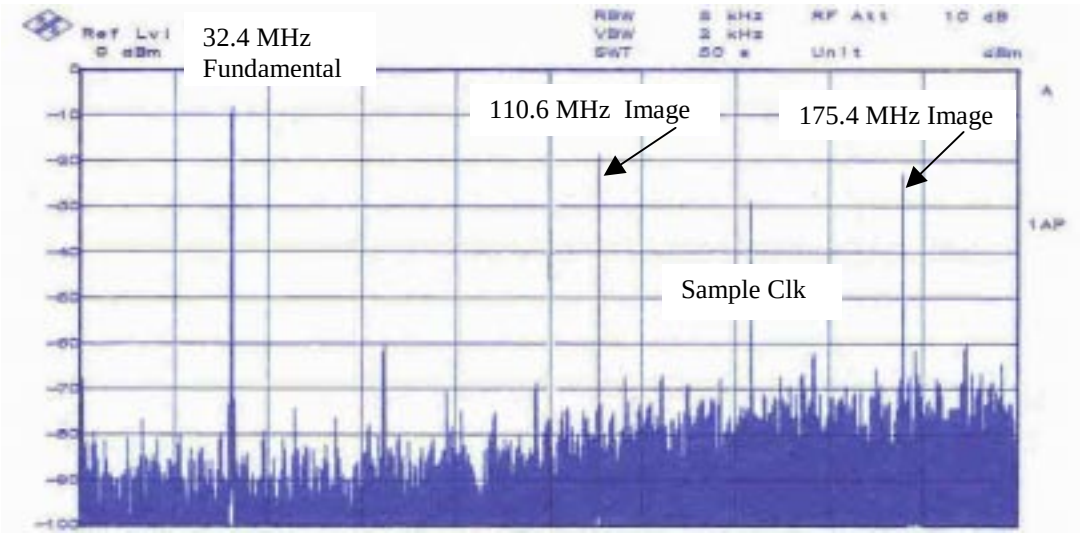
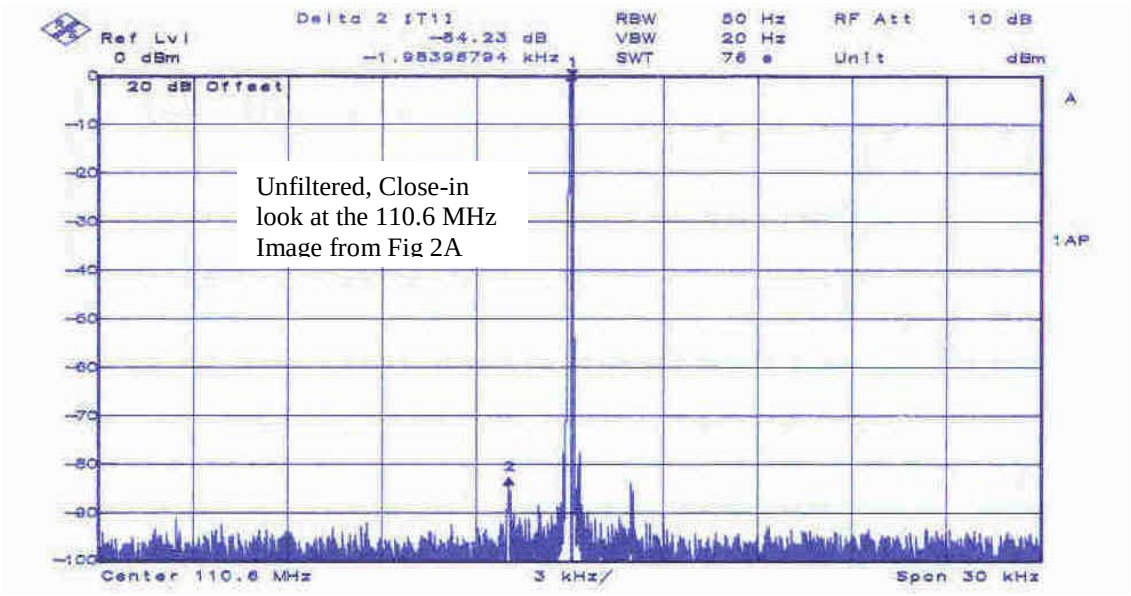


Figure 10-2A & B: A (above) Images in a dc to 200 MHz View and B (below), -84 dBc SFDR at 110.6 MHz



If the spectrum of the image in Figure 10-2B is intriguing then one can certainly see why DDS/DAC images should be considered as primary output signals. Perhaps the most challenging aspect of this technique is the need for adequate bandpass filtering of the image to separate the image from the surrounding spurs. SAW filters have been successfully implemented with DDS images at VHF and UHF frequencies; however, their inherent loss may require the use of an RF amplifier, such as inexpensive MMIC's. Figure 10-1 graphically shows the intent of bandpass filtering as well as identifying the power levels of various images calculated using sin(x)/x.

Figure 10-2A shows that the frequency region above Nyquist is heavily populated with spurious signals in addition to the images of the fundamental signal. This suggests that if SFDR is

important to a particular application, then the movement of the image frequency will be restricted due to the likely presence of nearby spurs. Applications such as frequency hopping will require special attention to avoid transmitting spurious signals along with the desired signal when operating in super-Nyquist areas.

Summary

Use of super-Nyquist DDS/DAC output frequencies can save several costly RF stages (oscillator, mixer, filter) by directly producing the desired output frequency through use of naturally occurring images. Images are not harmonics of the fundamental, they are linearly transposed replicas of the fundamental signal. The remarkable advantage of image utilization is that a low fundamental and sample clock frequency produce useable signals well above the accepted range that Nyquist's theory predicts. This technique is not suitable for some applications; however, many VHF applications, including clock-generators, can be well served using this simple technique. Use of images is restricted to the first three or four images since amplitude loss of succeeding images reduces SNR to unusable levels

Section 11. Ancillary DDS Techniques, Features, and Functions

By Ken Gentile, Systems Engineer, Analog Devices, Inc.

Improving FSDR with Phase Dithering

In Section 4 the effects of phase truncation in the architecture of a DDS were covered. The net result is that phase truncation can result in spurs in the DDS output spectrum depending on the choice of the tuning word. In some applications, it is desirable to reduce the spur energy at all costs. Phase dithering offers a means of reducing spur energy. However, this is at the expense of an elevated noise floor and increased phase noise.

Figure 11.1 is a simple functional block diagram of a DDS. In Figure 11.1(a) the T-bit tuning word feeds the input to the accumulator. The most significant A-bits of the accumulator output are fed to the angle-to-amplitude conversion (**AAC**) block, which then drives a D-bit DAC. The assumption is made that T > A > D.

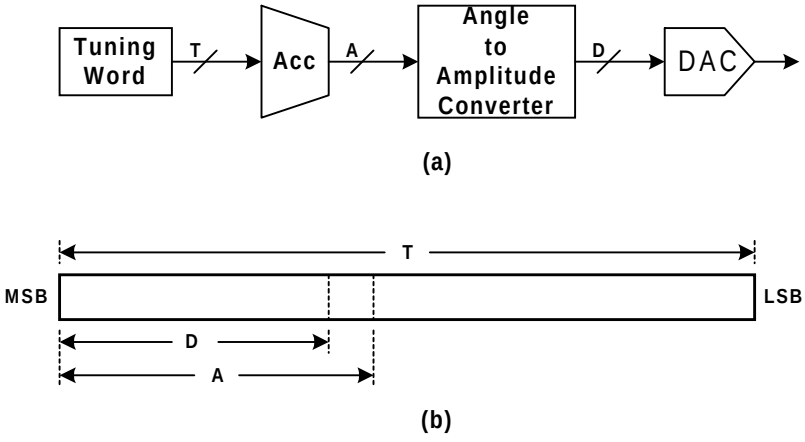


Figure 11.1. DDS Block Diagram

Figure 11.1(b) shows the relationship of the various word widths in the DDS. Note that the bits at the output of the accumulator constitute a word size (A) that is a subset of the tuning word size (T). Similarly, the output of the AAC constitutes a word size (D) that is a subset of the accumulator output word size (A). The relationship between the various word sizes in Figure 11.1(b) is typical of DDS architectures.

Phase dithering requires that the phase values generated by the accumulator contain a certain amount of noise. This can be accomplished by adding a small random number to each phase value generated at the output of the accumulator. This method is shown in the Figure 11.2.

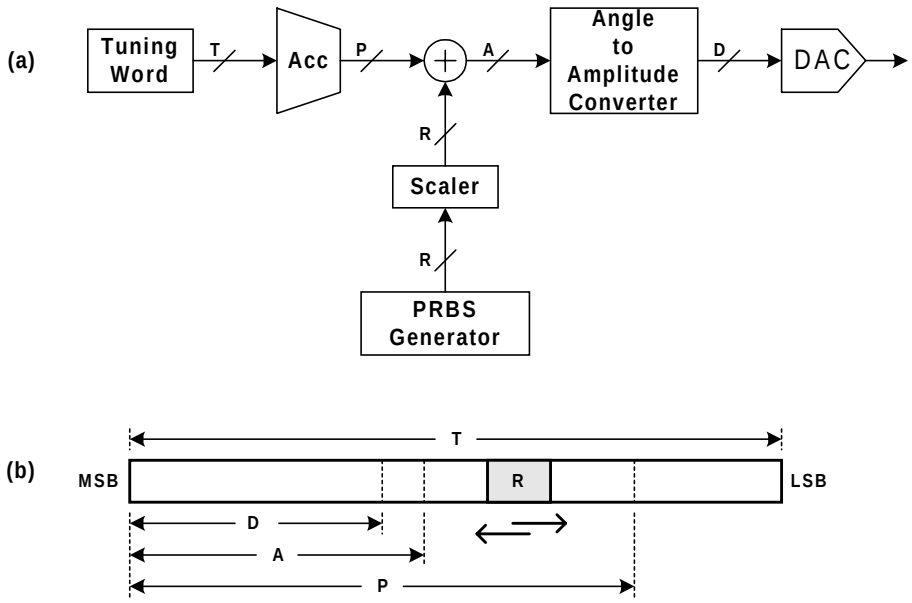


Figure 11.2. DDS Block Diagram

In the phase dithering model, a pseudorandom binary sequence (PRBS) generator is used to produce a new R-bit random number with each update of the accumulator (PRBS generators are covered in the Appendix). The PRBS numbers are scaled by powers of 2 (left- or right-shifted) to fit into the desired range of the P-bit word at the output of the accumulator. The random number is positioned so that its MSB is less than the LSB of the A-bit word at the input to the AAC. It should be noted that shifting the random number so that it overlaps the A-bit word is not recommended. Doing so defeats the purpose of having A-bits of phase resolution in the first place, since it adds noise that is greater than the quantization noise associated with the A-bit word.

The position of the R-bit word has significant impact on the magnitude of the phase dithering. This should be obvious since left-shifting the random number increases its impact when summed with the P-bit word taken from the output of the accumulator. Typically, the MSB of the random number is positioned one bit less than the LSB of the A-bit word that is fed to the AAC.

The width of the random number has an impact on the way in which the random phase is spread across the output spectrum of the DDS. Typically, a 3 or 4 bit random number is sufficient.

Understanding DDS Frequency "Chirp" Functionality

Frequency chirp is a method of transitioning between two different output frequencies, f_1 and f_2, over a specified time interval. The simplest chirp is a linear sweep from f_1 to f_2. However, in more advanced chirp systems the frequency transition from f_1 to f_2 can be a nonlinear function of time. A DDS-based chirp system is shown in Figure 11.3.

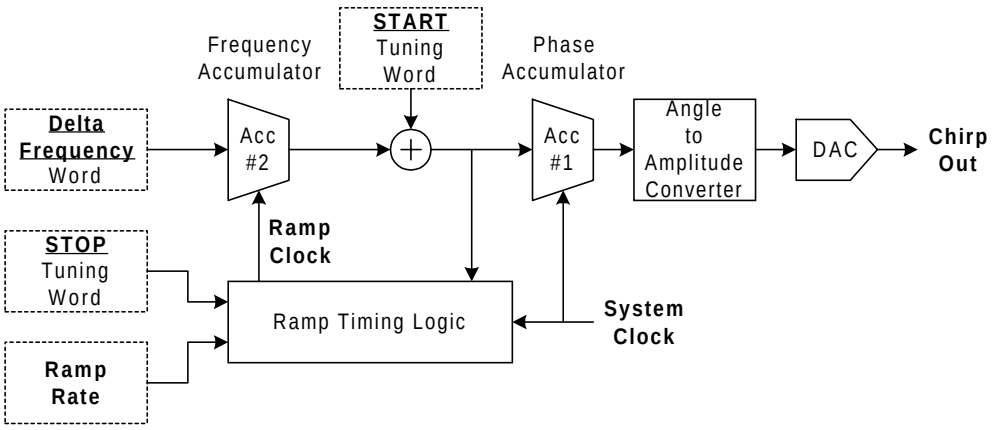


Figure 11.3. DDS-Based Chirp System

Beginning with the Phase Accumulator and working to the right, the system is a duplicate of the basic DDS. However, note that the Phase Accumulator is not driven by a static tuning word value, as it is the basic DDS. Instead, the input to the Phase Accumulator is the sum of the START Tuning Word and the output of the Frequency Accumulator. The START Tuning Word is the frequency tuning word that marks f_1 (the beginning frequency of the chirp).

The Frequency Accumulator recursively sums the Delta Frequency Word at a rate prescribed by the Ramp Clock. Thus, each occurrence of the Ramp Clock increments the value of the input to the Phase Accumulator by the Delta Frequency Word. Since the input to the Phase Accumulator determines the output frequency of the DDS and this value is changing with time, then the output frequency of the DDS is changing with time, also. This is the basic mechanism for generating a chirp waveform.

Assuming that the Ramp Clock generates an clock pulse at regular intervals and the Delta Frequency Word is constant, then the Frequency Accumulator output grows at linear rate. That is, the slope of the frequency vs. time function is constant. This constitutes a linear chirp.

It is also possible to generate nonlinear chirp frequency profiles. Notice that if the Delta Frequency Word is modified the rate at which the DDS output frequency changes is also modified. That is, the slope of the frequency vs. time function is modified. So, changing the Delta Frequency Word during the chirp interval changes the slope of the frequency vs. time function in mid-chirp. This provides a means to generate almost any chirp function using piecewise linear approximation.

Implementing chirp functionality in a DDS requires additional timing and control. This is the function of the Ramp Timing Logic. It actually serves two purposes. First, it must divide down the System Clock to produce the appropriate Ramp Clock frequency as defined by the Ramp Rate input. Second, it must terminate the Ramp Clock when the output frequency reaches f_2 (the STOP frequency of the chirp). Note that the STOP Tuning Word defines frequency, f_2. Terminating the Ramp Clock is accomplished by monitoring the output of the Frequency Accumulator and the STOP Tuning Word value. When the output of the Frequency Accumulator is greater than or equal to the STOP Tuning Word, the Ramp Clock is disabled. It should be noted that the Ramp Timing Logic should also ensure that Frequency Accumulator

does not exceed the STOP Tuning Word value. That is, it should clamp its value to the value of the STOP Tuning Word.

This basic architecture is implemented in ADI's **AD9852** DDS. The device offers a high degree of flexibility in implementing a variety of chirp applications.

Achieving Output Amplitude Control/Modulation Within a DDS Device

In some applications it is desirable to control the amplitude of the DDS output signal. An obvious application would be a DDS-based AM transmitter. Another use would be for those applications in which a carrier is only present during data transmission and is absent, otherwise. Burst applications such as this can cause a problem when multiple devices share the transmission medium. This is because when the carrier transitions from one power level to another, the abrupt change creates a burst of broadband noise. This may cause transmission errors for the other devices sharing the medium. Amplitude control allows the user to gradually change carrier power rather than have it abruptly switch from one state to the other. This significantly reduces the noise generated during the switching transient.

The basic DDS architecture does not provide for amplitude control. However, amplitude control is made possible by a simple modification; insertion of a multiplier preceding the DAC. A DDS with amplitude control is available in ADI's **AD9852**. A block diagram of a DDS with amplitude control is shown in Figure 11.4.

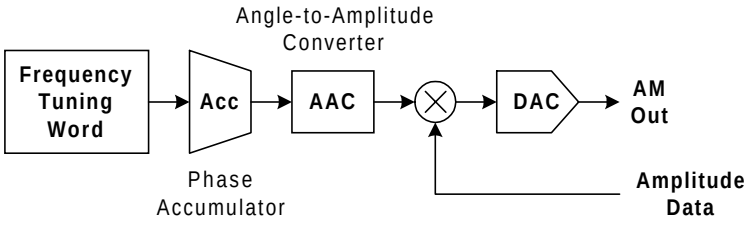


Figure 11.4. DDS With Amplitude Control

In Figure 11.4, assume that the value of the Amplitude Control word can take on values between 0 and 1, inclusive. When it is multiplied with the output of the Angle to Amplitude Converter (AAC), it effectively scales the AAC results. Thus, the data arriving at the input of the DAC is a scaled version of the AAC output. For example, if the Amplitude Control word is 0, then the input to the DAC is 0 and there is no AM signal present at the output of the DAC. As the value of the Amplitude Control word increases, the data arriving at the input to the DAC increases. For a Control Word of 1, the DAC data is the same as the data generated by the AAC. Thus, variations of the Amplitude Control word as a function of time effectively modulates the output of the DAC. This provides for a means to generate AM signals, as well as a means to gradually control carrier ON/OFF transients.

Synchronization of Multiple DDS Devices

There are applications in which it is desirable to synchronize two or more DDS's. A particular case is when it is necessary to generate time-synchronous analog I and Q channels. One DDS

can be used to generate the analog I channel and a second DDS the analog Q channel. However, it is imperative that the two devices be syncronized to minimize I-Q imbalance. A method for synchronizing two AD9850's or AD9851's is contained in the Appendix.

Alternatively, the AD9854 offers an integrated solution. It provides two output DACs; one for the I channel and one for the Q channel. Synchronization is handled on-chip, thus virtually eliminating phase imbalance between its quadrature outputs.

Section 12. DDS Evaluation Techniques

By Rick Cushing, Applications Engineer, Analog Devices, Inc

PC-based Evaluation Boards for DDS Devices

Given the proliferation of Personal Computers throughout most cultures and economies of the world, it makes good sense to choose the PC as a programming vehicle to aid in the designer's evaluation of DDS products. Software written in Visual Basic© or C++ programming languages allow designers to fully operate the DDS via a menu-driven "control panel", in a Windows© environment, as displayed on their PC monitor.

Mathematical computations of the frequency tuning words and control words necessary to achieve fixed or dynamic operation of the DDS are performed in the control software thus allowing the evaluator immediate bench access to performance results. Finally, a quality printed circuit board that is simple to comprehend and alter, yet provides uncompromised and consistent performance is essential to obtain reliable results.

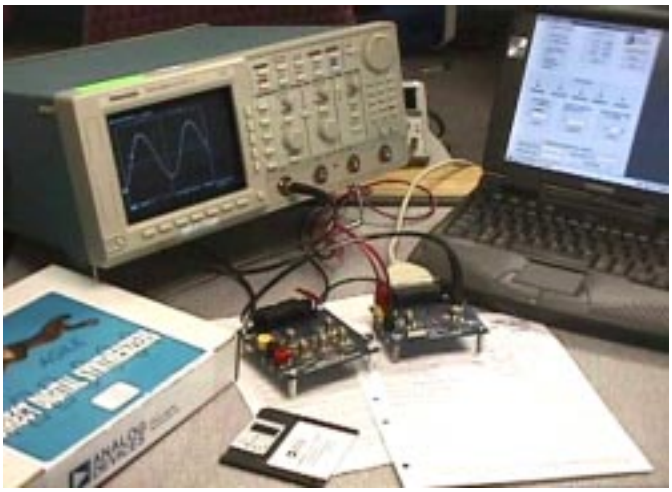


Figure 12-1. Typical Evaluation Kits for DDS Products

Pictured above are two Analog Devices DDS evaluation kits. Included in each kit are device data sheets, operating software, an evaluation board instruction manual, and the assembled evaluation board with DUT included. Although the DDS IC can be operated at very high speeds on the evaluation board, the instructions or programming to the IC are coming from a relatively slow PC printer port. To perform sophisticated high-speed frequency or phase changes such as frequency hopping, spread-spectrum, frequency or phase modulation requires additional user-provided hardware (FPGA, micro-controller or other) and software routines .

Basic Considerations for Bench Evaluation

To get the best performance from an evaluation board, care must be taken when connecting test and measurement devices so that performance is not adversely affected. Oscillators driving the DDS should be equal to or better than the quality you expect to see at the output of the DDS. See Section 5 of this tutorial material for examples showing the effects of good and poor clocking sources on the DDS output quality. Spectrum analyzers should be calibrated and properly configured so that the analyzer input mixer is not overdriven by the DDS signal, which causes erroneous spectral representations. If an output filter is provided on the evaluation board, make certain that no unusual loads (improperly terminated transmission lines, highly-capacitive probes, long leaded test probes, etc.) are placed at the input or output of the filter which would cause an impedance mismatch between the DDS output and the filter.

For those who do not wish to purchase an evaluation board, the data sheets for Analog Devices DDS products contain the PCB layout (usually not-to-scale) and schematic diagrams of the evaluation board. Furthermore, the PCB Gerber files can be provided on disk or via email upon request. For many applications, simply observing the PCB layout and number of layers of the evaluation board is sufficient for designers to be able to duplicate (or improve upon) the electrical performance using in-house PCB resources more closely aligned with the designers end product.

Section 13. Integrating DDS-based Hardware into a System Environment

Note - Please refer to Section 7 in the Analog Devices "High Speed Hardware Design Techniques" seminar book for complete detailed information on best practices for mixed-signal system implementation strategies, PCB layout, and power supply recommendations. Please visit the Analog Devices website at www.analog.com for information on obtaining the seminar handout.

DDS devices are examples of mixed-signal technology, which means they contain both digital and analog signals. High-frequency mixed-signal devices present a particular challenge in system implementation and PCB design. The combination of high-speed clocks and (up to) 12-bit data path resolutions in the sampled output, require a high degree of system and PCB layout consideration in order to protect the integrity of the output signal. Some common problems with DDS-based synthesizers that can be attributed to poor system implementation are:

- Excessive system clock feedthrough
- Degraded wideband spur performance
- Jitter in the output signal
- Degraded narrowband spur performance due to widening of the "skirt" around the fundamental
- Elevated noise floor in the output spectrum

Ground plane

One of the fundamental requirements in implementing a successful high-speed mixed-signal PCB design is to incorporate large, low-impedance ground planes. Low-impedance ground plans acts as a return-path for high-frequency analog and digital signals and reduces the overall level of EMI/RFI emissions. DDS devices typically provide ground connections separately labeled as "digital ground" and "analog ground". It is generally desirable to connect these analog and digital ground pins to a single massive ground plane within the PCB. All IC ground pins should be soldered directly to their corresponding ground planes to minimize series inductance. The IC's power supply pins should be decoupled to the ground plane, as close as possible to the chip, with low-inductance ceramic surface-mount capacitors. If through-hole capacitors must be used, their leads should be less than 1mm in length and Ferrite beads may be employed to compensate for parasitic resonance. Multi-layer PCBs are recommended for "motherboard" designs, with individual layers utilized as a dedicated analog and digital ground planes.

Power supplies

Most DDS devices will also have multiple supply pins labeled as "analog supply" and "digital supply" which identifies the nature of the stage that the individual pin is powering. It is recommended that separate analog and digital power supplies be used to power the associated pins. The preferred analog power supply is a linear supply with ripple limited to <1%. Digital supplies are typically switching supplies with output ripple in the range of up to 5%. It is common practice to utilize 3-terminal regulators to supply the voltage for mixed-signal devices that require non-standards voltage supplies for their analog sections.

Section 14. DDS Product Selection Guide

Analog Devices, Inc., has a wide range of DDS modulators and single-tone synthesizers available. Ranging from 20 MHz to 300 MHz internal clock rates, and DAC resolutions in 10 and 12-bits, this product family is sure to contain the perfect device for the most demanding signal synthesis application.

Please visit our DDS website at www.analog.com/DDS for complete information on our innovative products and to obtain technical information on applications and new product announcements.

ADI also has a broad array of other mixed-signal data converter and DSP solutions for just about any aspect of your electronic system design. Please visit www.analog.com for complete details.

Appendix A - Glossary of Digital Communications Systems-Related Terms

A/D Converter	(also A/D or ADC) Short for analog-to-digital converter. This device convertes real-world analog signals into a digital format that can be processed by a computer. Video-speed A/D converters are those able to digitize video bandwidth signals (greater than 1MHz): some are capable of sampling at rates up to 500 million-samples-per-second (Msps) and beyond. The most common architectures for video-speed A/D converters are "flash" and "subranging."
AC Linearity	A dynamic measurement of how well an A/D performs. In an ideal A/D converter, a pure sine wave on the analog input appears at the digital output as a pure (sampled) sine wave. In the real world, however, spurious signals due to nonlinear distortion within the A/D appear in the digital output. These anomalies are usually combinations of harmonics of the fundamental and intermodulation products, produced when the fundamental and its harmonics beat with the sampling frequency. Only the spurious and harmonically-related signals that fall within the A/D's input bandwidth (half the sampling rate) are generally considered important. AC linearity is usually characterized in terms of harmonic disortion, signal-to-noise ratio, and intermodulation distortion performance.
Acquisition Time	This term relates to sampling A/D's which utilize a track/hold amplifer on the input to acquire and hold (to a specified tolerance) the analog input signal. Acquisition time is the time required by the T/H amp to settle to its final value after it is placed in the track mode.
Active Filter	An active filter is one that uses active devices such as operational amplifiers to synthesize the filter response function. This technique has an advantage at high speeds because the need for inductors (with their poor high-frequency characteristics) is eliminated.
Aliased Imaging	This is a technique, commonly applied to **Direct Digital Synthesis** (DDS), for using intentional aliasing as a source of high-frequency signals. The following spectral plot illustrates a DDS output with a clock frequency of 51MHz and fundamental output of 25.5MHz. Aliased images appear at 32MHz, 70MHz, 83MHz, etc., which can be used as signal sources when isolated with a bandpass filter.

Aliasing	In a sampled data system, the analog input must be sampled at a rate of $F_S > 2F_A$ in order to avoid loss of data (Nyquist Theorem). Adhereing to the Nyquist Theorem prevents in-band "alias" signals, which are beat frequencies between the analog signal and the sampling clock that inherently occur at $F_S \pm F_A$. As the Nyquist limit is exceeded, the aliased signals move within the band of the analog input (DC - $F_S/2$) and create distortion. Likewise, high-frequency noise can also be aliased into the input signal range which mandates low-pass filtering, or anti-alias filtering, on the input of a sampled system. See also **Aliased Imaging**.
Analog Ground	In high-speed data acquisition applications, system ground is generally physically separated into "analog" and "digital" grounds in an attempt to supress digital switching noise and minimize its effect on noise-sensitive analog signal processing circuitry. Input signal conditioners, amplifiers, references, and A/D converters are usually connected to analog ground.
Aperture Delay Time	This term applies to A/D converters and Track/hold amplifiers and defines the time elaspsed from the application of the "hold" (or "encode") command until the sampling switch opens fully and the device actually takes the sample. Aperture delay time is a fixed delay time and is normally not in itself an error source since the "hold" clock edge can be advanced to compensate for it.
Aperture Jitter	Uncertainty, or sample-to-sample of variation, in the aperture delay time. Aperture jitter is a source of error in a sampling system and it determines the maximum slew rate limitation of the sampled analog input signal for a given system resolution.
Asymmetrical Digital	A digital communications application that allows for up to 7MBPS of Subscriber Line data transmission capacity over conventional twisted pair telephone lines. ADSL is a contender for a major piece of the "information highway" pie and it promises to deliver telephone, TV, and data services to your home over the exisiting telephone line.
Asynchronous Transfer Mode	A multiplexing and switching technique that organizes information into fixed-length cells consisting of an identification header field and an information field. The tranfer rate is asynchronous in that the recurrence of cells depends on the instantaneously required bit rate.

Autocorrelation	Multiplication of a signal with a time-delayed replica of itself.
Baseband	The frequency bandwidth of the fundamental signal of interest i.e., the voice, audio, or video signal bandwidth, within a communication system.
Base Station	The central transmitter in a communications system that acts as the cell hub for communicating with handsets and/or mobile units.
Baud Rate	The speed at which data is transmitted measured in symbols-per-second.
Bit Rate	The rate of transfer of information necessary to insure satisfactory reproduction of the information at the receiver.
Buffer Amplifier	A unity gain amplifier used to isolate the loading effect of one circuit from another. Buffer amplifiers are almost always used between the signal source and the input of a high-speed A/D converter.
Cable Telephony	This is the idea of using digital communications techniques to provide enhanced home telephone service via the exisiting home cable-TV connections. The bandwidth of cable is high enough to simultaneously support interactive cable-TV, telephone communications, and on-line data services. In this scenario, the cable connection becomes the primary link to the information highway vs. twisted pair telephone wire or a wireless connection.
Chip	A single frequency output from a frequency hopping signal source.
Chip Rate	In spread spectrum systems, this is the rate at which the pseudo-random noise code is applied. In frequency hopping systems, chip rate is the inverse of the dwell time which the output frequency occupies a single carrier frequency. Also called "chipping rate".
Chirp	Pulsed frequency modulation scheme in which a carrier is swept over a wide frequency band during a given pulse interval.
Cross-correlation	The degree of agreement between two unlike signals.
D/A Converter	(also D/A or DAC). Short for digital-to-analog converter, this is a device that changes a digitally-coded word into its "equivalent" quantized analog voltage or current. Just like the A/D device, there are very high-speed D/A's available, capable of converting at data rates up to 1GHz.

Differential Nonlinearity	(also DNL) In an ideal D/A and A/D converter, any two adjacent digital codes should result in measured ouput (or input) values that are exactly one LSB apart. Any positive or negative deviation in the measured "step" from the ideal differences is called differential non-linearity. It's expressed in (sub) multiples of 1 LSB. DNL errors more negative than -1 LSB can result in a non-monotonic response in a D/A and missing codes in an A/D.
Digital Downconversion	(also direct-IF-to-digital conversion) This refers to a demodulation technique for sampling an intermediate frequency (IF) signal with a wide-bandwidth A/D whose sampling rate is equivalent to the local oscillator frequency (<IF frequency). In this super-Nyquist application the A/D serves as the mixer stage and its digital output data is a beat frequency; the modulation data can be recovered with a DSP stage.
Digital Filtering	The process of smoothing, spectrally shaping, or removing noise from a signal has traditionally been accomplished with analog components. With the advent of high-speed DSP products, now filtering can effectively and economically be accomplished in the digital domain. Digital filters are basically mathematical functions that are performed on the digital data stream and their characteristics can be altered under software control which adds to their overall flexibility. Finite Impulse Response (FIR) and Infinite Inpulse Response (IIR) are examples of digital filter functions.
Direct Digital Synthesis	(DDS) A process by which you can digitally generate a frequency-agile, highly-pure sinewave, or arbitrary waveform, from an accurate reference clock. The digital output waveform is typically tuned by a 32-bit digital word which allows sub-Hz frequency agility. The DDS's frequency output is normally reconstructed with a high-speed, high- performance D/A to generate an analog output signal. The ability to add internal functions such as phase modulation, amplitude modulation, digital filtering, and I&Q outputs, are making DDS devices attractive for digital communication applications. They serve in capacities such as modulators, local oscillators, and clock detect/recovery circuits.
Dither	The technique of adding controlled amounts of noise to a signal to improve overall system loop control, or to smear quantizing error in an A/D converter application.

Dynamic Range	The ratio of the maximum output signal to the smallest output signal that can be processed in a system, usually expressed logarithmically in dB. Dynamic range can be specified in terms of harmonic distortion, signal-to-noise ratio, spurious-free dynamic range, or other AC input-based performance criteria
ENOBs	Stands for "Effective Number of Bits". ENOB's are a measure of an A/D's dynamic performance as compared to that of a theoretically perfect A/D transfer function. ENOB's are calcluated by the formula: (ENOB=SNR Actual- 1.76dB)/6.02. An high-speed A/D with 10-bits of resolution typically will <9 ENOBS of dynamic performance at a Nyquist analog input bandwidth.
FFT	Fast Fourier Transform. A computationally efficient mathematical technique which converts digital information from the time domain to the frequency domain for rapid spectral analysis. FFT's generally utilize a "time weighting" function to compensate for data records with a non-integer number of samples; some poplular weighting functions are Hanning Window and 4-term Blackman-Harris, which are presented in Appendix 4 of this glossary.
Frequency Hopping System	Carrier frequency shifting in discrete increments in a pattern dictated by a code sequence. The transmitter jumps from frequency-to-frequency within some predetermined set: the order of frequency hops is determined by a code sequence which, in turn, is detetected and followed by the receiver.
Frequency Shift Keying	A modulation scheme that shifts between two frequencies to represent a "1" or "0" state of data transmission.
Glitch	A spike caused by the skew (difference in turn-on/turn-off time) of switches or logic. Glitches are a troublesome source of error in high-speed D/A converters and they are most prevalent at the mid-scale switching location, when all digital input bits are switching. Glitch energy is specified in picovolt-seconds which describes the area under the voltage-time curve at its worst case occurence.
Group Delay	Distortion resulting from nonuniform speed of transmission of the various frequency components of a signal through a transmission medium, specifically, the propagation delay of a lower frequency is different from that of a higher frequency. This creates a time-related "delay distortion" error.

Heterodyne	A process by which two signals are mixed for the purpose of cross-frequency translation. Integral Nonlinearity (INL) This term describes the absolute accuracy of a converter. It is the maximum deviation, at any point in the transfer function, of the converter's output from its ideal value.
I&Q	In-phase and Quadrature - A modulation technique whereby signal information is derived from a carrier frequency at its 0° and 90° phase angles.
Jitter	Unwanted variations in the frequency or phase of a digital or analog signal.
Mixer	Circuit block used to translate signals from one frequency to another.
Multipath Propagation	A transmission path anomaly that acts as a time-varying source of signal non-linearity. Multipath can distort or reduce a received signal to the point of unreliable reception. In television, multipath is manifested as image "ghosting".
Nyquist Theorum	This theorum says that if a continuous bandwidth-limited signal contains no frequency components higher than f_C then the original signal can be recovered without distortion if its is sampled at a rate of at least 2 f_C.. This theorum applies to A/D converter applications as well as data transmission density over limited-bandwidth channels.
Orthogonal	Term used to signify that two signals (or signal attributes) are mutually transparent and non-interfering with each other. Frequency and amplitude modulation are orthogonal signal attributes.
Packet	A digital communications technique involving the transmission of short bursts of data in a protocol format that contain addressing, control, and error-checking information, along with the field information, in each transmission burst.
Phase Locked Loop	(PLL) A circuit containing a voltage-controlled oscillator whose output phase or frequency can be "steered" to keep it in sync with a reference source. A PLL circuit is generally used to lock onto and "up-convert" the frequency of a stable source.

Phase Noise	The amount of phase noise energy contained in a frequency carrier. Specified in dB/Hz, phase noise amplitude is usually characterized and plotted in 1Hz increments, offset from the carrier. The following illustrates phase noise from various frequencies synthesized by a DDS device clocked at 50MHz. [Insert phase noise plot]
Programmable Gain	(PGA) An amplifier with an analog- or digitally-controlled gain Amplifier function. A PGA can be used in front of an A/D converter to effectively increase its dynamic range.
Pulse Code Modulation	A method of quantizing audio-range analog signals into a digital form for transmission in digital communications systems, or for processing in DSP. Effectively the same as Analog-to-Digital conversion.
Pseudo-noise	Any group of code sequences that exhibit a noise-like characteristic.
Phase Shift Keying	(PSK) A digital modulation technique whereby the phase of a carrier frequency is shifted to represent a digital "1" or "0" state. In "Quadri-phase-shift keying" systems, the phase angle locations of 0°, ±90°, and 180°, are used as reference points to represent sixteen possible digital states (2^4).
Quadrature Amplitude	(QAM) This communication scheme involves the modulation of a carrier by two different signals. One signal modulates the carrier (I) and the other signal modulates the carrier shifted by 90° (Q). The two modulated carriers are then summed and transmitted as a single I&Q modulated carrier. The receiver decodes the I&Q channels and demodulates them 90° apart. QAM lends itself to the transmission of data in a digital format by assigning discrete levels to the two signal inputs which creates a "constellation" of possible digital word combinations on the I&Q graph. The following illustrates a 16-QAM constellation (4 levels of input on the I channel/4 levels of input on the Q channel).
Sin(X)/X	The output of a D/A converter is a series of quantized levels that represent an analog signal whose amplitude is determined by the sin (X)/X response. At higher output frequencies, a D/A converter application may require a sin(X)/X compensation filter to normalize its output amplitude.

Spread Spectrum	This communications technique has been used in secure military systems for a number of years and is now becoming popular in commercial systems. This format involves transmitting information which has been multiplied by a pseudo-random noise (PN) sequence which essentially "spreads" it over a relatively wide frequency bandwidth. The receiver detects and uses the same PN sequence to "despread" the frequency bandwidth and decode the transmitted information. This communications technique allows greater signal density within a given transmission bandwidth and provides a high degree of signal encrytion and security in the process.
Spurious-free Dynamic	(SFDR) This refers to the range between the highest level of the fundamental Range frequency and the highest level of any spurious, or harmonically- related, signal. SFDR is expressed in dB.
Wavelet	A mathematical algorithm that is used to efficiently compress and decompress the phase & frequency information that is contained in a transmitted signal.
Wireless Telephony	The idea of replacing the conventional twisted pair telephone service to the home with a wireless RF connection. High-speed digital communications techniques would be utilized to allow enhanced telephone, multimedia, and data services to be transmitted over the airwaves to the home subscriber. In this scenario, the link to the information highway will be wireless.

Appendix B - Digital Communications Systems-Related Acronyms

ADC	Analog-to-Digital Converter
ADSL	Asynchronous Digital Subscriber Loop
AGC	Automatic Gain Control
AM	Amplitude Modulation
AM-PSK	Amplitude Modulation with Phase-Shift Keying
AMPS	Advanced Mobile Phone System
ASK	Amplitude-Shift Keying
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate (Ratio)
BPSK	Binary-Phase-Shift-Keying
CCITT	International Telephone and Telegraph Consultive Commitee
CDMA	Code-Division Multiple Access
CDPD	Cellular Digital Packet Data
CMOS	Coated Metal Oxide Semiconductor
CODEC	Coder/decoder
CSMA	Carrier-Sense Multiple Access
CSMA/CD	Carrier-Sense Multiple Access with Collision Detection
CT-2	Cordless Telephone 2
DAC	Digital-to-Analog Converter
DBS	Direct Broadcast Satellite
DCPSK	Differentially Coherent Phase Shift Keying
DDS	Direct Digital Synthesis

DECT	Digital European Cordless Telephone
DFT	Discrete Fourier Transform
DM	Delay Modulation
DNL	Differential Non-Linearity
DPCM	Differential Pulse-Code Modulation
DPSK	Differential Phase-Shift Keying
DSL	Asymmetrical Digital Subscriber Line
DSK	Downstream Keyer
DSMS	Direct Sequence Modulated System
DSX	Digital Signal Cross-Connect
DQPSK	Differential Quadrature-Phase-Shift-Keying
DWT	Discrete Wavelet Tone
DSP	Digital Signal Processing
DWMT	Discrete Wavelet Multi-tone
ENOB	Effective Number of Bits
ESMR	Enhanced Specialized Mobile Radio
FDD	Frequency-Division Duplex
FDMA	Frequency-Division Multiple Access
FDX	Full Duplex
FFT	Fast Fourier Transform
FIR	Finite-Impulse-Response Filter
FM	Frequency Modulation
FSK	Frequency Shift Keying

GFSK	Gaussian Frequency-Shift-Keying
GMSK	Gaussian Minimum Shift-Keying
GPS	Global Positioning Satellite
GRI	Group Repetition Interval
GSM	Global System for Mobile Communications
HDSL	Highspeed Digital Subscriber Line
HDX	Half Duplex
HFC	Hybrid Fiber Coax
HSDS	High-speed FM Subcarrier Data System
IIR	Infinite Impulse Response
IMD	Intermodulation Distortion
INL	Integral Non-Linearity
I&Q	In-phase and Quadrature
ISI	Intersymbol Interference
JCT	Japanese Cordless Telephone
JDC	Japanese Digital Cellular (now PDC)
LAN	Local Area Network
LAP	Link Access Protocol
LNA	Low-Noise-Amplifier
LOS	Line-of-Sight
MAC	Medium Access Control
MAS	Multiple-Address-Services
MFM	Modified Frequency Modulation

Modem	Abbreviation for modulator-demodulator
MQPSK	Modified Quadri-Phase-Shift-Keying
MSS	Mobile Satellite Services
NADC	North American Digital Cellular (IS-136)
NRZ	Non-return-to-zero
OFDM	Orthoganol Frequency Division Modulation
OOK	On-Off Keying
OQPSK	Offset Quadri-Phase-Shift-Keying
PAM	Pulse Amplitude Modulation
PBX	Private-Branch-Exchange
PCN	Personal Communications Network
PCM	Pulse Code Modulation
PCS	Personal Communications Services
PCIA	Personal Communcations Industry Association
PDC	Personal Digital Cellular (formerly JDC)
PDM	Pulse Duration (Density) Modulation
PFM	Pulse Frequency Modulation
PGA	Programmable Gain Amplifier
PHS	Personal Handy Phone System
PLL	Phase Locked Loop
PN	Pseudo-Random Noise
PM	Phase (Pulse) Modulation
PRBS	Pseudo-Random Bit Stream

PSK	Phase-shift-Keying
PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
QPSK	Quadri-Phase-Shift-Keying
RF	Radio Frequency
RFI	Radio Frequency Interference
RFID	RF Identification.
RSSI	Received Signal Strength Indication
Rx	Receiver
RZ	Return to Zero
SAW	Surface Acoustic Wave
SCPDM	Suppressed Clock Pulse Duration Modulation
SDLIC	Synchronous Data Link Control
SFDR	Spurious-Free Dynamic Range
SINAD	Signal-to-Noise and Distortion
SMR	Specialized Mobile Radio
SMT	Surface Mount Technology
SNR	Signal-to-Noise Ratio
SONET	Synchronous Optical Network
SPP	Sequenced Packet Protocol
SS	Spread Spectrum
SSB	Single Sideband
SSHE	Spread Spectrum Headend

SQPSK	Staggered Quadriphase Shift Keying
SWR	Standing Wave Ratio
TACS	Total Access Communication System
TDD	Time-Division Duplex
TDM	Time Division Multiplexing
TDMA	Time-Division Multiple Access
TTL	Transistor-to-Transistor Logic
Tx	Transmitter
UPS	Uninterruptible Power Supply
UTC	Universal Time Code (Coordinated Universal Time)
VHF	Very High Frequency
VLF	Very Low Frequency
VSWR	Voltage Standing Wave Ratio
WAN	Wide Area Network
WAFU	Wireless Fixed Access Unit
WDM	Wavelength Division Multiplexing
WLAN	Wireless Local Area Network
WLL	Wireless Local Loop
WWW	World Wide Web
Z	Impedance

Appendix C - An FM Modulator using DDS

Appendix D - Pseudo - Random Generator

Pseudorandom Binary Sequence (PRBS) Generator Basics

The problem at hand is to devise a simple and efficient means to randomly generate a sequence of random bits (i.e., a sequence of 1's and 0's). The method behind generating random bits is based on the theory of *modulo 2 primitive polynomials*.

In general, an n^{th} degree polynomial takes the form:

$$a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_2 x^2 + a_1 x + a_0$$

Or, in more compact form:

$$\sum_{k=0}^{n} a_k \, x^k$$

where a_k are the coefficients of the polynomial. A modulo 2 polynomial is simply a polynomial in which the coefficients are constrained to two possible values: 0 and 1. An example of a modulo 2 polynomial (4^{th} degree) is shown below.

$$x^4 + x^3 + x^1 + 1$$

Note that it is not necessary to explicitly display the coefficients. Since the coefficients are constrained to 0 or 1, a polynomial term is either present (coefficient of 1) or absent (coefficient of 0). In the example above, all coefficients are 1 except for the 2^{nd} degree term; hence its absence.

In the opening paragraph reference was made to *primitive* polynomials. A modulo 2 polynomial is *primitive* if it cannot be factored. It should be noted that the polynomial example shown earlier is <u>not</u> primitive. It can be factored as follows:

$$x^4 + x^3 + x^1 + 1 = (x^3 + 1)(x + 1)$$

An example of a *primitive* 4^{th} degree polynomial is shown below. Notice that it cannot be factored.

$$x^4 + x^1 + 1$$

It is interesting to note that the coefficient of the 0^{th} term of a modulo 2 *primitive* polynomial is always 1. That is, a modulo 2 primitive polynomial will always end with "… + 1". The reason should be clear. If the polynomial does not end with a 1, then it must end with some k^{th} degree term ($k < n$). Since the k^{th} degree term is the lowest term in the polynomial, then x^k can be factored out of the polynomial. This makes the polynomial factorable by x^k, thus rendering it *non*-primitive.

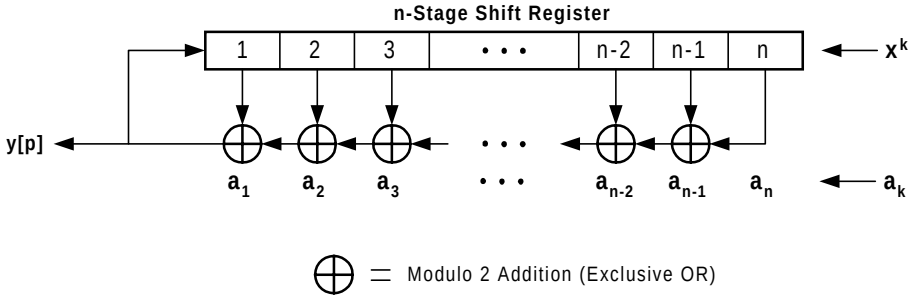
One of the properties of PRBS generators is that the "random" sequence eventually repeats. Hence the term, "pseudorandom". The number of bits that a PRBS can generate before repeating defines the *length*, L, of the PRBS generator. It turns out that there is a relationship between the length, L, of a PRBS generator and degree, n, of the primitive polynomial used to define it. That relationship is shown below:

$$L = 2^n - 1.$$

The first step in designing a PRBS generator is to determine the required length, L. This establishes the minimum degree of the primitive polynomial that must be employed in the design.

The second step in designing a PRBS is to select a suitable primitive polynomial of appropriate degree. Finding primitive polynomials of a particular degree can be a challenge, but there is help. Published lists are available from a number sources. For example, in Ref. [2] primitive polynomials for $n \leq 100$ can be found. Also, a nearly inexhaustible supply of primitive polynomials can be found with a search of the Internet.

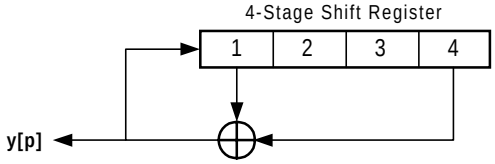
The form of the primitive polynomial relates directly to the design of a *linear feedback shift register* (**LFSR**). The general form of an LFSR is shown below. The output, y[p], is the pseudorandom binary sequence.



n-Stage Linear Feedback Shift Register (LFSR)

Note that each stage of the shift register corresponds to one of the x^k terms of a polynomial, while each associated XOR operation corresponds to one of the a_k coefficients. The coefficients serve as feedback elements. This allows for a direct correlation between the primitive polynomial and the LFSR. It should be noted that for every 0 coefficient of the primitive polynomial, the associated XOR in the feedback path of the LFSR can be removed. The underlying reason is that in modulo 2 addition we have: $0 \oplus q = q$. The implication is that the XOR of 0 with a value yields the same value, rendering the operation superfluous.

With the knowledge of how the primitive polynomial and LFSR are related it is possible to design a PRBS of any desired degree. For example, using the 4^{th} degree primitive polynomial listed previously the corresponding PRBS generator is shown below.



4-th Order PRBS

The above PRBS generator has a pseudorandom sequence of length, $L = 2^4 - 1 = 15$. It should be mentioned at this point that a PRBS fashioned in this manner must be given a non-zero "seed" value in order to guarantee start up. That is, the shift register must be preloaded with a non-zero value. Notice that if the shift register is initialized with all 0's, y[p] will always be 0. Thus, any hardware implementation must ensure that the shift register starts with some non-zero value. It is interesting to note that the choice of the seed value is immaterial. The n-bit initial seed value will not repeat until L bits have been generated at y[p].

A pseudorandom binary sequence generated in the manner just described possesses some interesting properties. Some of these properties are listed below.

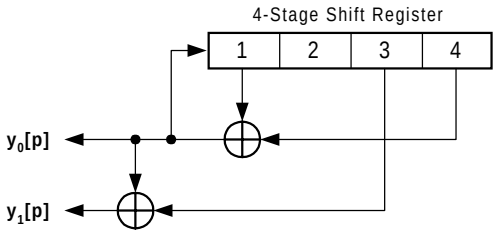
1. Any group of L consecutive bits of y[p] will contain 2^{n-1} ones and $2^{n-1} - 1$ zeroes.

2. In any group of L consecutive bits of y[p], if we slide an n-bit window along the sequence, each n-bit word thus obtained will appear exactly once (excluding the "all zeroes" word, which does not occur at all).

3. The modulo 2 sum of a PRBS and any shifted version of itself, produces another shifted version of itself.

It is important to mention that a PRBS is only useful for generating a random sequence of individual bits. It is not useful for generating random M-bit words. In other words, suppose that it is desired to generate a random sequence of 4-bit numbers. One might be tempted to take groups of 4 bits generated by a PRBS and consider them as a random sequence of 4-bit numbers. Such is not the case. The solution is to generate 4 independent PRBS streams and concatenate their outputs to produce random 4-bit numbers. As a matter of fact, it is preferable that each of the PRBS generators used in a multi-bit random number generator be of a different order.

It turns out, that Property #3 above can be used to generate multi-bit random numbers using a single LFSR. The diagram below is a 2-bit random number generator using a modification of the 4^{th} order PRBS designed earlier.



2-Bit 4^{th} Order Pseudorandom Number Generator

Here, $y_0[p]$ is the original pseudorandom bit sequence. Note that $y_1[p]$ is the modulo 2 sum of $y_0[p]$ (the original) and a tap of the 3rd stage of the shift register. But the 3rd stage of the shift register is simply a replica of $y_0[p]$ delayed by 3. Thus, Property #3 dictates that $y_1[p]$ is a shifted version of $y_0[p]$. Therefore, we can think of $y_0[p]$ and $y_1[p]$ as two separate 4th order pseudorandom sequences but with differing seed values. This makes the two sequences useful for generating the individual bits of a 2-bit number. Hence, we have the makings of a 2-bit random number generator. This concept can be extended to generate pseudorandom numbers with any desired bit width.

References:

[1] Gibson, J. D., 1993, Principles of Digital and Analog Communications, Prentice-Hall, Inc.

[2] Press, H. P., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 1992, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press.

Appendix E - Jitter

ANALOG DEVICES Technical Note
ONE TECHNOLOGY WAY P.O.BOX 9106 NORWOOD, MASSACHUSSETTS 02062-9106
781/329-4700

Jitter Reduction in DDS Clock Generator Systems

by: Rick Cushing, HSC Applications Engineer

One of the most frequently asked questions regarding DDS clock generator applications is how to minimize clock edge jitter. Here are some basic steps in assuring best jitter performance:

1. Use a stable reference clock for the DDS
2. Filter the DDS output to reduce all non-harmonic spurs to at least –65 dBc.
3. Drive the comparator inputs differentially.
4. Provide plenty of comparator over-drive (at least 1 volt p-p).
5. Provide low Z inputs to the comparator to suppress high Z noise sources.
6. Use an external comparator or divider for very demanding applications.
7. Avoid using slow slew rate signals.
8. Use spur reduction techniques.

Using a stable reference clock to the DDS is obvious since jitter in = jitter out. Filtering of the DDS output requires some elaboration. Bandpass filtering is best since spurs may reside well below and above the fundamental output. If wideband output is required, then a low pass filter is the only choice, but jitter performance will be compromised. To make filtering easier, keep the system clock frequency as high as possible to distance the image spurs as far as possible from the fundamental.

Driving the comparator inputs with a low impedance, differential, 1 V p-p input signal is “as good as it gets”. The low impedance input discourages extraneous noise and comparator “kick-back”. The merits of differential inputs include common-mode noise rejection and 2X the input slew rate. A passive broadband 1:1 RF transformer is useful in changing from single-ended to differential configuration; however, it will not pass dc so some means of biasing to the comparator input range may be needed. Sufficient comparator input overdrive keeps output dispersion (variation in propagation delay) to a minimum and promotes decisive switching.

Use of an off-chip comparator is recommended since the hostile (noisy) DDS environment degrades jitter performance. In choosing an external high-speed comparator, look for good PSRR specifications, proper output logic levels, low dispersion and single supply operation.

5 MHz and above: These are the easiest signals to handle due to their fast slew rates. A reason-able jitter figure for these frequencies is about 75 ps p-p using the above techniques. Higher frequency signals have higher harmonic distortion. This is not bad until the harmonic is “aliased” back into the passband to the comparator where it becomes a non-harmonically related product that will increase jitter.

Aliased harmonics can be reduced by passing the filtered fundamental and spurs to a passive frequency mixer, upconverting to a much higher frequency and then dividing down (example VHF divider is a Mitel SP8402) to your desired output frequency. The frequency division process does two very desirable things:

1. Reduces spurious components by 20LOG(N) – where N is the division ratio.
2. Gives you a square wave output which may eliminate the need for a comparator if the divider has suitable logic levels.

Below 5 MHz: *Much more difficult to get good jitter performance due to the slow slew rate of the fundamental. Best results are obtained by outputting a much higher frequency than needed and then dividing down to your desired output. The same benefits as 1 and 2 above will apply. This method of spur reduction is especially useful when extremely low frequencies with low jitter are needed. Depending on external frequency division ratios, jitter levels approaching 75 ps p-p seem obtainable. Without frequency division, a 1 kHz sine wave could produce 10 ns p-p jitter just due to the slow slew to the comparator.*