DDS v5.0

DS246 April 28, 2005 **Product Specification**

Features

- Drop-in module for Virtex™, Virtex-E, Virtex-II, Virtex-II Pro, Virtex-4, Spartan™-II, Spartan-IIE, Spartan-3, and Spartan-3E FPGAs
- Sine, Cosine, or quadrature outputs
- Look-up table can be allocated to distributed or block memory
- Phase dithering and Taylor series correction option provide high dynamic range signals using minimal amount of FPGA resources. SFDR range is 18 dB to 115 dB
- Phase dithering removes the spectral line structure associated with conventional phase truncation waveform synthesis architectures
- Support for 1 to 16 independent channels
- High-precision synthesizer with fine frequency resolution ($\Delta f = 0.02$ Hz @ fclk = 100 MHz, 32-bit phase accumulator)
- 4 to 32-bit two's complement output sample precision
- Optional phase offset capability providing support for multiple synthesizers with precisely controlled phase differences
- Simple fixed-output frequency option
- Uses relationally placed macro (RPM) mapping and placement technology for maximum and predictable performance
- Incorporates Xilinx Smart-IP™ technology for utmost parameterization and optimum implementation
- For use with v7.1i and later of the Xilinx CORE Generator™ system

Applications

- Digital radios and modems
- Software-defined radios (SDR)
- Digital down/up converters for cellular and PCS base stations
- Waveform synthesis in digital phase locked loops
- Generating injection frequencies for analog mixers

General Description

Direct digital synthesizers (DDS), or numerically controlled oscillators (NCO), are important components in many digital communication systems. Quadrature synthesizers are used for constructing digital down and up converters, demodulators, and implementing various types of modulation schemes, including PSK (phase shift keying), FSK (frequency shift keying), and MSK (minimum shift keying). A common method for digitally generating a complex or real valued sinusoid employs a look-up table scheme. The look-up table stores samples of a sinusoid. A digital integrator is used to generate a suitable phase argument that is mapped by the look-up table to the desired output waveform. A simple user interface accepts system-level parameters such as the desired output frequency and spur suppression of the generated waveforms.

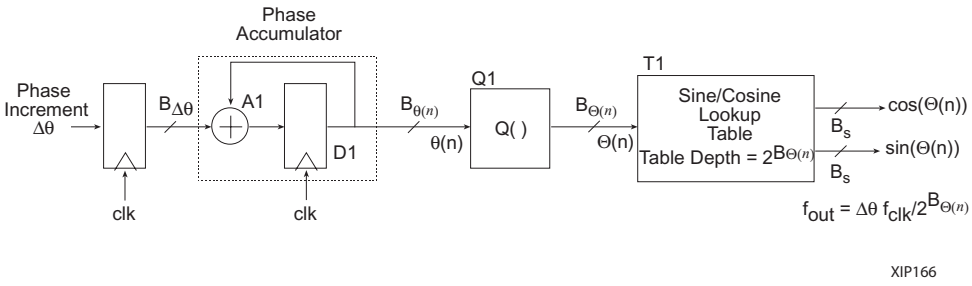

Figure 1: **Phase Truncation DDS (A simplified view of the DDS core)**

Theory of Operation

A high-level view of the DDS Core is presented in Figure 1. The integrator (components D1 and A1) computes a phase slope that is mapped to a sinusoid (possibly complex) by the look-up table T1. The quantizer Q1, which is simply a slicer, accepts the high-precision phase angle $\Theta(n)$ and generates a lower precision representation of the angle denoted as $\Theta(n)$ in the figure. This value is presented to the address port of a look-up table that performs the mapping from phase-space to time.

The fidelity of a signal formed by recalling samples of a sinusoid from a look-up table is affected by both the phase and amplitude quantization of the process. The length and width of the look-up table affect the signal's phase angle resolution and the signal's amplitude resolution respectively. These resolution limits are equivalent to time base jitter and to amplitude quantization of the signal, and add spectral modulation lines and a white broad-band noise floor to the signal's spectrum.

Direct digital synthesizers use an addressing scheme with an appropriate look-up table to form samples of an arbitrary frequency sinusoid. If an analog output is required, the DDS presents these samples to a digital-to-analog converter (DAC) and a low-pass filter to obtain an analog waveform with the specific frequency structure. Of course, the samples are also commonly used directly in the digital domain. The look-up table traditionally stores uniformly spaced samples of a cosine and a sine wave. These samples represent a single cycle of a length $N = 2^{B_{\Theta(n)}}$ prototype complex sinusoid and correspond to specific values of the sinusoid's argument $\Theta(n)$ as shown in Eq.(1).

$$\Theta(n) = n\frac{2\pi}{N} \tag{1}$$

where *n* is the time series sample index.

Quarter wave symmetry in the basis waveform can be exploited to construct a DDS that uses shortened tables. In this case, the two most significant bits of the quantized phase angle $\Theta(n)$ are used to perform quadrant mapping. This implementation results in a more area efficient implementation because the memory requirements are minimized: either fewer FPGA block RAMs or reduced distributed memory. Based on the Core customization parameters, the DDS core will automatically employ quarter-wave symmetry when appropriate[1].

Output Frequency

The output frequency, f_{out}, of the DDS waveform is a function of the system clock frequency f_{clk}, the number of bits $B_{\Theta(n)}$ in the phase accumulator and the phase increment value $\Delta\theta$. That is, $f_{out} = \mathrm{f}\left(f_{clk}, B_{\Theta(n)}, \Delta\theta\right)$.

Output frequency in Hertz is defined as

$$f_{out} = \frac{f_{clk}\Delta\theta}{2^{B_{\Theta(n)}}}\ \mathrm{Hz} \tag{2}$$

For example, if the DDS parameters are

$$\begin{aligned} f_{clk} &= 120\ \mathrm{MHz} \\ B_{\Theta(n)} &= 10 \\ \Delta\theta &= 12_{10} \end{aligned} \tag{3}$$

the output frequency will be

$$\begin{aligned} f_{out} &= \frac{f_{clk}\Delta\theta}{2^{B_{\Theta(n)}}}\ \mathrm{Hz} \\ &= \frac{120\times10^{6}\times12}{2^{10}} \\ &= 1.406250\ \mathrm{MHz} \end{aligned} \tag{4}$$

The phase increment value required to generate an output frequency f_{out} Hz is

$$\Delta\theta = \frac{f_{out}\,2^{B_{\Theta(n)}}}{f_{clk}} \tag{5}$$

Frequency Resolution

The frequency resolution Δf of the synthesizer is a function of the clock frequency and the number of bits $B_{\theta(n)}$ employed in the phase accumulator. The frequency resolution can be determined using the following equation

$$\Delta f = \frac{f_{clk}}{2^{B_{\theta(n)}}} \tag{6}$$

For example, for the DDS parameters

$$\begin{aligned} f_{clk} &= 120\ \mathrm{MHz} \\ B_{\theta(n)} &= 32 \end{aligned} \tag{7}$$

the frequency resolution is

$$\begin{aligned} \Delta f &= \frac{f_{clk}}{2^{B_{\theta(n)}}} \\ &= \frac{120\times10^{6}}{2^{32}} \\ &= 0.0279396\ \mathrm{Hz} \end{aligned} \tag{8}$$

Phase Increment

The phase increment is an unsigned value. The phase increment term $\Delta\theta$ defines the synthesizer output frequency. Consider a DDS with the following parameterization

1. For very short tables, FPGA logic resources are actually minimized by storing a complete cycle. The user is not required to make any design decisions in this context; the CORE Generator will always produce the smallest core possible.

$$f_{clk} = 100 \text{ MHz}$$
$$B_{\theta(n)} = 28$$
$$B_{\Theta(n)} = 12 \quad (9)$$

To generate a sinusoid with frequency $f_{out} = 19$ MHz, the required phase increment would be

$$\Delta\theta = \frac{f_{out} 2^{B_{\Theta(n)}}}{f_{clk}} = \frac{19 \times 10^6 \times 2^{12}}{100 \times 10^6} = 778.24 \quad (10)$$

Spectral Purity Considerations

The fidelity of a signal formed by recalling samples of a sinusoid from a look-up table is affected by both the phase and amplitude quantization of the process. The length and width of the look-up table affect the signal's phase angle resolution and the signal's amplitude resolution respectively. These resolution limits are equivalent to time base jitter and to amplitude quantization of the signal and add spectral modulation lines and a white broad-band noise floor to the signal's spectrum.

In conjunction with the system clock frequency, the phase accumulator width determines the frequency resolution of the DDS. The accumulator must have a sufficient field width to span the desired frequency resolution. For most practical applications, a large number of bits are allocated to the phase accumulator in order to satisfy the system frequency resolution requirements. By way of example, if the required resolution is 1 Hz, and the clock frequency is 100 MHz, the required field width of the accumulator is

$$B_{\theta(n)} = \log_2\left\lceil \frac{f_{clk}}{\Delta f} \right\rceil = \left\lceil \log_2 \frac{100 \times 10^6}{1} \right\rceil = \lceil 26.5754 \rceil = 27 \text{ bits} \quad (11)$$

where $\lceil\ \rceil$ denotes the ceiling operator. Due to excessive memory requirements, the full precision of the phase accumulator cannot be used to index the sine/cosine look-up table. A quantized (or truncated) version of the phase angle is used for this purpose. The block labeled Q1 in the phase truncation DDS, Figure 1, performs the phase angle quantization. The lookup table can be located in block or distributed memory.

Quantizing the phase accumulator introduces time base jitter in the output waveform. As shown in Eq. (12), this jitter results in undesired phase modulation that is proportional to the quantization error.

$$\Theta(n) = \theta(n) + \delta\theta(n)$$
$$e^{j\Theta(n)} = e^{j[\theta(n)+\delta\theta(n)]} = e^{j\theta(n)} e^{j\delta\theta(n)}$$
$$e^{j\Theta(n)} \approx e^{j\theta(n)}\left[1 + j\delta\theta(n)\right] \approx e^{j\theta(n)} + j\delta\theta(n) e^{j\theta(n)} \quad (12)$$

Figure 2 shows the look-up table addressing error, complex output time-series and the spectral domain representation of the output waveform produced by the DDS structure shown in Figure 1. The normalized frequency for this signal is 0.022 Hz, which corresponds to phase accumulation steps of 7.92 degrees per output sample. The angular resolution of the 256-point look-up table is 360 / 256 or 1.40625 degrees per address, which is equivalent to 7.92 / 1.40625 or 5.632 addresses per output sample. Since the address must be an integer, the fractional part is discarded and the resultant phase jitter is the cause of the spectral artifacts. Figure 3 provides an exploded view of the spectral plot in Figure 2(c).

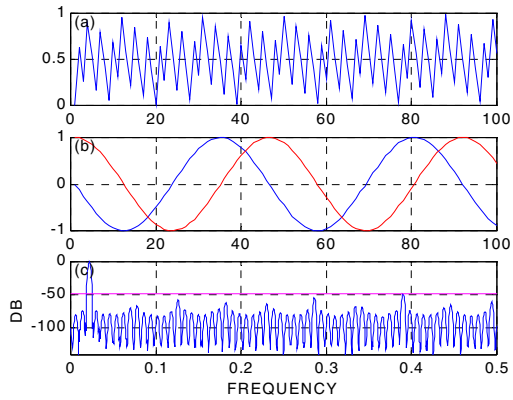

Figure 2: **Phase truncation DDS. $f_{out} = 0.022$ Hz, table depth = 256 12-bit precision samples. (a) Phase angle addressing error; (b) Complex output time series; (c) Output spectrum.**

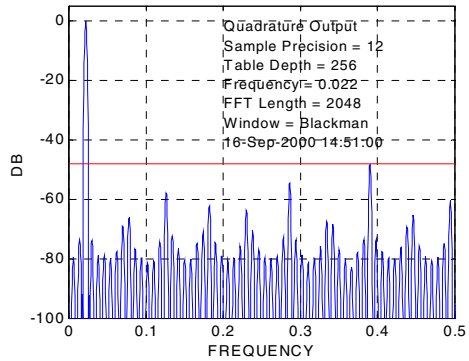

Figure 3: **Phase truncation DDS. $f_{out} = 0.022$ Hz, table depth = 256 12-bit precision samples. Exploded view of Figure 2 (c).**

We make two observations related to the phase jitter structure level. Observe that the fractional part of the address count is a periodic (sawtooth) error sequence which is responsible for the harmonic rich (and aliased) low-level phase modulation evident in Figure 3. We also note that the peak distortion level due to incidental phase modulation is approximately 48 dB below the desired signal level, which is consistent with 6 dB/bit of address space. Put another way, if *S* dB of spur suppression is required in the output waveform, as referenced to the 0 dB primary tone, the DDS lookup table must support at least $\lceil S/6 \rceil$ address bits. For example, if *S* = 70 dB, which means that the highest spur will be 70 dB below the main signal, then the minimum number of address bits for the lookup table is $\lceil 70/6 \rceil$ 12 bits; that is, a 4096-deep table.

Figure 4 and Figure 5 demonstrate the performance of a similar DDS to the one presented in Figure 2 but in this example 16-bit precision output samples have been used. Observe that the highest spur is still at the –48 dB level, and allocating 4 additional bits to the output samples has not contributed to any further spur reduction. For a phase truncation DDS, the only option to further reduce the spur levels is to increase the depth of the look-up table.

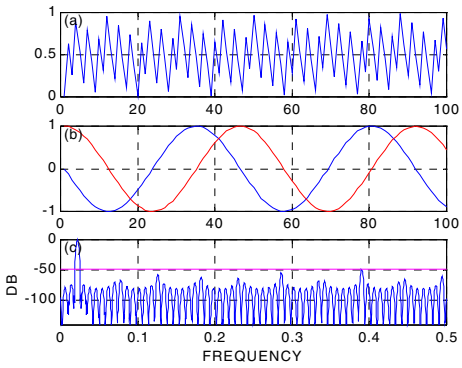

Figure 4: **Phase truncation DDS. $f_{out} = 0.022$ Hz, table depth = 256 16-bit precision samples. (a) Phase angle addressing error. (b) Complex output time series. (c) Output spectrum.**

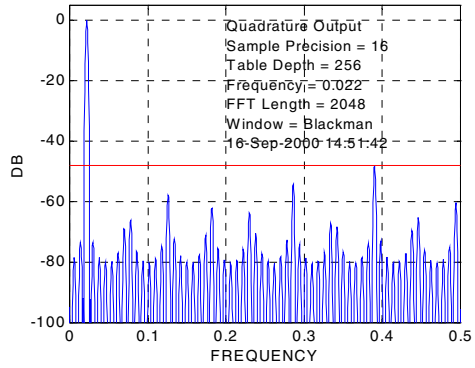

Figure 5: **Phase truncation DDS. $f_{out} = 0.022$ Hz, table depth = 256 16-bit precision samples. Exploded view of Figure 4 (c).**

Further examples illustrating the performance of various DDS configurations are shown in Figure 6 through Figure 15. The configuration details are annotated on the plot. For some of these examples, the synthesized frequency has been swept across a small range of the available output bandwidth. For these cases, the sweep start frequency, stop frequency, frequency increment δf and the number of tones in the sweep interval (*Num Tones*) is indicated. The analysis transform length and window function applied to the output time series is also indicated on the plots.

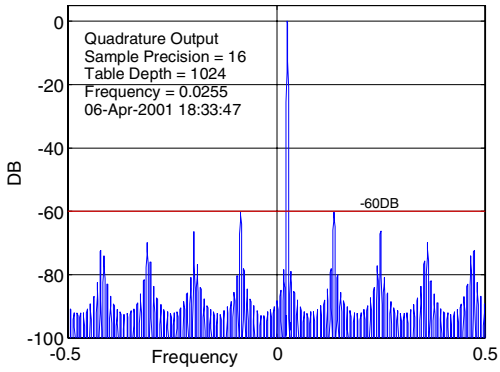

Figure 6:

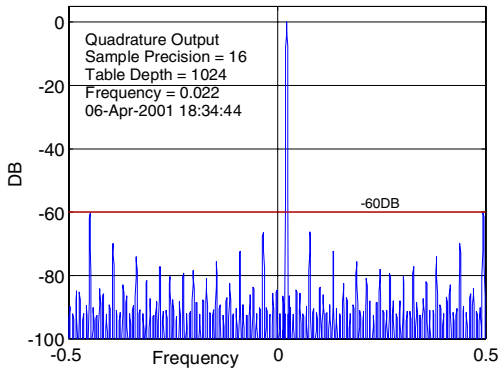

Figure 7:

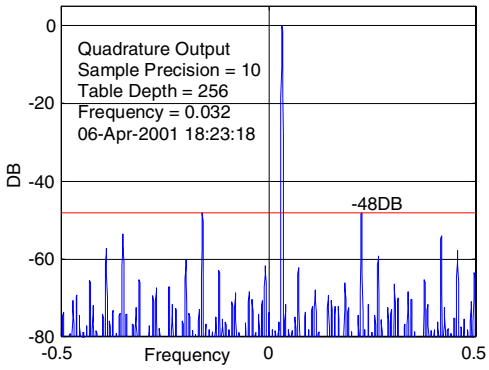

Figure 10:

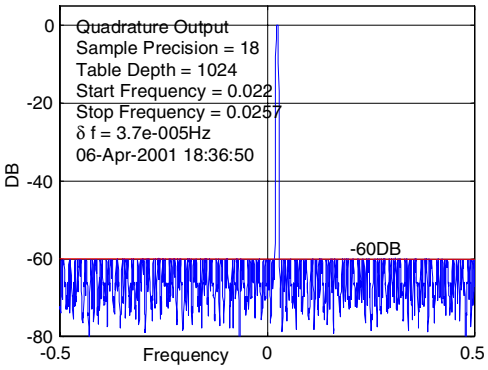

Figure 8:

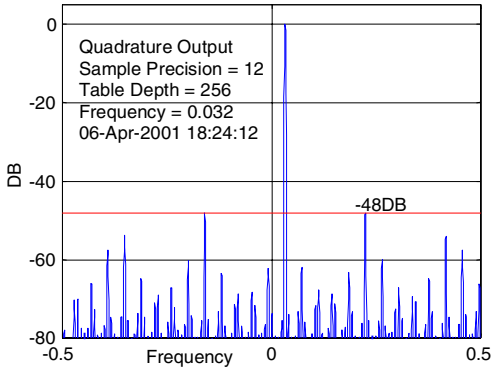

Figure 11:

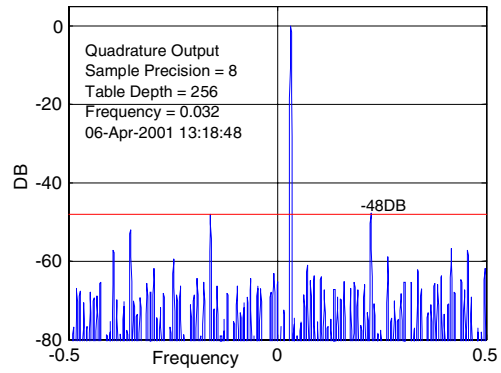

Figure 9:

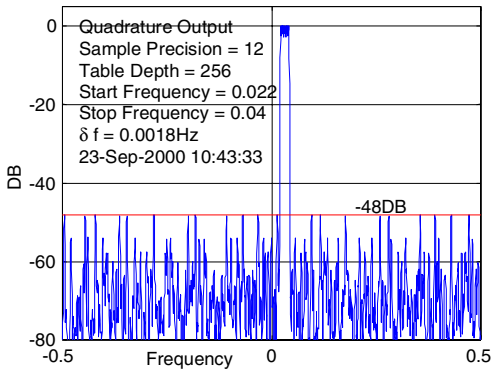

Figure 12:

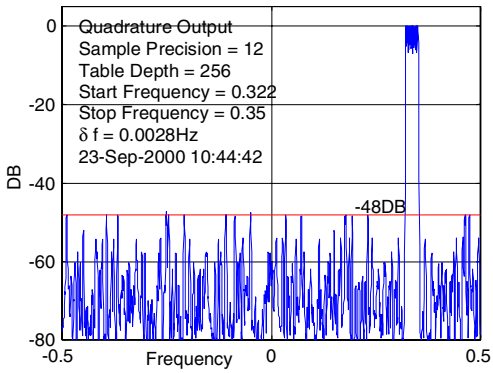

Figure 13:

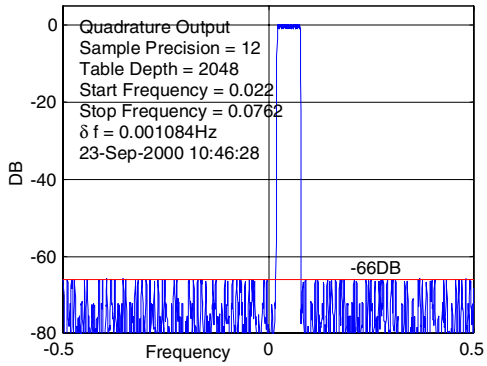

Figure 15:

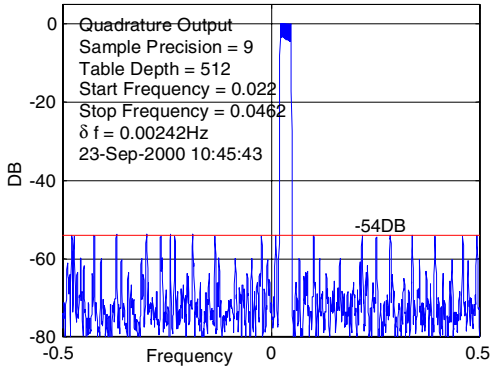

Figure 14:

Figure 1 provides a simplified view of the DDS core. A detailed view is shown in Figure 16. This detailed figure is similar to the simplified view, but also indicates the DDS control and interface *signals CE, A, WE, DATA, RFD* and *RDY*. Also note the inclusion of the *PHASE OFFSET* register designated *POFF*. This register is used for applying a constant phase offset to the phase slope computed in the phase accumulator *PACC*. When the Core is customized, the phase offset source can be defined as either a register, a constant, or it can be omitted entirely. When the *register* option is selected, the phase offset value is supplied via the *DATA* port. The phase offset value is treated as an unsigned quantity. If necessary, the phase offset is zero-extended before it is added to the phase accumulator.

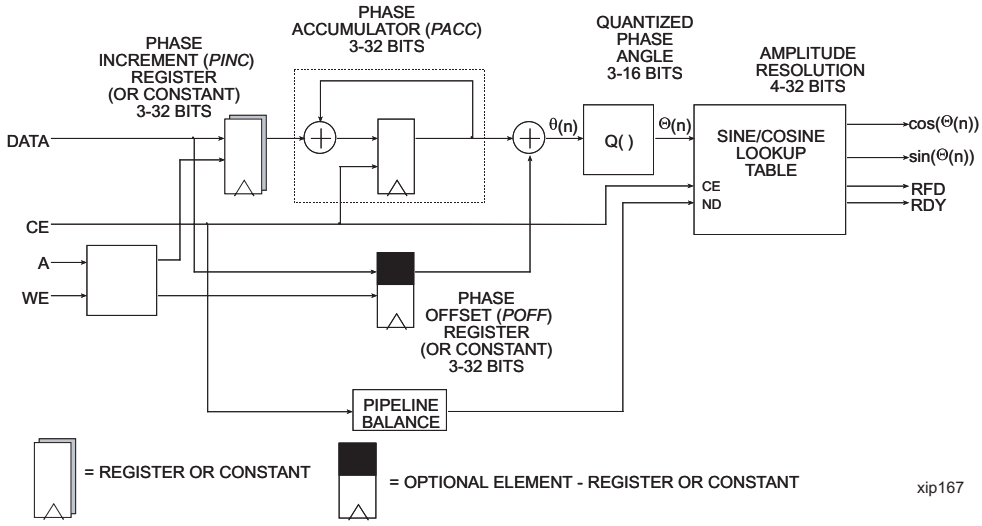

Figure 16: **DDS Core (Detailed View)**

The phase increment value can be sourced from either a register or a constant. When the registered option is selected, the *DATA* port supplies the phase increment value to the phase increment register. When the *constant* option is selected, the DDS output frequency is fixed and cannot be adjusted once the core is embedded in a design.

Phase Dithered DDS

In the phase truncation DDS architecture shown in Figure 1 the quantizer Q1 introduces a phase error in the phase slope by discarding the least significant part, actually fractional component, of the high-precision phase accumulator. The phase error due to the discarded fractional part of the address count is a periodic series which results in an undesired spectral line structure. Figure 17 provides an example of this process for a DDS with a table depth N=1024 and table sample precision of 16 bits. Figure 17(a) is the phase error generated by taking the difference between the quantizer input and output signals, Figure 17 (b) is the output time series and Figure 17(c) is the signal output spectrum. Observe in Figure 17(a) the periodic sawtooth structure of the phase error signal. The line spectrum associate with this correlated error sequence is impressed on the final output waveform and results in spectral lines in the synthesizer output spectrum. These spurious components can be clearly seen in Figure 17(c).

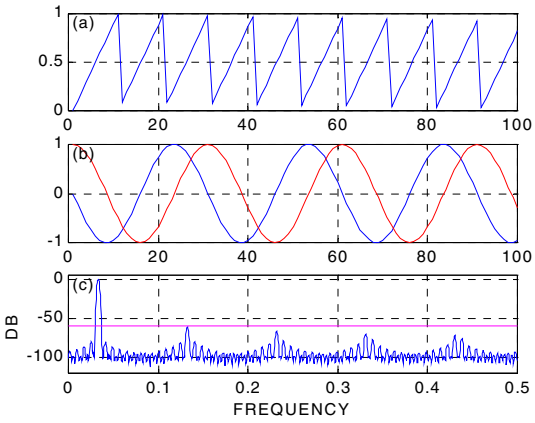

Figure 17: **DDS plots showing (a) phase error time series, (b) complex output time series, (c) output spectrum. 1024 deep lookup table, 16-bit samples, output frequency is 0.333 Hz.**

This structure can be suppressed by breaking up the regularity of the address error with an additive randomizing signal. This randomizing sequence, called *dither*, is a noise sequence, with variance approximately equal to the least significant integer bit of the phase accumulator. The dither sequence is added to the high-precision accumulator output prior to quantization by Q1. The resulting *dithered DDS* architecture is shown in Figure 18.

The dithered DDS supplies, approximately, an additional 12 dB of spurious free dynamic range (SFDR) in comparison to a phase truncation design. The additional logic resources required to implement the dither sequence generator are not significant.

To provide S dB of spur suppression using a phase truncation DDS, as referenced to the 0 dB primary tone, the internal lookup table must support at least $\lceil S/6 \rceil$ address bits. To achieve this same performance using the dithered architecture requires two fewer address bits, minimizing the number of block RAMs (or logic slices for a distributed memory implementation) used in the FPGA implementation. In summary, for a dithered DDS implementation, the number of address bits needed to support S dB spur suppression is equal to $\lceil S/6 \rceil - 2$.

Figure 19, Figure 20, and Figure 21 provide the results for several dithered DDS simulations. Figure 19 shows eight simulations for a complex dithered DDS employing a table depth $N = 4096$ and 16-bit precision samples. For each plot the output frequency is different and is annotated on the plot. A phase truncation design would typically generate output spurs 72 dB below the output frequency, independent of the actual value of the output frequency. Indicated on each of the plots by the parameter *A* is the peak spur level achieved for the simulation. The eight spurs are -88.12, -88.22, -86.09, -88.80, -87.21, -87.55, -87.83, -87.12 dB below the output frequency. The worst case value of -86.09 is 14.09 dB better than a similarly configured phase truncation DDS.

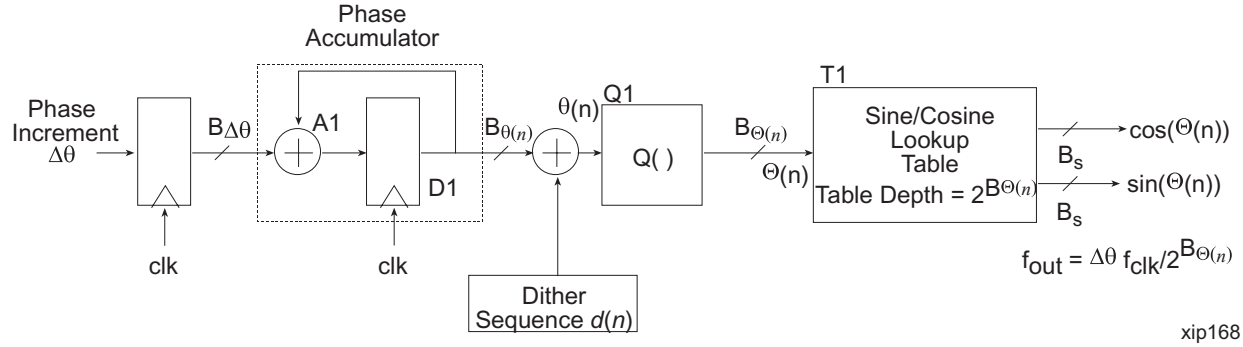

Figure 18: **Phase Dithered DDS Architecture**

To achieve this same SFDR by extending the table length of a phase truncation design would require extending the table by more than a factor of four.

Figure 20 and Figure 21 provide two more dithered DDS simulations where the output frequency is swept over a band of frequencies. The spectrum for each discrete tone in the sweep band is overlaid to construct the final plot. The sweep start frequency, end frequency, number of tones in the sweep, and DDS configuration are annotated on the plot.

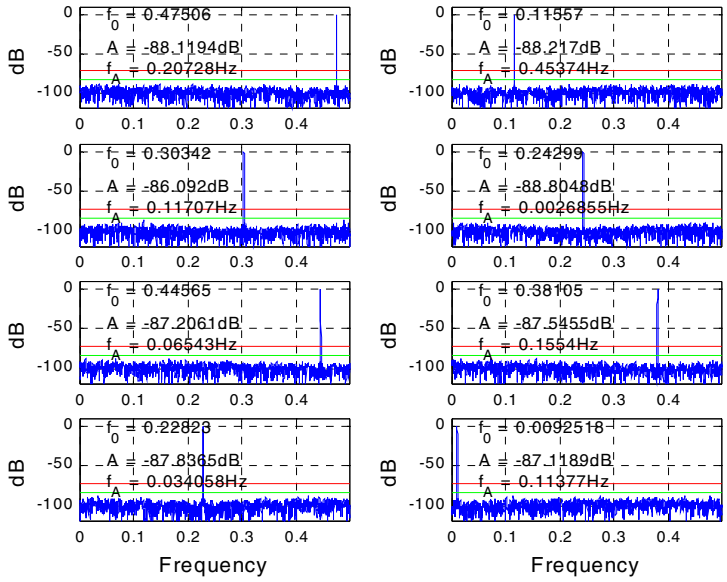

Figure 19: **Dithered DDS Simulations. The DDS configuration is $N = 4096$, $B_s = 16$. The eight plots are spectral domain representations for eight different output frequencies. Each plot is annotated with the peak spur level.**

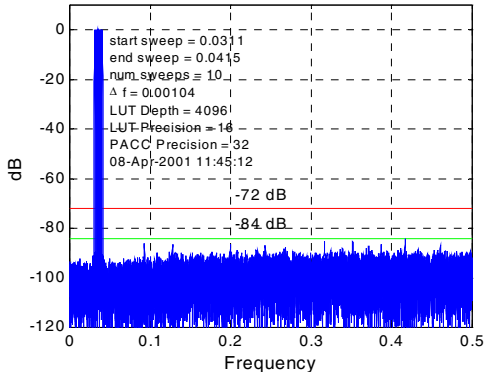

Figure 20:

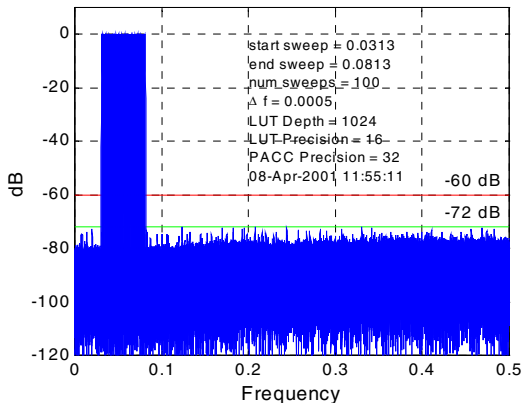

Figure 21:

In Figure 20 the synthesized signal is swept over a range of frequencies starting from 0.0311 to 0.0415 Hz. There are ten tones in the sweep separated in frequency by 0.00104 Hz. In this example the phase truncation DDS would produce peak spurs at –72 dB with respect to the 0 dB primary signal. The dithered DDS provides approximately 12 dB better performance with the peak spur –84 dB below the output signal.

Figure 21 shows the results of a second swept frequency dithered DDS simulation. In this case the start frequency is 0.0313 Hz, the sweep termination frequency is 0.0813 Hz and there are 100 tones in the sweep. The frequency differential between successive simulations is 0.0005 Hz. A sim-

ilarly configured phase truncation DDS, with a table depth of 1024 16-bit samples, would produce spurs approximately 60 dB below the output signal. Observe from the plot that the dithered DDS generates spurs that are 72 dB below the 0 dB output signal.

A further advantage of the dithered DDS is that the spectral line structure present in a phase truncation design is removed and the out-of-band signal is significantly whitened. This white broadband noise floor is more desirable than the line structured spectrum. In digital communication receivers that use a DDS for generating mixing signals for performing channelization functions, the spurs in a phase truncation DDS can act as low-level mixing tones and cause undesirable spectral contamination of the desired channel. For virtually all applications the preferred implementation is the dithered DDS.

Taylor Series Corrected DDS

The phase dithered DDS, as well as the phase truncation DDS, have a quantizer Q1 which produces a lower precision $\Theta(n)$, by discarding the fractional component of the high precision $\theta(n)$. The reason for this quantization step is to keep the size of the lookup memory to a reasonable size. The trade-off is spectral purity. With the availability of embedded multipliers in the Virtex-II, Virtex-II Pro, Spartan-IIE, and Spartan-3 FPGAs, it is now practical, from a silicon-area standpoint, to use the previously discarded fractional bits to calculate corrections that can be added to the lookup table values and produce outputs with very high spurious free dynamic range (SFDR). These embedded multipliers are ASIC like multipliers and do not consume any of the logic fabric. The Taylor series correction DDS fixes a number of the parameters as described in the Parameters section under Noise Shaping. The additional resources required over the phase truncation DDS are two embedded multipliers, one constant coefficient multiplier, and four adders.

Figure 22, Figure 23, Figure 24, and Figure 25 show the results of four different Taylor series corrected DDS simulations. The Taylor series corrected architecture uses a table depth $N = 4096$ and 18-bit precision samples. However, the precision at the output of the feed-forward error processor is 20 bits. For each plot the output frequency is different and annotated directly on the plot. A similarly configured phase truncation DDS would produce spurs at –72 dB and a phase dithered DDS at –84 dB. The peak spurs for the four plots are –118.25, –118.13, –118.10, and –118.17 dB below the output frequency.

Figure 27 shows a swept frequency Taylor series corrected DDS. The starting frequency for this example is 0.0313 Hz, the final frequency is 0.0813 and there are 100 tones in the sweep. Using this configuration, a phase truncation DDS would produce peak spurs at approximately 72 dB below the output signal and a phase dithered DDS would produce peak spurs at approximately 84 dB below the output signal. As shown in the plot the Taylor series corrected DDS produced spurs that were all the way down to 118 dB below the output signal. This result is 34 dB better than the phase dithering DDS, 46 dB better than the phase truncation DDS, and still only consumes a single 18Kb block RAM for the lookup storage. Figure 28 shows another frequency sweep simulation with 35 tones over a broader frequency range.

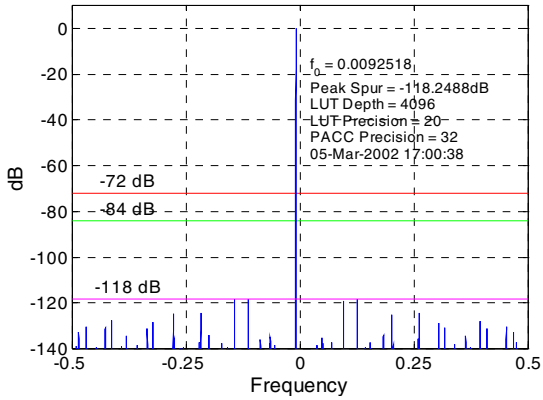

Figure 22: **Taylor series corrected DDS – single tone test. f_0 = 0.0092518.**

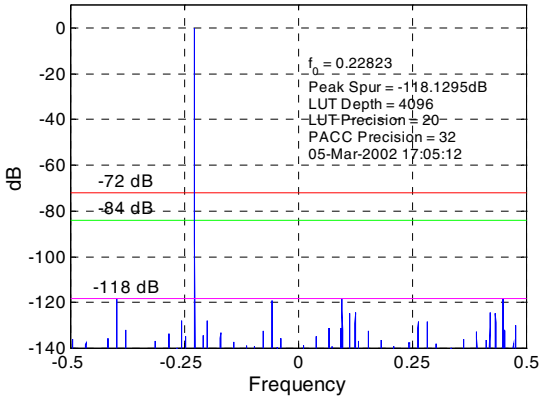

Figure 23: **Taylor series corrected DDS - single tone test. f_0 = 0.22823.**

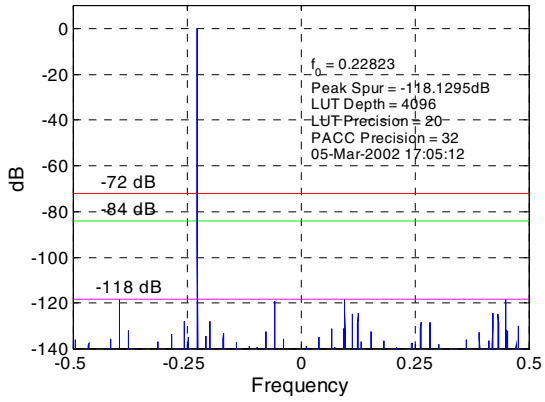

Figure 24: **Taylor series corrected DDS - single tone test. f_0 = 0.22823.**

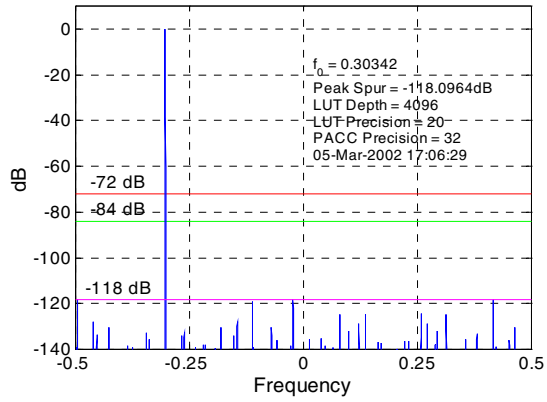

Figure 25: **Taylor series corrected DDS - single-tone test. f_0 = 0.30342.**

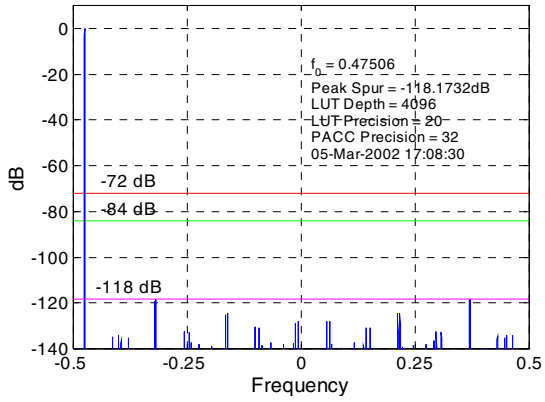

Figure 26: **Taylor series corrected DDS - single tone test. f_0 = 47506.**

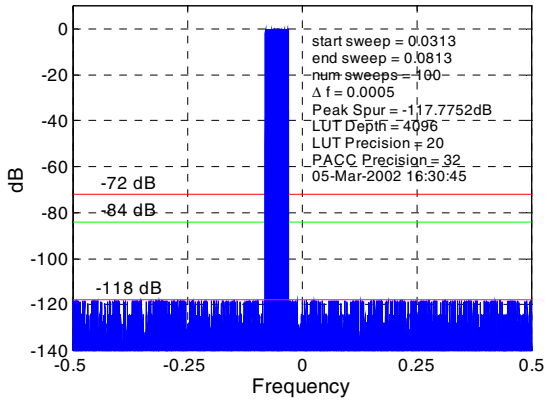

Figure 27: **Taylor series corrected DDS - frequency sweep simulation. 100 tones.**

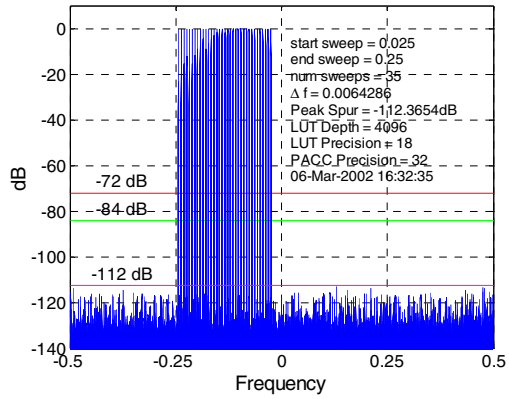

Figure 28: **Taylor series corrected DDS - frequency sweep simulation. 35 tones.**

Interface, Control, and Timing

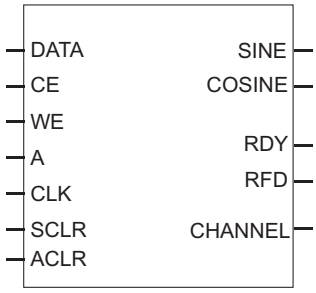

Figure 29: **DDS Symbol**

Table 1: **Core Signal Pinout**

Signal Name	Direction	Description
CLK	Input	Master Clock - active rising edge.
A†	Input	Address select for writing to the phase increment (*PINC*) memory and the phase offset (*POFF*) memory. When A_{MSB}=0, the *PINC* memory is selected. When A_{MSB}=1, the *POFF* memory is selected. The four lower order bits of A are used to address up to 16 channels for the currently selected memory. The memory map is shown in Table 2.
WE†	Input	Write enable - active high. Enables a write operation to the *PINC* or *POFF* memories.
CE†	Input	Clock enable - active high. *CE* must be high during normal Core operation, but it is not required to be active during a write access to the *PINC* or *POFF* memories.
DATA†	Input	Time shared data bus. The *DATA* port is used for supplying values to the *PINC* or *POFF* memories.
ACLR†	Input	Asynchronous clear - active high. When *ACLR* is asserted, the all registers in the Core are cleared. *RDY* is also deasserted.
SCLR†	Input	Synchronous clear - active high. When *SCLR* is asserted, all registers in the Core are cleared. *RDY* is also deasserted.
RDY†	Output	Output data ready - active high. Indicates when the output samples are available.
RFD†	Output	Ready for data - active high. *RFD* is a dataflow control signal present on many Xilinx LogiCOREs. In the context of the DDS, it is supplied only for consistency with other LogiCOREs. This optional port is always tied to VCC.
CHANNEL†	Output	Channel index. Indicates which channel is currently available at the output when the DDS is configured for multi-channel operation. This is an unsigned two's complement signal. It's width is determined by the number of channels.
SINE	Output	Sine time-series.
COSINE	Output	Cosine time-series.

† denotes optional pin

Figure 30 shows the timing sequence for a single-channel DDS core. In this example, the DDS has both a phase increment *(PINC)* and a phase offset *(POFF)* memory. The *PINC* memory is first written with the *PINC* value 0x0010. This is realized by supplying the value on the *DATA* port and addressing the *PINC* memory by defining the MSB of the *A* port as A_{MSB}=0, as shown by the memory map in Table 2. Since this example is a single-channel case, the remaining four bits of the *A* port should be set to zeros. The write is performed on the positive clock edge. *WE* must be active; *i.e., WE*=1, to perform this operation. Immediately after the *PINC* memory is loaded, the value 0x1000 is written to the *POFF* memory. This requires A_{MSB}=1 and *WE*=1.

Table 2: **Phase Increment and Phase Offset memory map**

Address	Location Description
00000 - 01111	Phase Increment (PINC) values for channel 0 through 15
10000 - 11111	Phase Offset (POFF) values for channel 0 through 15

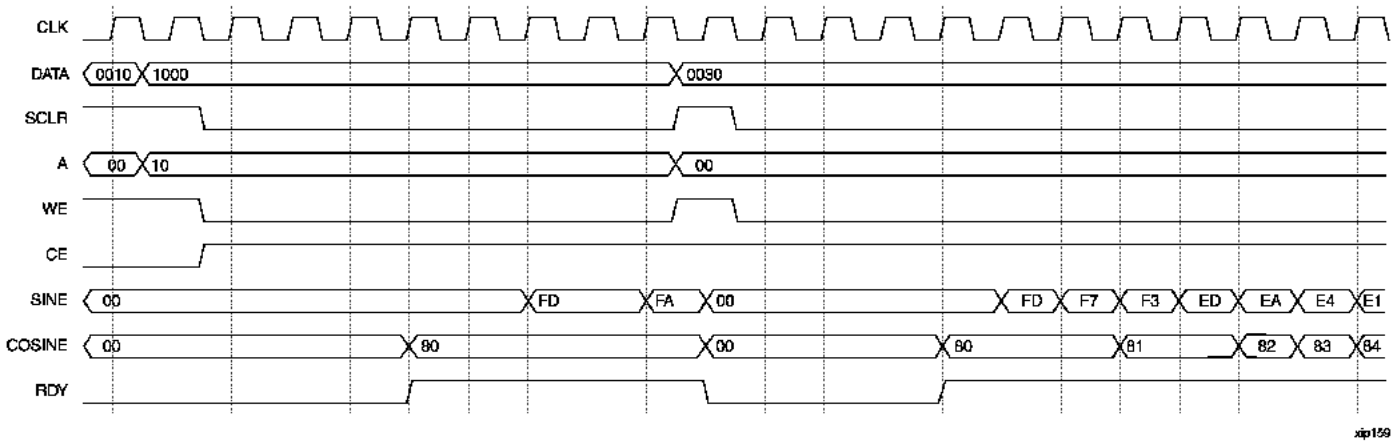

Figure 30: **DDS Timing: single-channel**

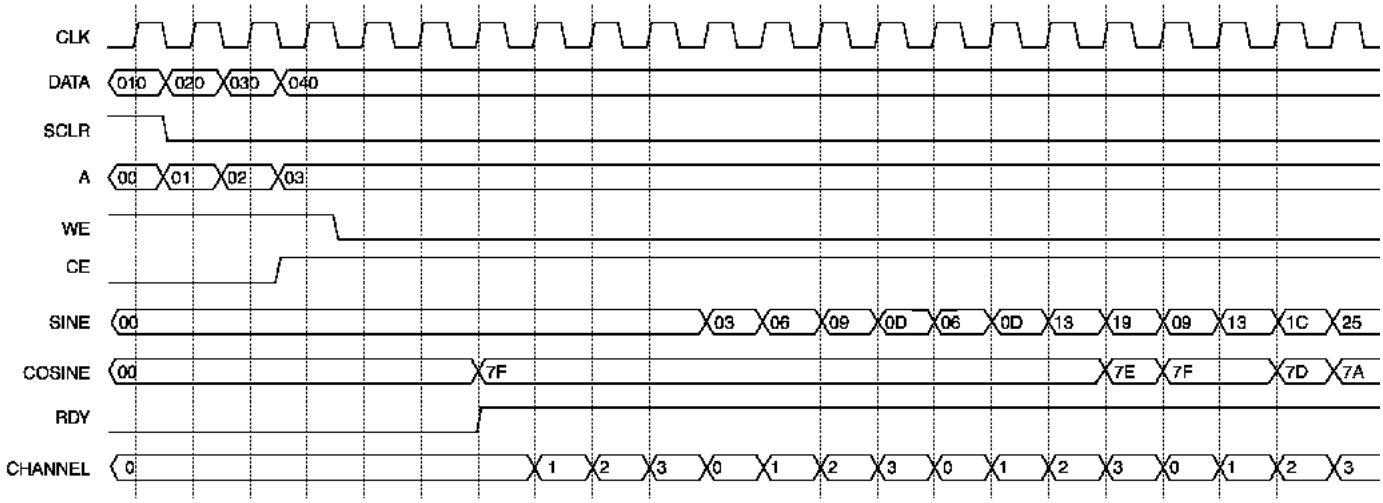

Figure 31: **DDS Timing: multichannel**

The memory is loaded on the rising edge of the clock. *CE* does not have to be active to write to either the *PINC* or *POFF* memories. The DDS will start operating once the clock enable is applied (*CE*=1). Since *CE* is an optional pin, DDS configurations that do not include this pin will begin operating once the FPGA is configured and the system clock is active. After a start-up latency (measured from the assertion of *CE*[2]) that depends on the pipelining configuration chosen for the core samples will be presented on the output port(s). This is indicated by *RDY*=1. For most configurations, the assertion by the Core of *RDY* indicates the first valid output sample. However, there is an exception. If a DDS is customized such that the *0-cycle* latency phase accumulator option is selected, and the sine-cosine look-up table is in distributed memory, and the table is purely combinatorial, any writes to the *PINC* register will be immediately reflected at the output port(s). This is irrespective of whether *CE* is asserted or not. In this situation, there are no registers between the *PINC* register and the output nodes, there is only a combinatorial arrangement of logic. Therefore, the *CE* pin cannot have any influence on this path through the system. The *CE* pin will, of course, still control the operation of the register (now in the upper arm of the phase accumulator shown in Figure 36 (b)) in the *PACC*. As illustrated in Figure 30, valid samples begin appearing at the output ports when *RDY* goes high.

The DDS can have an optional asynchronous clear or synchronous clear port. When either type of clear is applied, the *SINE* and *COSINE* output ports will assume a value of

2. Assuming this part is present.

0. Figure 30 shows the *SCLR* being asserted and a new *PINC* value being written on the same clock. This results in RDY going low until the pipeline fills, and then back high when valid samples are available again. The new *PINC* value is 0x0030, which represents an output frequency that is three times that of the previous *PINC* value. The new output samples can be seen to be changing at a faster rate, as expected.

Figure 31 shows the timing for a four-channel DDS implementation. The *PINC* value for each channel is written on the first four clock cycles. Valid samples are available on the outputs when the *RDY* signal goes high. The additional *CHANNEL* port indicates which channel is currently available at the output.

Parameters

The DDS parameterization screens are shown in Figure 32, Figure 33, Figure 34, Figure 36, and Figure 37.

The customization parameter definitions are:

- **Component Name:** The user-defined DDS component name.
- **Function:** The DDS may have a quadrature output (sine and cosine), or a single output port — either sine or cosine. In addition, the sign of the output signal(s) can be defined using the *Negative Sine* and *Negative Cosine* checkboxes. For example, if the quadrature output option is selected (*Sine and Cosine* GUI option), and both the *Negative Sine* and *Negative Cosine* boxes are unchecked, then the output signal *s(n)* is

$$s(n) = e^{j\Theta(n)} = \cos\Theta(n) + j\sin\Theta(n) \quad (13)$$

If the Negative Sine box has been checked, the output signal is defined by

$$s(n) = e^{-j\Theta(n)} = \cos\Theta(n) - j\sin\Theta(n) \quad (14)$$

  - Channels: The DDS core can generate a single-channel implementation as well as a multichannel implementation with support for up to 16 independent channels. If a multichannel implementation is generated, all channels will time-share the DDS outputs.

- DDS Performance Options: Both system-level and circuit-level performance requirements are specified and the DDS core generates an implementation to meet these requirements
  - **DDS Clock Rate:** The frequency at which the DDS core will be clocked
  - **Spurious Free Dynamic Range (SFDR):** This parameter defines the frequency domain requirements of the out-of-band noise generated by the DDS outputs. The range is from 16 to 115 dB of spur suppression. Note that an SFDR value of 102 dB or greater will force an implementation employing a Taylor Series Correction which requires the use of embedded multipliers.
  - **Frequency Resolution:** This parameter determines the granularity of the tuning frequency. If the value entered is 10, the tuning frequency can be adjusted to a precision of 10 Hz. As an example, you could tune the DDS to a frequency of 5.00003 MHz.

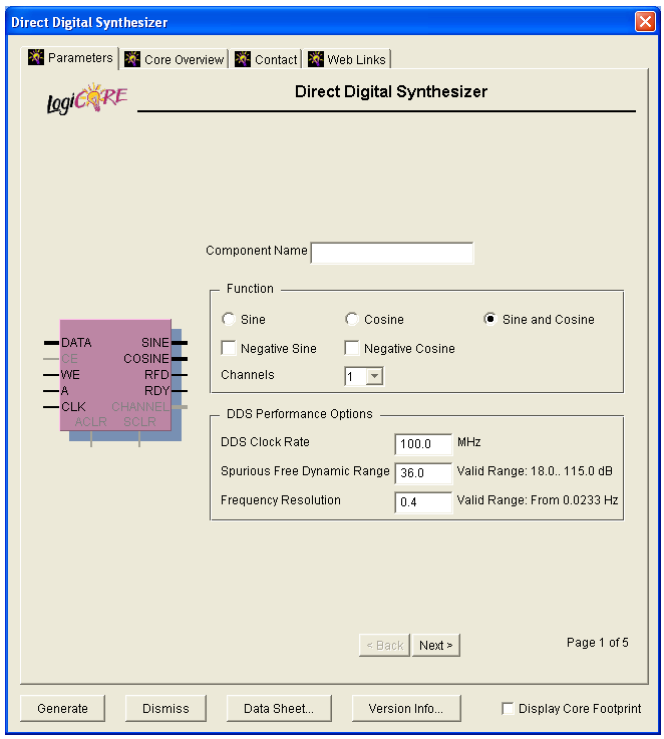

Figure 32: **DDS Parameterization Screen - Field 1**

- **Output Frequencies:** For each channel, an independent frequency can be entered into the table for the Sine and Cosine outputs. The allowable upper range is displayed above the table and is calculated by taking one half of the DDS clock rate and then dividing by the total number of channels. This upper limit is required so that the DDS clock rate does not drop below the Nyquist frequency.
  - **Phase Increment:** The user has the choice of making the output frequency a constant value or in-circuit changeable. Selecting *Fixed* will make the frequencies constant and selecting *Programmable* will allow the phase increment values to be changed in-circuit. If *Programmable* is selected, the values entered in the table will be the initial frequencies that take effect once the FPGA has been configured. If an ACLR or SCLR signal is asserted, the output frequencies settings will not be altered. Note that the *Fixed* or *Programmable* option is applied for all DDS channels.
- **Phase Offset Angles:** An independent offset can be added to the phase angle of each channel by entering a value into the table. The entered values will be multiplied by 2π radians. The valid range is -1.0 to 1.0.
  - **Phase Offset:** The user has the choice of making the phase offset angles a constant value, in-circuit changeable, or not used. Selecting *Fixed* will make the offsets constant; selecting *Programmable* will allow them to be changed in-circuit; selecting *None* will prevent any offset from being added. If *Programmable* is selected, the values entered in the table will be the initial offsets that will take effect once the FPGA has been configured. If an ACLR or SCLR signal is asserted, the phase offset angles settings will not be altered. Note that the *Fixed*, *Programmable*, or *None* option is applied for all DDS channels.
- **Clear Options:** If the *ACLR Pin* is selected, the core will be generated with a asynchronous reset. If the *SCLR Pin* is selected, the core will be generated with a reset that is synchronized to the clock. When asserted, the internal logic returns to its initialized state. Note that all programmable values are retained. The Sine and Cosine output ports will be driven to zeros until enough clock cycles have passed to fill the core's internal pipeline, which can be determined by the Latency value.
- **Clock Enable:** The Core can have an optional clock enable port.

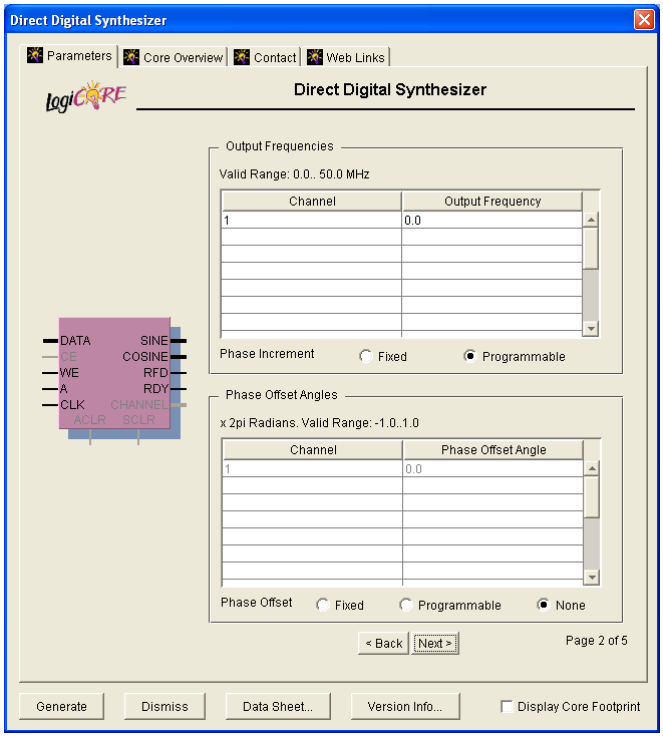

Figure 33: **DDS Parameterization Screen - Field 2**

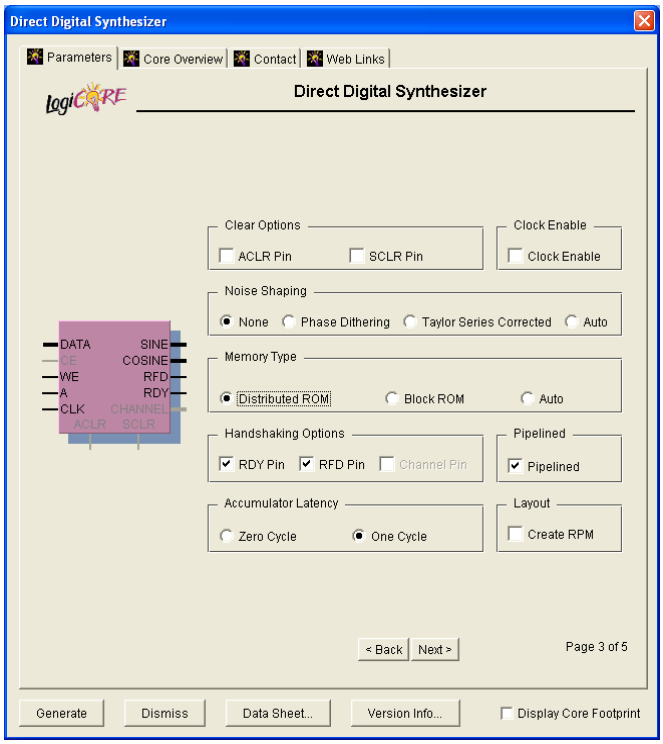

Figure 34: **DDS Parameterization Screen - Field 3**

- **Noise Shaping:** The radio buttons in this frame control whether a phase truncation, dithered DDS, or Taylor series corrected DDS implementation are generated. When *Auto* is selected, the noise-shaping type will be automatically determined, based on other core parameters including *SFDR*. When *None* is selected, a phase truncation DDS is produced. When *Phase Dithering* is selected, a dithered DDS implementation is generated. When *Taylor Series Corrected* is selected, a Taylor series corrected DDS implementation is generated. Taylor series corrected implementations are supported for only Virtex-II, Virtex-II Pro, Spartan-IIE, and Spartan-3 FPGAs. The *Taylor Series Corrected* radio button will be disabled when other architectures are selected. When *Taylor Series Corrected* is selected, the accumulator width will be fixed to 32 bits; the phase angle width will be fixed to 12 bits; the memory type will be fixed to block ROM; outputs required will be fixed to sine and cosine; negative sine will be fixed to true; negative cosine will be fixed to false; pipelining will be fixed to true, and the output width will be fixed to 20 bits.
- **Memory Type:** This field controls the location of the DDS trigonometric lookup table. When *Distributed ROM* is selected, the table is placed in distributed memory. If *Block ROM* is selected, the table will be implemented using block memory. If *Auto* is selected, the actual memory type will be automatically determined, based on other core parameters, including *SFDR*.
- **Handshaking Options:** Optional handshaking ports—*RDY* and *RFD*—can be included on the Core. The *RFD* output signal is simply tied to VCC and is an optional port that can be included for compatibility with other Xilinx LogiCORE products that employ this style of dataflow interface. As shown in Figure 30, the *RDY* output signal identifies when valid sine/cosine samples appear on these ports after the Core is started from rest — either after system power-on or a reset (synchronous or asynchronous). Any type of core reset will cause RDY to be removed (*RDY*=0). When the *Channels* parameter is set to a value of two or more and the *Channel Pin* parameter is selected, an additional output will be generated to indicate which channel the current output samples belong to.
- **Pipelined:** When *Pipelined* is selected, the core will be generated with pipelining registers inserted throughout the datapath. The insertion of pipeline registers enables the core to run at higher clock rates by shortening the delays between register stages. Pipelining increases the latency of the core, which is reported by the *Latency* value on the summary page of the GUI (Figure 35). If *Pipelined* is not selected, the latency of the core will be zero when a distributed memory implementation has been selected and the latency will be one when a block memory implementation has been selected.

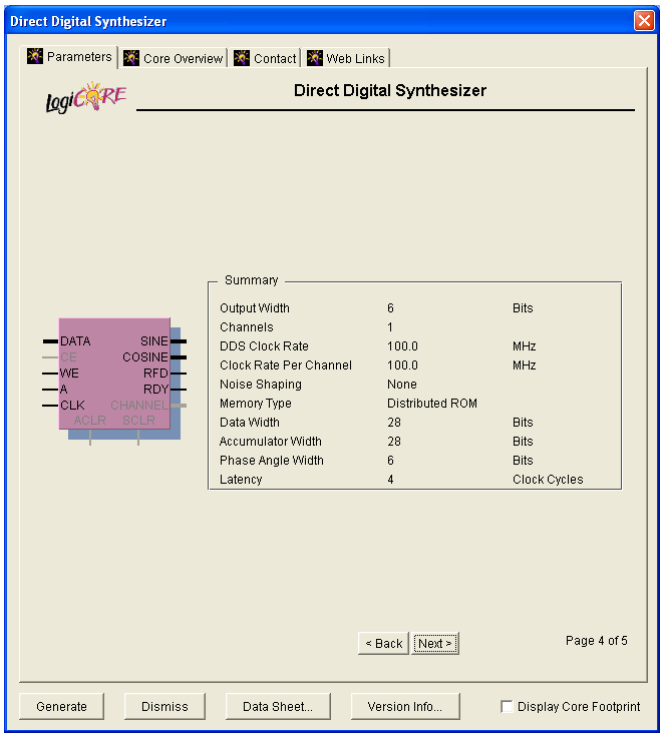

Figure 35: **DDS Parameterization Screen - Field 4**

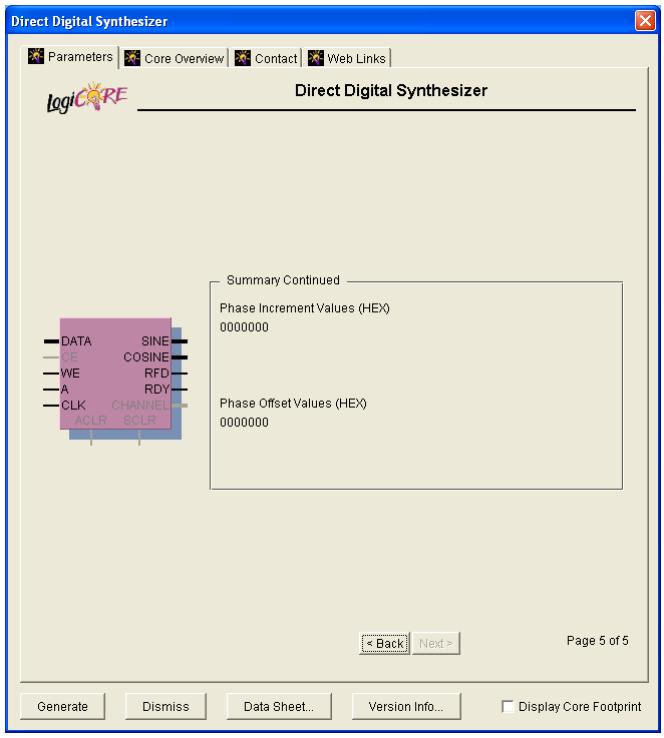

Figure 37: **DDS Parameterization Screen - Field 5**

- **Phase Accumulator:** Phase accumulator precision. This field defines the precision of the *PACC* register (Figure 16). The location of the register in the phase accumulator is controlled by the latency selection options. When the *one-cycle* latency option is selected, the phase accumulator will be as shown in Figure 36(a). When the *zero-cycle* option is selected, the arrangement in Figure 36(b) is employed.
- **Layout:** This checkbox controls whether a relationally placed MACRO (RPM) or a module with no placement information is generated. When checked, an RPM is produced.

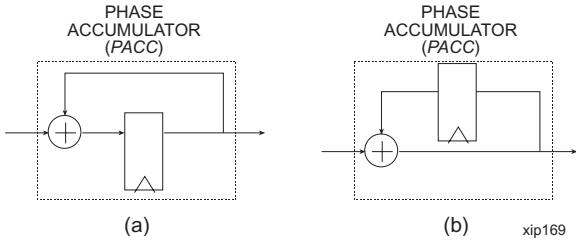

Figure 36: **Register Options for the Phase Accumulator--PACC**

Table 3: **Virtex Core Resource Utilization for Various Table Sizes (*Distributed ROM Implementation: 32-bit Data Width; 32-bit Phase Accumulator Width*)**

Phase Angle Width	Output Width	Single Output	Sine and Cosine
6*	8	32	49
	12	40	65
	16	48	81
	32	80	145
8	8	43	71
	12	53	91
	16	63	111
	32	103	191
10	8	95	175
	12	133	251
	16	171	327
	32	323	631

* Fullwave gets stored.

Note: Slice count is an approximation.

Table 4: **Virtex-II Core Resource Utilization for Various Table Sizes (*Distributed ROM Implementation: 32-bit Data Width; 32-bit Phase Accumulator Width)***

Phase Angle Width	Output Width	Single Output	Sine and Cosine
6*	8	32	49
	12	40	65
	16	48	81
	32	80	145
8*	8	80	145
	12	112	209
	16	144	273
	32	272	529
10	8	89	162
	12	123	230
	16	157	296
	32	292	568

* Fullwave gets stored.
Note: Slice count is an approximation.

Table 5: **Virtex Core Resource Utilization for Various Table Sizes (*Block ROM Implementation*: *32-bit Data Width; 32-bit Phase Accumulator Width*)**

Theta Width	Output Width	Single Output		Sine and Cosine	
		No. of Slices	No. of Block ROMs	No. of Slices	No. of Block ROMs
6	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	16	1*	62	2
8	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	42	1	68	2
10	8	33	1	50	1
	12	35	1	54	1
	16	37	1	58	1
	32	45	2	74	2
13	8	37	4	59	4
	12	39	6	63	6
	16	41	8	67	8
	32	48	16	81	16

Table 5: **Virtex Core Resource Utilization for Various Table Sizes (*Block ROM Implementation*: *32-bit Data Width; 32-bit Phase Accumulator Width*) *(Continued)***

Theta Width	Output Width	Single Output		Sine and Cosine	
		No. of Slices	No. of Block ROMs	No. of Slices	No. of Block ROMs
16	8	69	28	120	28
	12	87	44	156	44
	16	105	60	192	60
	32	239	124	461	124

Table 6: **Virtex-II Core Resource Utilization for Various Table Sizes (*Block ROM Implementation*: *32-bit Data Width; 32-bit Phase Accumulator Width)***

Phase Angle Width	Output Width	Single Output		Sine and Cosine	
		No. of Slices	No. of Block ROMs	No. of Slices	No. of Block ROMs
6	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	16	1*	17	1*
8	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	16	1*	17	1*
10	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	45	1	74	1
13	8	36	1	57	1
	12	38	2	61	2
	16	40	2	65	2
	32	49	4	83	4
16	8	39	7	62	7
	12	63	10	110	10
	16	65	14	114	14
	32	122	28	226	28

* Fullwave gets stored. *Note: Slice count is an approximation.*

The additional logic required for the phase dithered DDS is 55 slices. The total slice count can be calculated by adding 55 to any of the slice count data shown in Table 3, Table 4, Table 5, and Table 6.

The utilization for the Taylor series corrected DDS is as follows: 315 slices; one 18Kb block RAM, and two 18x18s embedded multipliers.

XCO File Parameters

The parameters supplied via the filter GUI are captured and logged to the*.xco* file. The full name of this file is simply the Component Name with an .xco file extension. Table 7 defines the .xco file parameter names and range specifications. Figure 38 is an example .xco file.

Table 7: **XCO File Parameter Names, Definitions, and Range Specifications**

Parameter Name	Definition	Range
BusFormat	Controls the notation employed for identifying buses in the output edif netlist file	{BusFormatAngleBracket | BusFormatParen}
SimulationOutputProducts	Core HDL simulation selection — either VHDL or Verilog	{VHDL | VERILOG}
ViewlogicLibraryAlias	Pathname to Viewlogic directory	Valid path name for the user's operating system
XilinxFamily	The FPGA target device family	{Virtex | Virtexe | Virtex2 | Virtex2p | Spartan2 | Spartan2e | Spartan3}
DesignFlow	HDL flow specifier	{VHDL | VERILOG}
FlowVendor	Design flow vendor information	{Other | Synplicity | Exemplar | Synopsis | Foundation}
channels	The number of independent channels the generated core will support	[1,…,16]
create_rpm	When *create_rpm=true* a Core with embedded physical placement information is generated. If *create_rpm=false* the Core is generated without placement data.	{true | false}
pipelined	This parameter controls the degree of pipelining employed in the sine/cosine lookup table. When set to *true* the table is fully pipelined. When set to *false* the minimum (zero for the case of a distributed ROM) number of pipeline registers are used.	{true | false}
noise_shaping	When *noise_shaping=Taylor_Series_Corrected,* a Taylor series corrected DDS is generated. When *noise_shaping=Phase_Dithering,* a dithered DDS is generated. When *noise_shaping=None*, a phase truncation DDS is generated. When *noise_shaping=Auto,* the noise shaping type will be determined by the core.	{Taylor_Series_Corrected | {Phase_Dithering | None | Auto}
clock_enable	Clock enable. When *ce_enable=true* a clock enable port is included on the component. When *clock_enable=false* the clock enable port suppressed.	{true | false}
spurious_free_dynamic_range	A quantity that describes the spectral purity of the output waveforms	[18,…,115]

Table 7: **XCO File Parameter Names, Definitions, and Range Specifications *(Continued)***

Parameter Name	Definition	Range
accumulator_latency	The user may control the position of the register in the phase accumulator. When *accumulator_register=ONE_CYCLE* the register location shown in Figure 38(a) is employed. When *accumulator_register=ZERO_CYCLE,* the register location shown in Figure 38(b) is employed.	{ONE_CYCLE | ZERO_CYCLE}
sclr_pin	Synchronous clear. When *sclr_pin=true,* a synchronous reset port is included on the component. When *sclr_pin=false,* the synchronous re reset port suppressed.	{true | false}
rdy_pin	When *rdy_pin=true,* the DDS Core includes a *RDY* output signal. If *rdy_pin=false*, the *RDY* port is suppressed.	{true | false}
memory_type	The sine/cosine samples can be stored in distributed or block memory. When *memory_type=Block_ROM,* the samples are stored in Block memory. When *memory_type=Distributed_ROM,* distributed memory is used. When *memory_type=Auto,* the memory type will be determined by the core.	{Block_ROM | Distributed_ROM | Auto}
phase_increment	The phase increment value, that is the delta phase increment supplied to the phase accumulator, may be sourced (by the *PACC*) from either a memory (*phase_increment=Programmable*) or a constant (*phase_increment=Fixed*).	{Programmable | Fixed}
channel_pin	If *channels* is greater than one, a *CHANNEL* port can optionally be added	{true | false}
dds_clock_rate	Frequency of the DDS clock signal	[0,...,300]
aclr_pin	Asynchronous clear. When *aclr_pin=true,* an asynchronous reset port is included on the component. When *aclr_pin=false,* the asynchronous reset port suppressed.	{true | false}
component_name	Textbox that defines the DDS component name	Any valid file name for the user's operating system consisting of the letters a…z, 0…9 and '_'. The component name may be a maximum of 32 characters.
negative_sine	Applicable only if a sine port has been specified for the particular core instance. When *negative_sine=false*, the signal presented at the *sine* port is sin $\theta(n)$. When negative_sine=true, the signal presented at the sine port is - sin $\theta(n)$.	{true | false}

Table 7: **XCO File Parameter Names, Definitions, and Range Specifications *(Continued)***

Parameter Name	Definition	Range
outputs_required	The DDS may be customized to provide a sine-only output (*function=Sine*), cosine-only output (*function=Cosine*) or both sine and cosine (quadrature) outputs (*function=Sine_and_Cosine*).	{Sine | Cosine | Sine_and_Cosine}
output_frequencies	A comma-delimited set of frequency values. The number of values is determined by the *channels* value. Frequency values are decimals.	
negative_cosine	Applicable only if a cosine port has been specified for the particular core instance. When *negative_cosine=false*, the signal presented at the *cosine* port is cos $\theta(n)$. When *negative_cosine=true*, the signal presented at the cosine port is - cos $\theta(n)$.	{true | false}
rfd_pin	When *rfd_pin=true,* the DDS core includes a *RFD* output signal. If *rfd_pin=false,* the *RFD* port is suppressed.	{true | false}
phase_offset_angles	A comma-delimited set of phase offset angle factors. The number of values is determined by the *channels* value. Phase offset values are decimals from -1.0 to 1.0.	
frequency_resolution	A decimal value which determines the granularity of the *output_frequencies*	
phase_offset	As shown in Figure 16, an optional phase offset can be introduced at the output of the phase accumulator. When *phase_offset=Programmable,* this value is supplied by a memory. When *phase_offset=Fixed,* the phase offset is a constant that is supplied by the *phase_offset_angles* field when the core is elaborated. If *phase_offset=None,* no phase offset is included in the datapath.	{Programmable | Fixed | None}

```
# Xilinx CORE Generator 5.1.02i; Cores Update # 1
# Username = nyquist
# COREGenPath = c:\xilinx\coregen
# ProjectPath=C:\xilinx_projects\DSPGroup\ip_portfolio\dds\core
# ExpandedProjectPath = C:\xilinx_projects\DSPGroup\ip_portfolio\dds\core
# Core name: xco_example
SET BusFormat = BusFormatParen
SET SimulationOutputProducts = VHDL
SET ViewlogicLibraryAlias = ""
SET XilinxFamily = Virtex2
SET DesignFlow = VHDL
SET FlowVendor = Synplicity
SELECT Direct_Digital_Synthesizer Virtex2 Xilinx,_Inc. 4.2
CSET channels = 1
CSET create_rpm = true
CSET pipelined = true
CSET noise_shaping = Taylor_Series_Corrected
CSET clock_enable = true
CSET spurious_free_dynamic_range = 104
CSET accumulator_latency = One_Cycle
CSET sclr_pin = false
CSET rdy_pin = true
CSET memory_type = Block_ROM
CSET phase_increment = Programmable
CSET channel_pin = false
CSET dds_clock_rate = 100.0
CSET aclr_pin = false
CSET component_name = xco_example
CSET negative_sine = true
CSET outputs_required = Sine_and_Cosine
CSET output_frequencies = 33.333
CSET negative_cosine = false
CSET rfd_pin = false
CSET phase_offset_angles = 0.0
CSET frequency_resolution = 0.4
CSET phase_offset = None
GENERATE
```

Figure 38: **Example DDS XCO File**

DDS Design Example 1

Consider a DDS that is to satisfy the following requirements

$$\text{frequency resolution } \Delta f = 0.25 \text{ Hz}$$
$$\text{minimum spur suppression } S = 96 \text{ dB}$$
$$f_{clk} = 120 \text{ MHz}$$
$$\text{Quarature output: } s(n) = e^{-j(\Theta(n)+\phi)} \cos(\Theta(n)+\phi) - \sin(\Theta(n)+\phi))$$
$$\text{where the phase offset } \phi = +\pi/4 \text{ radians}$$

The DDS GUI accepts system-level parameters instead of low-level parameters such as the width of the phase accumulator, width of the phase angle, etc. Because of this all of the requirements above can be entered into the GUI directly without having to calculate low-level core details. To illustrate this example 1 will show some calculations that the core does automatically.

The field width of the phase accumulator (PACC in Figure 29) defines the DDS frequency resolution according to

$$\Delta f = \frac{f_{clk}}{2^{B_{\theta(n)}}} \quad (15)$$

Therefore, the bit-precision for the PACC register is

$$\begin{aligned} B_{\theta(n)} &= \log_2\left\lceil \frac{f_{clk}}{\Delta f} \right\rceil \\ &= \left\lceil \log_2 \frac{120 \times 10^6}{0.25} \right\rceil \\ &= \lceil 28.838459 \rceil \\ &= 29 \text{ bits} \end{aligned} \quad (16)$$

The spur suppression requirement is the dominant consideration when selecting the depth of the sine/cosine lookup table. Of course, the table samples must be specified with an appropriate precision to support this value. As noted in an earlier section of this document, each sine/cosine table address bit contributes approximately 6 dB of spur suppression. To meet the -96 dB spur level, the number of address bits required is

$$\begin{aligned} B_{\Theta(n)} &= \left\lceil \frac{96}{6} \right\rceil \\ &= \lceil 16 \rceil \\ &= 16 \text{ bits} \end{aligned} \quad (17)$$

The table depth *N* is

$$N = 2^{B_{\theta(n)}} = 2^{16} = 65536 \text{ samples} \quad (18)$$

The phase offset value *POFF* is determined as

$$\begin{aligned} \text{POFF} &= 2^{B_\theta} \cdot \frac{\phi}{2\pi} \\ &= \left\lceil 2^{29} \cdot \frac{\pi/4}{2\pi} \right\rceil \\ &= 67108864_{10} \end{aligned} \quad (19)$$

To generate a 10.2 MHz output signal the phase increment would be

$$\begin{aligned} \Delta\theta &= \frac{f_{\text{out}} 2^{B_{\theta(n)}}}{f_{\text{clk}}} \\ &= \frac{10.2\times10^6 \times 2^{29}}{120\times10^6} \\ &= \lceil 45634027.52 \rceil \\ &= 45634028 \end{aligned} \quad (20)$$

If the *PINC* is configured to be programmable then the output frequency can be adjusted while the DDS is running. In order for the DDS to be tuned to 8.2 MHz the new *PINC* can be calculated to be

$$\begin{aligned} \Delta\theta &= \frac{f_{\text{out}} 2^{B_{\theta(n)}}}{f_{\text{clk}}} \\ &= \frac{8.2\times10^6 \times 2^{29}}{120\times10^6} \\ &= \lceil 36686178.98 \rceil \\ &= 36686178 \end{aligned} \quad (21)$$

To write this new value to the *PINC* memory it must be presented on the *DATA* port along with an active *WE* and appropriate address on the *A* port. The *DATA* port will assume the same width as the *PACC* which was calculated to be 29 bits. The width of the DATA port is also reported on page 4 (Figure 35) of the GUI. The PINC value represented as a 29-bit binary value is

$$\Delta\theta = 00010001011111100100101100 0$$

Figure 39 is a spectral plot of the 10.2 MHz output signal. Observe that the highest spur meets the 96 dB spur suppression requirement.

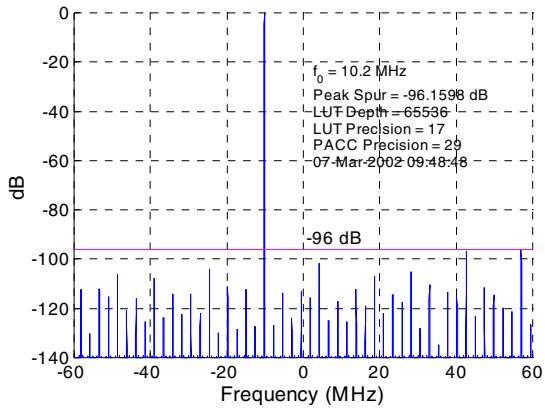

Figure 39: **Phase truncation DDS example 1. 10.2 MHz output tone. The highest spur is below the required 96 dB suppression value. The table depth is 65536 samples.**

In this example, the phase truncation DDS requires 16 of the 18Kb block RAMs for the lookup table storage. If a phase dithered DDS is used the table depth can be reduced to 16384 samples which reduces the memory requirement to 4 block RAMs. The output power spectrum for the preferred dithered DDS design is shown in Figure 40. Note that the minimum spur suppression has been achieved and the line structure of the spectrum shown in Figure 39 is no longer present.

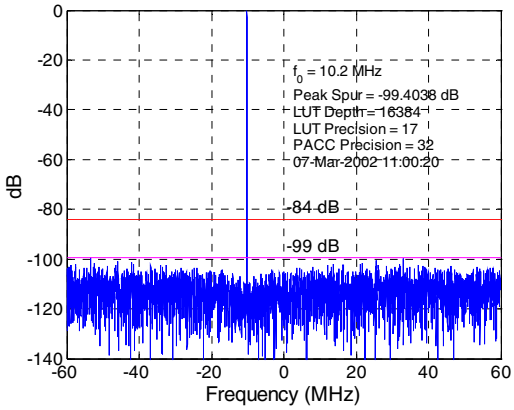

Figure 40: **Dithered DDS design for example 1. 10.2 MHz output tone.**

Using the dithered DDS, the highest spur, at -99.4038 dB, is well below the required 96 dB suppression value. If Virtex-II, Virtex-II Pro, Spartan-IIE, or Spartan-3 FPGAs are being used and two embedded multipliers are available the Taylor series corrected DDS can be used. While this architecture produces spurs at about -115 dB, which is much cleaner than the required -96 dB, the table depth is reduced to 4096 samples which fits in a single 18Kb block RAM. The output

power spectrum for the Taylor series corrected DDS is shown in Figure 41.

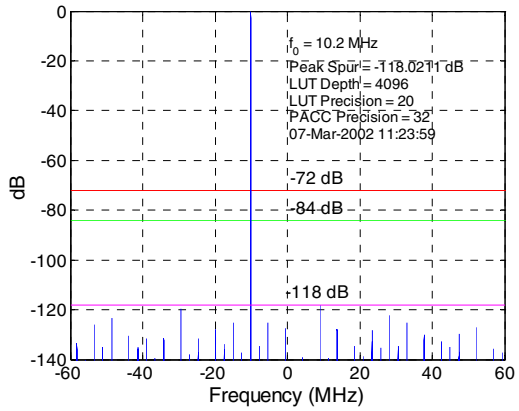

Figure 41: **Taylor series corrected DDS design for example 1, 1. 10.2Mhz output tone.**

Using the same requirements as before this example can be extended to use a multi-channel DDS. In doing so some of the calculations change slightly. Assuming a four channel DDS the frequency resolution is defined by

$$\Delta f = \frac{\frac{f_{clk}}{channels}}{2^{B_{\theta(n)}}} \quad (22)$$

Therefore, the bit-precision for the PACC register is

$$\begin{aligned} B_{\theta(n)} &= \log_2\left\lceil \frac{\frac{f_{clk}}{channels}}{\Delta f} \right\rceil \\ &= \left\lceil \log_2 \frac{\frac{120\times10^6}{4}}{0.25} \right\rceil \\ &= \lceil 26.83845 \rceil \\ &= 27 \text{ bits} \end{aligned} \quad (23)$$

Assuming that the four tones to be generated are 8.2 MHz, 10.2 MHz, 12.2 MHz, and 14.2 MHz, the phase increment would be

$$\begin{aligned} \Delta\theta &= \frac{f_{out}2^{B_{\theta(n)}}}{\frac{f_{clk}}{4}} \\ &= \frac{f_{out}\times 2^{12}}{\frac{120\times10^6}{4}} \end{aligned} \quad (24)$$

$$\Delta\theta_0 = 1120$$
$$\Delta\theta_1 = 1393$$
$$\Delta\theta_2 = 1666$$
$$\Delta\theta_3 = 1939$$

These four PINC values represented as 27-bit binary values are

$$\begin{aligned} \Delta\theta_0 &= 000000000000000010001100000_2 \\ \Delta\theta_1 &= 000000000000000010101110001_2 \\ \Delta\theta_2 &= 000000000000000011010000010_2 \\ \Delta\theta_3 &= 000000000000000011110010011_2 \end{aligned} \quad (25)$$

Figure 42 shows the power spectrum for the four-channel DDS implemented as a Taylor series corrected DDS.

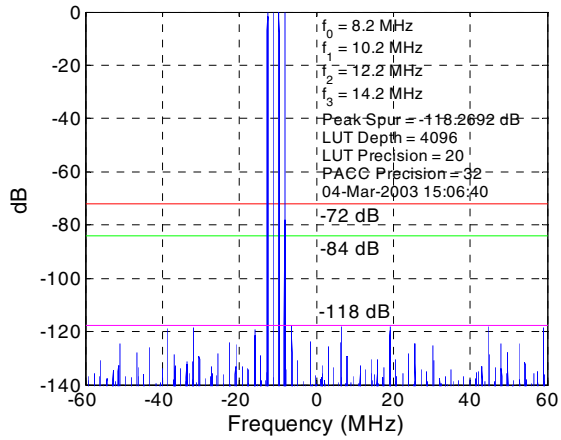

Figure 42: **Power spectrum for the four-channel DDS implemented as a Taylor series corrected DDS.**

DDS Design Example 2

Consider a DDS that is to satisfy the following requirements

$$\text{frequency resolution } \Delta f = 0.3 \text{ Hz}$$
$$\text{minimum spur suppression } S = 82 \text{ dB}$$
$$f_{clk} = 80 \text{ MHz}$$
$$\text{Quadrature output: } s(n)=e^{-j(\Theta(n)+\phi)} \cos(\Theta(n)+\phi)\text{-}\sin(\Theta(n)+\phi))$$
$$\text{where the phase offset } \phi\text{=-50 degrees}$$

The DDS GUI has been redesigned allowing the user to enter system-level parameters instead of low-level parameters such as the width of the phase accumulator. Because of this all of the requirements above can be entered into the GUI directly without having to calculate low-level core details. As an exercise this example will show some calculations that the core does automatically.

The field width of the phase accumulator (PACC in Figure 29), defines the DDS frequency resolution according to

$$\Delta f = \frac{f_{clk}}{2^{B_{\theta(n)}}} \quad (26)$$

Therefore, the bit-precision for the PACC register is

$$\begin{aligned} B_{(n)} &\geq \log_2\left\lceil \frac{f_{clk}}{f} \right\rceil \\ &\geq \left\lceil \log_2 \frac{80 \quad 10^6}{0.3} \right\rceil \\ &\geq \lceil 27.99046 \rceil \\ &= 28 \text{ bits} \end{aligned} \quad (27)$$

The spur suppression requirement is the dominant consideration when selecting the depth of the sine/cosine look-up table. Of course, the table samples must be specified with an appropriate precision to support this value. As noted in an earlier section of this document, each sine/cosine table address bit contributes approximately 6 dB of spur suppression. To meet the -82 dB spur level, the number of address bits required is

$$\begin{aligned} B_{\Theta(n)} &= \left\lceil \frac{82}{6} \right\rceil \\ &= \lceil 13.666 \rceil \\ &= 14 \text{ bits} \end{aligned} \quad (28)$$

The table depth *N* is

$$N = 2^{B_{\Theta(n)}} = 2^{14} = 16384 \text{ samples} \quad (29)$$

In this example, the required phase offset has been specified as a negative value. The phase offset value -50 degrees is the same as +310 degrees.

The phase offset value *POFF* is determined as

$$\begin{aligned} \text{POFF} &= N \cdot \frac{\phi}{360} \\ &= \left\lceil 16384 \cdot \frac{310}{360} \right\rceil \\ &= 14336_{10} \end{aligned} \quad (30)$$

To generate an 8.4 MHz output signal, the phase increment would be

$$\begin{aligned} \Delta\theta &= \frac{f_{out} 2^{B_{\theta(n)}}}{f_{clk}} \\ &= \frac{8.4\times10^6 \times 2^{14}}{80\times10^6} \\ &= \lceil 1720.32 \rceil \\ &= 1720 \end{aligned} \quad (31)$$

If the DDS is configured such that the output frequency is programmable then the value of 1720 must be presented on the *DATA* port for writing to the *PINC* memory. The *DATA* port will assume the same width as the *PACC* which was calculated to be 28 bits. The *PINC* value represented as a 28-bit binary value is $\Delta\theta = 0001101011100001010001111011_2$.

Figure 43 is a spectral plot of the 8.4 MHz output signal. Observe that the highest spur meets the 82 dB spur suppression requirement.

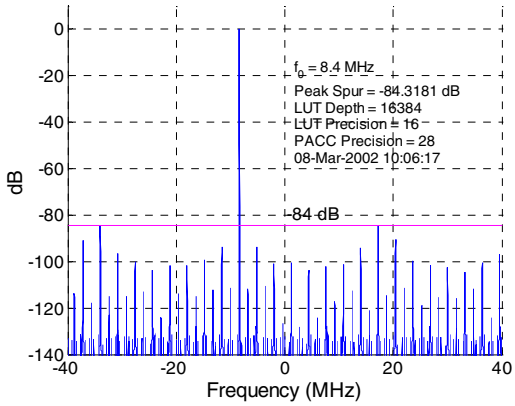

Figure 43: **Phase truncation DDS example 2. 8.4 MHz output tone. The highest spur is below the required 82 dB suppression value.**

The output power spectrum for the preferred dithered DDS design is shown Figure 44. In this case a 4096-deep table has been employed. Note that the minimum spur suppression has been achieved and the line structure of the spectrum shown in Figure 43 is no longer present. Since the desired spur suppression has been achieved while only consuming a single 18Kb block RAM the phase dithered DDS is the best area efficient solution, since moving to the Taylor series corrected architecture would require more resources.

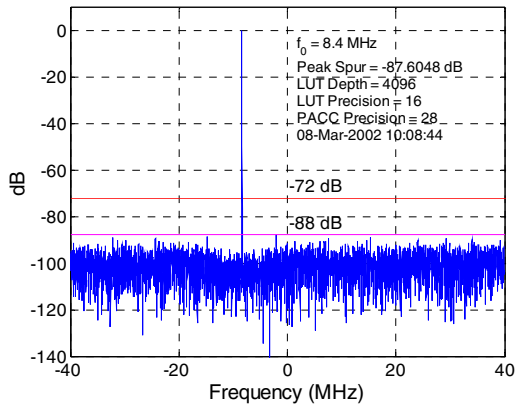

Figure 44: **Dithered DDS Design for Example 2. 8.4 MHz output tone. Using the dithered DDS the highest spur, at 88 dB, is below the required 82 dB suppression value. The design uses a table that is one quarter the depth of the phase truncation implementation.**

Ordering Information

This core may be downloaded from the Xilinx **IP Center** for use with the Xilinx CORE Generator system v7.1i and later. The CORE Generator system is bundled with the ISE Foundation software at no additional charge.

To order Xilinx software, please visit the Xilinx **Silicon Xpresso Cafe** or contact your local Xilinx **sales representative**.

Information on additional Xilinx LogiCORE modules is available on the Xilinx **IP Center**.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/15/03	1.0	Initial revision history.
05/21/04	2.0	Updated document to support Xilinx software v6.2i and Virtex-4 FPGA.
04/28/05	2.1	Updated support for Spartan-3E and Xilinx ISE software v7.1i.