Introduction

The LogiCORE™ IP DDS (Direct Digital Synthesizer) Compiler core sources sinusoidal waveforms for use in many applications. A DDS consists of a *Phase Generator* and a *SIN/COS Lookup Table*. These parts are available individually or combined via this core.

Features

- Drop-in module for Kintex™-7, Virtex®-7, Virtex-6, Virtex-5, Virtex-4, Spartan®-6, Spartan-3/XA, Spartan-3E/XA and Spartan-3A/AN/3A DSP/XA FPGAs
- Phase Generator and SIN/COS Lookup Table can be generated independently or combined together with optional Dither circuit to provide complete DDS solution
- High speed, including optimal and optional use of XtremeDSP™ slice
- Sine, Cosine, or quadrature outputs
- Lookup table can be allocated to distributed or block memory
- A phase dithering option spreads the spectral line energy structure associated with conventional phase truncation waveform synthesis architectures
- Phase dithering or Taylor series correction options provide high dynamic range signals using minimal FPGA resources. Supports Spurious Free Dynamic Range (SFDR) from 18 dBs to 150 dBs
- Support for 1 to 16 independent time-multiplexed channels
- Optional channel output indication for multi-channel use
- High-precision synthesizer with fine frequency resolution using up to 48-bit phase accumulator with XtremeDSP slice or fabric options
- 3-bit to 26-bit two's complement output sample precision
- Optional phase offset capability allows multiple synthesizers with precisely controlled phase differences
- Frequency and phase offset may be independently configured as constant, programmable or dynamic (for modulation)
- Choice of amplitude modes allows either maximal use of output dynamic range or unit circle amplitude
- Optional inversion of Sine or Cosine outputs
- GUI entry selectable in terms of System (SDFR and frequency resolution) or Hardware (phase and output width) parameters.
- For use with Xilinx CORE Generator™ software v13.1

LogiCORE IP Facts Table					
Core Specifics					
Supported Device Family[(1)]	Virtex-7 and Kintex-7, Virtex-6, Virtex-5, Virtex-4, Spartan-6, Spartan-3/XA, Spartan-3E/XA, Spartan-3A/3AN/3A DSP/XA				
Supported User Interfaces	Not Applicable				
	Resources[(2)]				**Frequency**
Configuration	**LUTs**	**FFs**	**DSP Slices**	**Block RAMs[(3)]**	**Max. Freq.[(4)]**
SFDR70, 12-bit phase, 12-bit sin/cos output, use DSP48	71	131	1	1	400 MHz
Provided with Core					
Documentation	Product Specification				
Design Files	Netlist				
Example Design	Not Provided				
Test Bench	Not Provided				
Constraints File	Not Applicable				
Simulation Model	VHDL behavioral model in the xilinxcorelib library VHDL UniSim structural model Verilog UniSim structural model				
Tested Design Tools					
Design Entry Tools	CORE Generator tool 13.1 System Generator for DSP 13.1				
Simulation	Mentor Graphics ModelSim 6.6d Cadence Incisive Enterprise Simulator (IES) 10.2 Synopsys VCS and VCS MX 2010.06 ISIM 13.1				
Synthesis Tools	N/A				
Support					
Provided by Xilinx, Inc.					

1. For a complete listing of supported devices, see the release notes for this core.
2. Resources listed here are for Virtex-6 devices. For more complete device performance numbers, see Table 6 - Table 13.
3. Based on 18K block RAMs (or 36K - select appropriate size).
4. Performance numbers listed are for Virtex-6 FPGAs. For more complete performance data, see Performance and Resource Utilization, page 25.

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 lookup table scheme. The lookup table stores samples of a sinusoid. A digital integrator is used to generate a suitable phase argument that is mapped by the lookup 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.

Theory of Operation

The simplest form of the DDS Compiler core uses phase truncation, as shown in Figure 1.

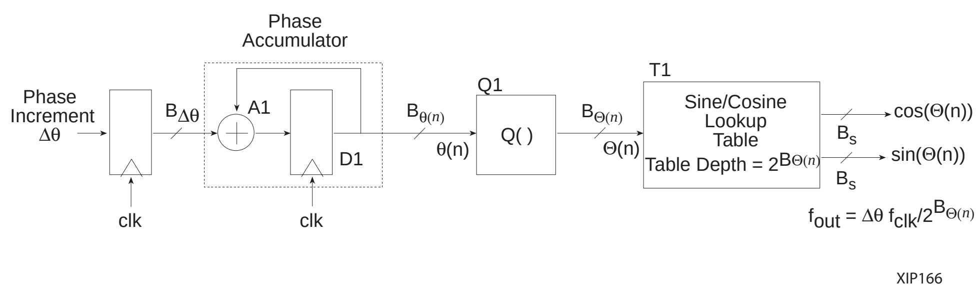


Figure 1: **Phase Truncation DDS (A Simplified View of the DDS Core)**

The integrator (components D1 and A1) computes a phase slope that is mapped to a sinusoid (possibly complex) by the lookup 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 lookup table that performs the mapping from phase-space to time.

The fidelity of a signal formed by recalling samples of a sinusoid from a lookup table is affected by both the phase and amplitude quantization of the process. The depth and width of the lookup table affect the signal's phase angle resolution and the signal's amplitude resolution, respectively. See Spectral Purity Considerations for more details.

Direct digital synthesizers use an addressing scheme with an appropriate lookup 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 lookup 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 follows:

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

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 resource efficient implementation because the memory requirements are minimized, offering either fewer FPGA block RAMs or reduced distributed memory. Based on the core customization parameters, the DDS core automatically employs quarter-wave or half-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 phase width, that is, number of bits, $B_{\theta(n)}$, in the phase accumulator and the phase increment value $\Delta\theta$. The output frequency in Hertz is defined by:

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

For example, if the DDS parameters are:

$$f_{clk} = 120MHz$$

$$B_{\theta(n)} = 10$$

$$\Delta\theta = 12_{10}$$

then the output frequency is calculated as follows:

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

$$= \frac{120 \times 10^6 \times 12}{2^{10}}$$

$$= 1.406250\ MHz$$

The phase increment value $\Delta\theta$ required to generate an output frequency f_{out} Hz is:

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

If we time-division multiplex the DDS core to do multiple channels, then we reduce the effective clock frequency per channel. For C channels, the phase increment required is:

1. For very shallow 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 software always produces the smallest core possible.

$$\Delta\theta = \frac{Cf_{out}2^{B_{\theta(n)}}}{f_{clk}}$$

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:

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

For example, for the following DDS parameters:

$$f_{clk} = 120\text{ MHz}$$

$$B_{\theta(n)} = 32$$

the frequency resolution is:

$$\Delta f = \frac{f_{clk}}{2^{B_{\theta(n)}}} = \frac{120 \times 10^6}{2^{32}} = 0.0279396\text{ Hz}$$

In the time-division multi-channel case, the frequency resolution is improved by the number of channels, as follows:

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

Phase Increment

The phase increment is unsigned in the sense that if it needs to be extended it will be padded with 0s rather than sign-extended. However, when the phase increment value width matches the phase width, it can be considered to be unsigned or signed without impact since the range 0 to 2^N describes the range [0,360) degrees whereas the range $-2^{(N-1)}$ to $2^{(N-1)}-1$ describes the range [-180,180) degrees (where N is the number of bits in the phase accumulator). The phase increment term $\Delta\theta$ defines the synthesizer output frequency. Consider a DDS with the following parameterization:

$$f_{clk} = 100\text{ MHz}$$

$$B_{\theta(n)} = 18$$

$$B_{\Theta(n)} = 12$$

To generate a sinusoid with frequency $f_{out} = 19\text{ MHz}$, the required phase increment is:

$$\Delta\theta = \frac{f_{out}B_{\theta(n)}}{f_{clk}} = \frac{19 \times 10^6 \times 2^{18}}{100 \times 10^6} = 49807.36$$

This value must be truncated to an integer giving the following actual frequency:

$$f_{out} = \frac{\Delta\theta f_{clk}}{B_{\theta(n)}} = \frac{49807 \times 100 \times 10^6}{2^{18}} = 18.9998627\,MHz$$

Spectral Purity Considerations

The fidelity of a signal formed by recalling samples of a sinusoid from a lookup table is affected by both the phase and amplitude quantization of the process. The depth and width of the lookup table affect the phase angle resolution and the amplitude resolution of the signal, respectively. These resolution limits are equivalent to time base jitter and 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 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 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 width of the accumulator is:

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

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 lookup 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. This jitter results in undesired phase modulation that is proportional to the quantization error, as shown by the following:

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

Figure 2 shows the lookup 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 lookup 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).

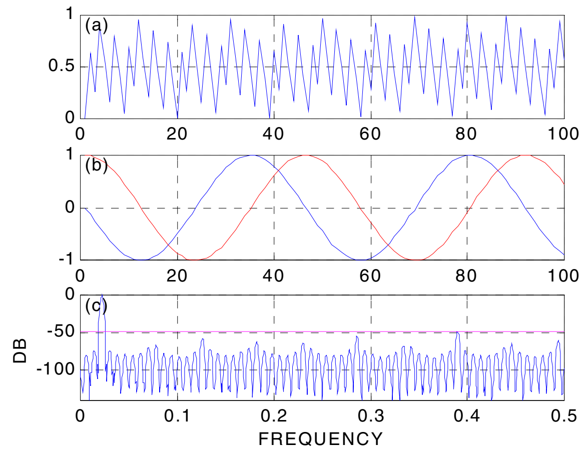


Figure 2: **Phase Truncation DDS. f_{out} = 0.022Hz, Table Depth = 256 12-Bit Precision Samples. (a) Phase Angle Addressing Error (b) Complex Output Time Series (c) Output Spectrum**

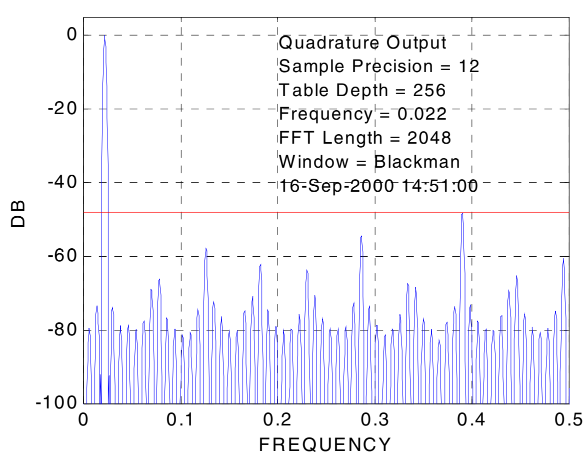


Figure 3: **Exploded View of Figure 2 (c).**

Two observations related to the phase jitter structure level can be made. First, 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. Also, 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.

Figures 4 and 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 four 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 lookup table.

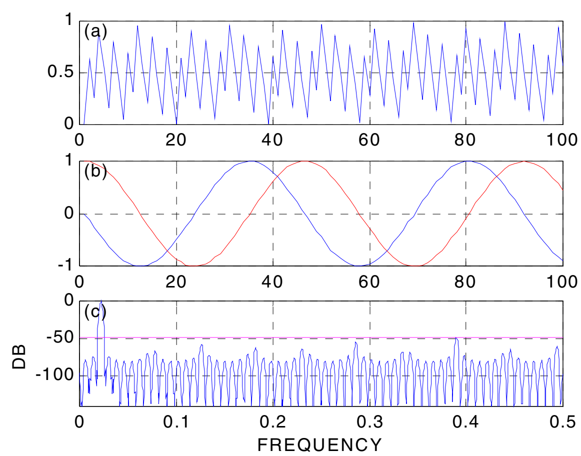


Figure 4: **Phase Truncation DDS. f_{out} = 0.022Hz, Table Depth = 256 16-Bit Precision Samples. (a) Phase Angle Addressing Error (b) Complex Output Time Series (c) Output Spectrum**

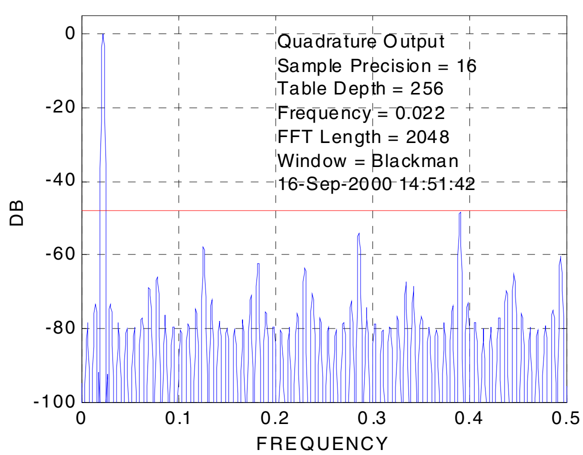


Figure 5: **Exploded View of Figure 4 (c).**

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 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 6 provides an example of this process for a DDS with a table depth N = 1024 and table sample precision of 16 bits. Figure 6 (a) is the phase error generated by taking the difference between the quantizer input and output signals, Figure 6 (b) is the output time series and Figure 6 (c) is the signal output spectrum. Observe in Figure 6 (a) the periodic sawtooth structure of the phase error signal. The line spectrum associated 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 6 (c).

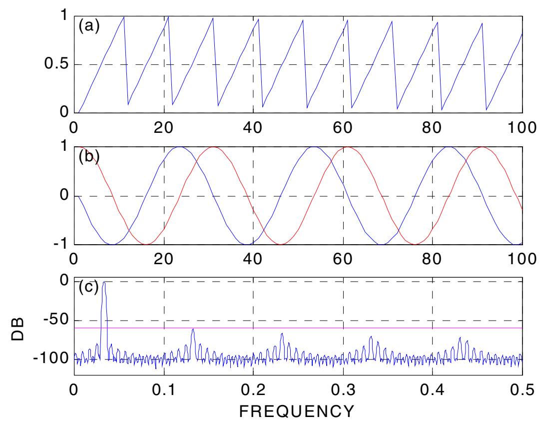


Figure 6: **DDS Plots Showing (a) Phase Error Time Series (b) Complex Output Time Series (c) Output Spectrum. 1024 Deep Lookup Table, 16-Bit Samples, Output Frequency 0.333 Hz.**

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 dithered DDS supplies, approximately, an additional 12 dB of spurious free dynamic range (SFDR) in comparison to a phase truncation design. This is achieved by spreading the spectral energy of the phase error signal. 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 $\lceil S/6 \rceil - 2$.

Figures 7 and 8 provide the results for several dithered DDS simulations. Figure 7 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.

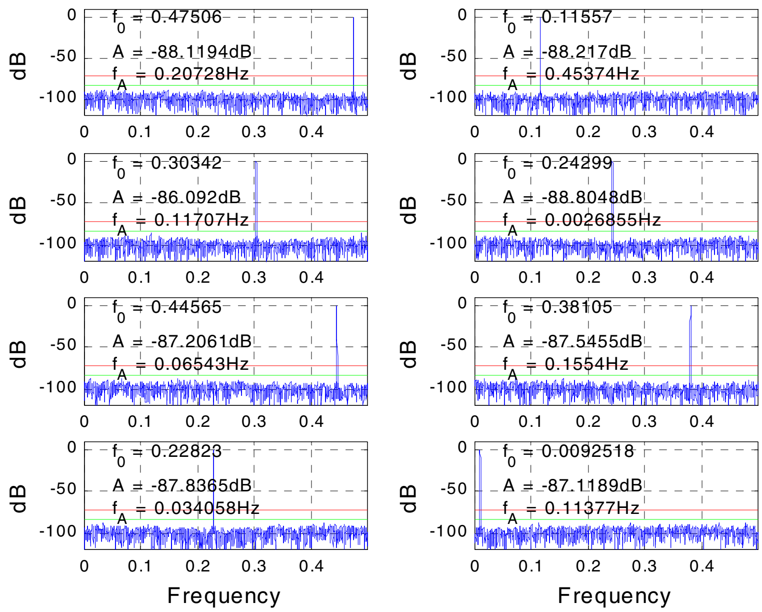


Figure 7: **Dithered DDS Simulations. The DDS configuration is N = 4096, B_S = 16. The eight plots are spectral domain representations for eight different output frequencies. Each plot is annotated with the peak spur.**

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

Figure 8 provides one more dithered DDS simulation 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.

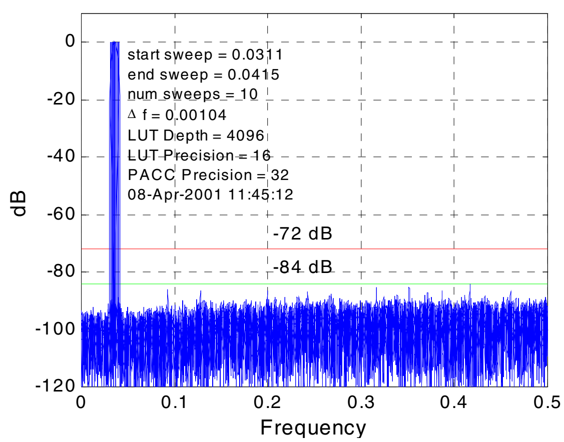


Figure 8: **Example Plot for Dithered DDS Simulation with Frequency Sweep**

In Figure 8, 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.

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 that 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 or XtremeDSP slices in FPGAs, it is now practical to use the previously discarded fractional bits to calculate corrections that can be added to the lookup table values to produce outputs with very high SFDR.

Figures 9 through 12 show the simulation results of four different Taylor series corrected DDS simulations. The Taylor series corrected architecture in this example 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 13 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 are 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 14 shows another frequency sweep simulation with 35 tones over a broader frequency range.

As shown in the plots, linear correction of the RAM values can extend the SFDR to 118 dB using only a single block RAM and three multipliers. To achieve SFDR beyond 118dB, it is necessary to deepen the RAM or to use quadratic correction (an extra term of the Taylor series). Since the RAM size would double for each additional 6dB, the DDS Compiler uses quadratic correction to achieve SFDR values of up to 150dB. Introducing the extra term of the Taylor series expansion of Sine or Cosine requires an additional multiplier per Sine and Cosine output and an additional block RAM to both scale and square the phase error.

Optimization of Memory Usage

The Taylor Series Correction implementation in the DDS Compiler v4.0 core typically results in an SFDR higher than that requested in order to guarantee SFDR. This results in extra block RAMs for values of SFDR above 102dBs. However, in many cases, depending on the phase increment values used, a specified SFDR target value of 102dB will provide higher SDFR, but with one 18k block RAM.

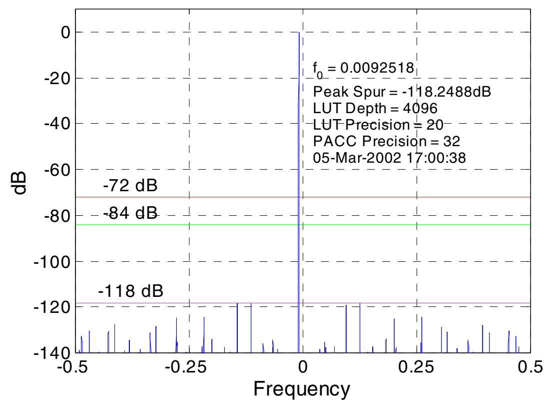


Figure 9: **Taylor Series Corrected DDS – Single-Tone Test, f_0 = 0.0092518**

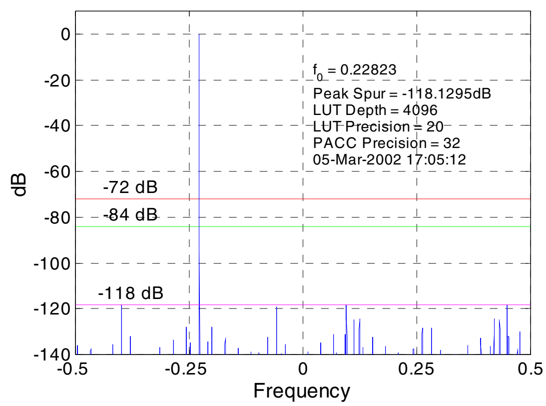


Figure 10: **Taylor Series Corrected DDS – Single-Tone Test, f_0 = 0.22823**

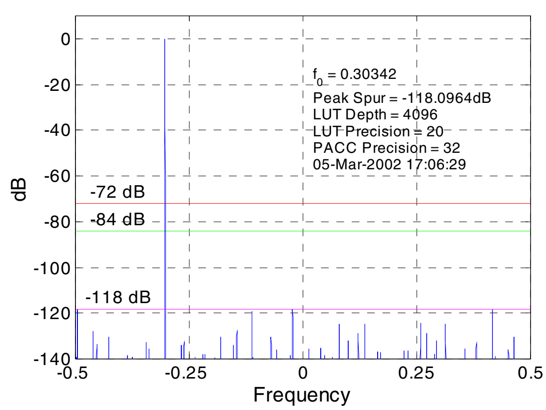


Figure 11: **Taylor Series Corrected DDS – Single-Tone Test, f_0 =0.30342**

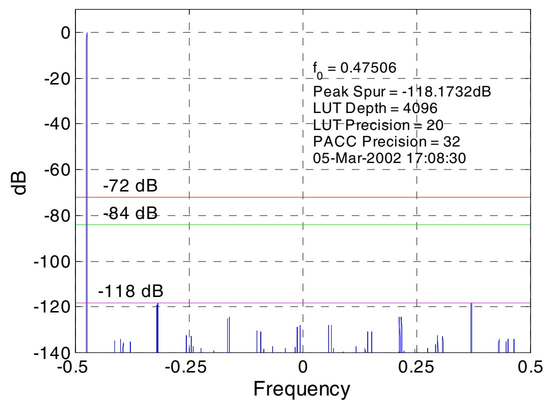


Figure 12: **Taylor Series Corrected DDS – Single-Tone Test, f_0 = 47506**

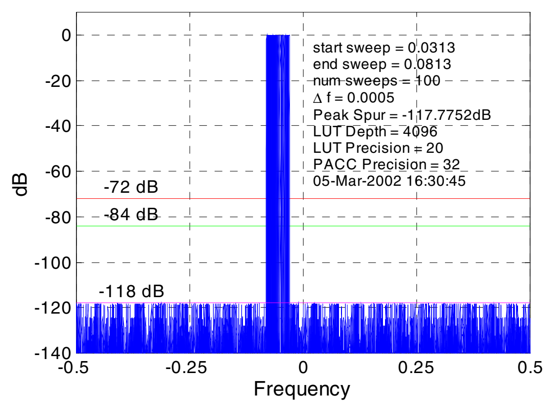


Figure 13: **Taylor Series Corrected DDS – Frequency Sweep Simulation, 100 Tones**

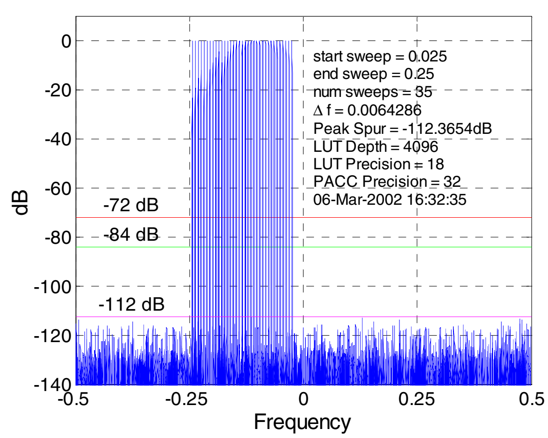


Figure 14: **Taylor Series Corrected DDS – Frequency Sweep Simulation, 35 Tones**

Core Architecture Overview

Figure 15 provides a block diagram of the DDS Compiler core. The core consist of two main parts, a Phase Generator and SIN/COS LUT, which can be used independently or together with an optional dither generator to create a DDS capability. A time-division multi-channel capability is supported, with independently configurable phase increment and offset parameters.

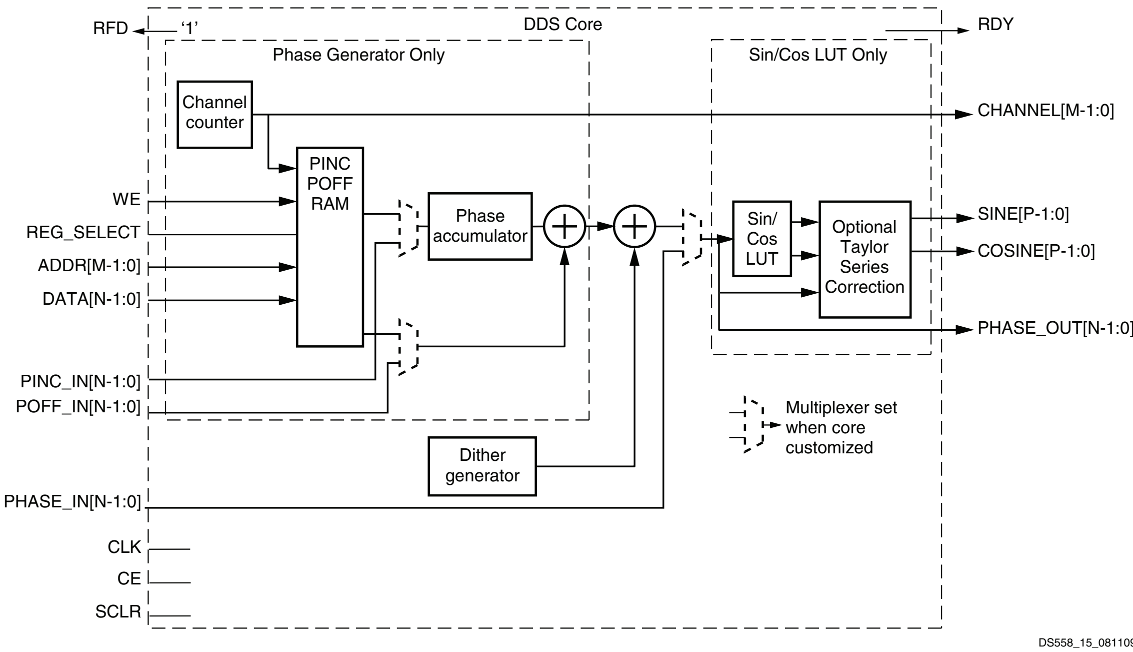


Figure 15: **DDS Core Architecture**

Phase Generator

The Phase Generator consists of an accumulator followed by an optional adder to provide addition of phase offset. When the core is customized the phase increment and offset can be independently configured to be either fixed, programmable or supplied by the PINC_IN and POFF_IN input ports respectively.

When set to programmable, registers are implemented with a bus interface, consisting of ADDR, REG_SELECT, WE, and DATA signals. The address input, ADDR, specifies the channel for which DATA is to be written when multi-channel, with REG_SELECT specifying whether DATA is phase increment or offset.

When set to fixed the DDS output frequency is set when the core is customized and cannot be adjusted once the core is embedded in a design.

SIN/COS LUT

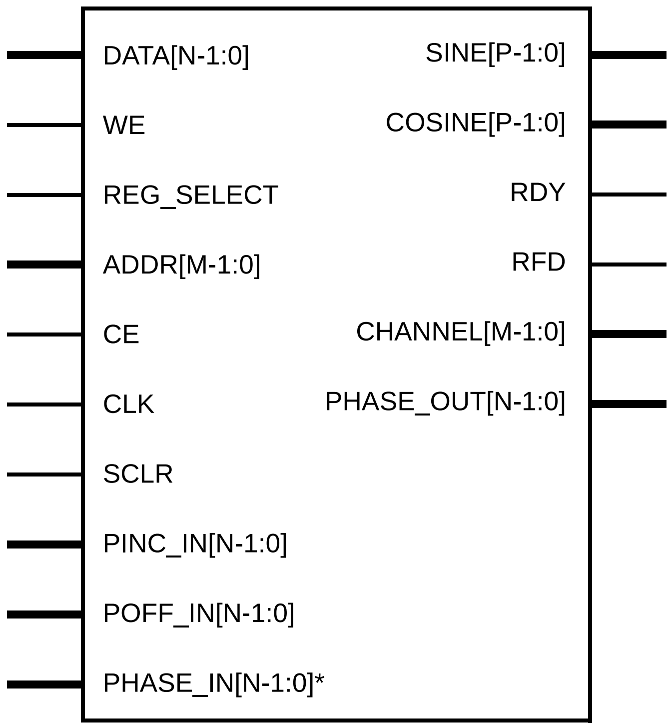
When configured as a SIN/COS LUT, the Phase Generator is not implemented, and the phase is input via the PHASE_IN port, and transformed into the sine and cosine outputs using a look-up table. Efficient memory usage is achieved using halfwave and quarterwave storage schemes. The presence of both outputs and their negation are configurable when the core is customized. Precision can be increased using optional Taylor Series Correction. This exploits XtremeDSP slices on FPGA families that support them to achieve high SFDR with high speed operation.

Phase Generator and SIN/COS LUT (DDS)

The Phase Generator is used in conjunction with the SIN/COS LUT to provide either a Phase Truncated DDS or Taylor Series Corrected DDS. An optional dither generator can be added between the two blocks to provide a Phase Dithered DDS.

Interface, Control, and Timing

The DDS Compiler core pinout is shown in Figure 16. All of the possible pins are shown, though the specific pins in any instance depend upon parameters specified when the core is generated.



*Used only with the SIN/COS LUT

Figure 16: **DDS Symbol**

Table 1 summarizes the pinout of the core. All control inputs are active High. Should active Low input be required for a specific control pin, an inverter must be placed in the path to the pin and will be absorbed appropriately during mapping.

Table 1: **Core Signal Pinout**

Name	Direction	Description
CLK	Input	Rising edge clock
REG_SELECT[1]	Input	Address select for writing DATA to the phase increment (PINC) and the phase offset (POFF) registers. When REG_SELECT = 0, PINC registers are selected. When REG_SELECT = 1, POFF registers are selected. This pin will only exist if both POFF and PINC are programmable.
ADDR[M-1:0][1]	Input	This bus is used to address up to 16 channels. The number of bits in ADDR is 1 for 2 channels, 2 for 3 or 4 channels, 3 for 5 to 8 channels, and 4 for 9 to 16 channels. This bus will exist if either PINC or POFF is programmable and there are multiple channels.

Table 1: **Core Signal Pinout** ***(Cont'd)***

Name	Direction	Description
WE[1]	Input	Write enable - active High. Enables a write operation to the PINC or POFF registers.
CE[1]	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 registers.
DATA[N-1:0][1]	Input	Time shared data bus. The DATA port is used for supplying values to the PINC or POFF memories. The value input to DATA describes a phase angle. DATA port width (N) is determined by the Phase Width parameter, selected or displayed on the GUI. If the bus is N bits wide, a full circle is cut into 2^N phase segments, so the ranges [0, 2^N -1] and [-2^{N-1}, 2^{N-1} -1] both describe a full circle. Because of this it makes no difference if this port is considered signed or unsigned. Both PINC and POFF registers are fixed point. So, if lower precision values are all that is required for one or the other, these values must be right-justified and input to the upper bits of the DATA bus. For example, if DATA width is 8 bits, but POFF only needs to be 1/8, 2/8, 3/8, etc., the 3 bits used to describe the phase offset must be bits 7:5 of the DATA port.
SCLR[1]	Input	Synchronous clear - active High. When SCLR is asserted, the accumulator and channel counter are reset. Depending upon the Latency of the core, RDY may also deasserted (further details given later). The outputs of the core, CHANNEL, SINE, COSINE and PHASE_OUT are undefined following SCLR until RDY is active. SCLR does not reset PINC or POFF registers, and a write to these registers may occur when SCLR is active.
PINC_IN[N-1:0][1]	Input	Streaming input for Phase Increment. This input allows DDS output frequency to be modulated.
POFF_IN[N-1:0][1]	Input	Streaming input for Phase Offset. This input allows DDS output phase to be modulated.
PHASE_IN[N-1:0][1]	Input	For use when the DDS is configured as SIN/COS LUT only. This is the phase input to the SIN/COS LUT.
RDY[1]	Output	Output data ready - active High. Indicates when the output samples are valid.
RFD[1]	Output	Ready for data - active High. RFD is a dataflow control signal present on many Xilinx LogiCORE cores. In the context of the DDS, it is supplied only for consistency with other LogiCORE cores. This optional port is always tied High.
CHANNEL[M-1:0][1]	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 signal. Its width is determined by the number of channels. It is qualified by RDY.
SINE[P-1:0][1]	Output	Sine time-series. Port width (P) is determined by the Output Width parameter.
COSINE[P-1:0][1]	Output	Cosine time-series. Port width (P) is determined by the Output Width parameter.
PHASE_OUT[N-1:0][1]	Output	Phase output, supplied in synchronism with SINE and COSINE outputs.

1. Denotes optional pin.

CORE Generator Graphical User Interface Parameters

Customization parameter definitions:

- **Component Name:** The user-defined DDS component name.
- **Configuration Options:** The full DDS, or optionally the Phase Generator part or SIN/COS Lookup table part may be generated.
    - **Phase_Generator_and_SIN_COS_LUT:** DDS is provided by combining Phase Generator and SIN/COS LUT with an optional Dither circuit.
    - **Phase_Generator_only:** Only the phase generator is provided.
    - **SIN_COS_LUT_only:** Only the SIN/COS LUT with optional Taylor Series Correction circuit is provided.
- **System Requirements:** The general context of the DDS is set by this group of parameters:
    - **System Clock:** The frequency at which the DDS core will be clocked. The value provided influences architectural choices, and is used to calculate the value of phase increment from output frequency (it is the relative value of output frequency to system clock that specifies phase increment, and so doubling System Clock while maintaining output frequency will result in a doubling of phase increment). **The specified clock rate may not be achievable by the final implementation, as this will depend upon the FPGA family and how much is being packed into the device.**
    - **Number of Channels:** The DDS and Phase Generator can support 1 to 16 time-multiplexed channels. The channels are time-multiplexed, which reduces the effective clock frequency per channel.
    - **Frequency per Channel (Fs):** Because of time division multiplexing, the effective system clock to each channel is the real system clock divided by the number of channels.
- **Parameter Selection:** DDS key parameters may be specified using System Parameters, which are aimed at system architects (frequency domain parameters) or Hardware Parameters, which are aimed at hardware engineers (time-domain parameters). The Phase Generator and SIN/COS LUT are only specified in terms of Hardware parameters.
- **System Parameters**
    - **Spurious Free Dynamic Range (SFDR):** The targeted purity of the tone produced by the DDS. This sets the Output Width (as described below) as well as internal bus widths and various implementation decisions.
    - **Frequency Resolution:** Specified in Hz, this specifies the minimum frequency resolution and is used to determine the Phase Width, as employed by the phase accumulator and its associated phase increment (`PINC`) and phase offset (`POFF`) values. Small values will give high frequency resolution and will require larger accumulators. Larger values will reduce hardware resources. Depending upon the choice of Noise Shaping, the Phase Width may be increased, and the frequency resolution higher than that specified.
- **Noise Shaping:** This controls whether phase truncation, dithering, or Taylor series correction is used. The options are:
    - **None:** Phase truncation DDS is produced.
    - **Dithering:** Phase dither is used to improve SFDR at the expense of increased noise floor. See "Phase Dithered DDS."
    - **Taylor Series Corrected:** SIN/COS values are interpolated using the otherwise discarded bits from phase truncation. See "Taylor Series Corrected DDS."
    - **Auto:** Noise-shaping will be automatically determined, based on System Parameters such as SFDR. The selected noise shaping option is presented in the GUI summary pages. Auto is only available when Parameter Selection is System Parameters.

  The availability of particular noise shaping options depends upon the configuration option selected and Parameter Selection method. System Parameter entry automatically constrains whether a particular Noise Shaping option is possible. When Hardware Parameter entry is selected, the options summarized in Table 2 are made available, and the choice of the Noise Shaping option then constrains the hardware parameter to ranges to those supported by the selected option.

Table 2: **Availability of Noise Shaping Options for Hardware Parameters**

Setting	DDS part	Phase Generator part	Sin/Cos LUT part
None	Available	Available	Available
Dithering	Available		
Taylor	Available		Available
Auto	Available		

Based upon the System Parameters entered and Noise Shaping selected, the minimum Phase Width and Output Width are derived by the GUI in the following way. The Phase Width may be increased to enable a particular Noise Shaping option. For example, Taylor Series Correction requires a minimum Phase Width of 12 bits.

$$\text{Phase Width} = \left\lceil \log_2\left(\frac{\text{DDS Clock Rate}}{\text{Channels} \times \text{Frequency Resolution}}\right)\right\rceil$$

Table 3: **Calculation of Output Width from SFDR and Noise Shaping**

Noise Shaping	Output Width
None and Dithering	$\text{Output Width} = \left\lceil \frac{\text{SFDR}}{6} \right\rceil$
Taylor	$\text{Output Width} = \left\lceil \frac{\text{SFDR}}{6} \right\rceil + 1$

Figure 17 shows the regions of SFDR and Phase Width over which each Noise Shaping option operates. There are three overlapping regions for None, Phase Dithering and Taylor Series Correction, and deeper levels of shading have been used to show where regions overlap. The darkest region is where all 3 regions overlap and all 3 noise shaping options are possible. The lower dashed line signifies that Taylor Series Correction is only valid for SDFR > 66.0 dBs (and not 66.0 dBs). As mentioned previously Phase Width may be increased to maximize the number of noise shaping options for a particular SFDR target.

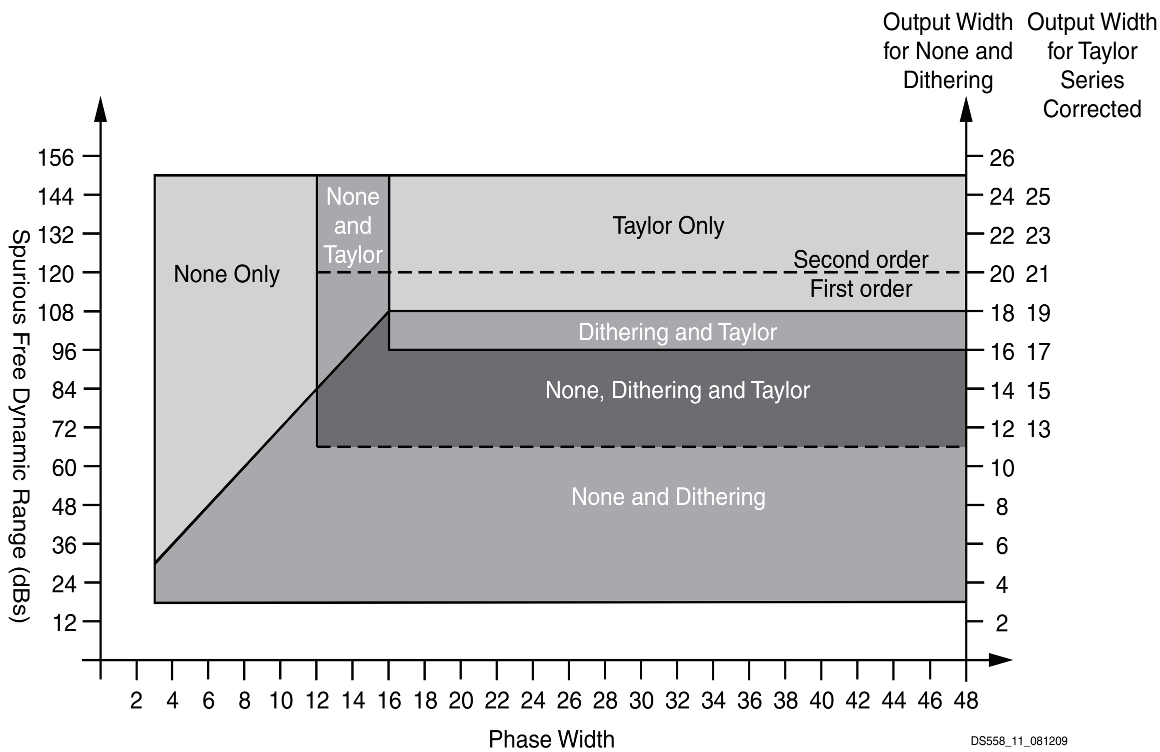


Figure 17: **Noise Shaping Regions**

- **Hardware Parameters:**
  - **Phase Width:** Sets the width of PHASE_OUT, the phase accumulator, associated phase increment and offset registers and programming port DATA input (if present).
  - **Output Width:** Only enabled when DDS or SIN/COS LUT part selected, as it is not required by the Phase Generator part. Sets the width of SINE and COSINE outputs. The SFDR that this provides is dependent upon Noise Shaping option previously selected. The following equations can be used to estimate the SFDR that is achieved:

Table 4: **Calculation of SFDR for given Noise Shaping**

Noise Shaping	SFDR
None, Dither	$SFDR = \text{Output Width} \times 6$
Taylor	$SFDR = (\text{Output Width} - 1) \times 6$

- **Phase Increment Programmability:** Selects the means by which the PINC value is set.
  - **Fixed:** PINC is fixed at generation time and cannot be changed at run-time. Fixed requires minimal resource.
  - **Programmable:** PINC value can be changed at run-time using the ADDR, WE, REG_SELECT and DATA ports. This is recommended when the DDS frequency is to change between modes of operation.
  - **Streaming:** PINC value is taken directly from the PINC_IN port. This is recommended when the PINC value has to change often, or with deterministic latency, for example when frequency modulation is required.
- **Phase Offset Programmability:** Selects the means by which the POFF value is set.

- **None:** No phase offset facility and the required hardware is not generated. This saves FPGA resources.
- **Fixed:** `POFF` is fixed at generation time and cannot be changed at run-time.
- **Programmable:** `POFF` value can be changed at run-time using the `ADDR`, `WE`, `REG_SELECT` and `DATA` ports. This is recommended when the DDS phase is to change between modes of operation.
- **Streaming:** `POFF` value is taken directly from the `POFF_IN` port. This is recommended when the `POFF` value has to change often, or with deterministic latency, for example when phase modulation is required.

- **Output Selection:**
  - **Output_Selection:** The DDS may have a quadrature `SINE` and `COSINE` outputs, or a single output port – either `SINE` or `COSINE`.
  - **Polarity:** The `SINE` and `COSINE` outputs can be inverted. This allows conversion of a DDS used as a transmitter mixer to a receiver mixer, using conjugated outputs; hence both instantiations would be identical except for the values of the two selections here.
    - **Negative Sine:** Checking this selection will result in the `SINE` output being negated at run-time.
    - **Negative Cosine:** Checking this selection will result in the `COSINE` output being negated at run-time.
  - **Amplitude Mode:** This selection allows for one of two amplitudes from the DDS.
    - **Full Range:** Aimed at communications applications where the maximum amplitude within the two's complement representation is desired, but the exact value of amplitude is not very important (indeed, the target amplitude is $1\text{-}2^{(\text{Output Width-2})}$ in most cases).
    - **Unit Circle:** For applications where the exact amplitude of the DDS output is important, say for FFT twiddle factor generation. When Unit Circle, the DDS output amplitude will be half full range (that is, values will range from 01000..(+0.5). to 110000..(-0.5)). As the amplitude is reduced over Full Range by a factor of 2, the SDFR will be reduced by 6dBs. Increase SFDR or Output Width to accommodate this requirement.
- **Implementation Options**
  - **Memory Type:** This controls the implementation of the SIN/COS Lookup Table. The Auto setting will select Distributed ROM for small cases where the SIN/COS table can be contained in a single layer of memory and will select Block ROM for larger cases. (That is, Distributed ROM will be selected when Phase Width $\leq$5-bits on FPGA devices that support 6-input LUT, and Phase Width < 5-bits on other devices). This selection can be overridden by selecting Distributed ROM or Block ROM explicitly.
  - **Optimization Goal:** In some cases, circuit clock speed can be increased at the expense of extra pipelining registers. This selection controls whether the implementation decisions will target highest speed or lowest resource.
  - **DSP48 Use:** This controls the implementation of the phase accumulator and following addition stages (for phase offset and/or dither noise addition). When set to Minimal, the phase accumulator and following stages will be implemented in fabric. When Maximal, all will be implemented using XtremeDSP slices. In the case of single channel, the XtremeDSP slice can also provide the register to store programmable phase increment and/or phase offset and thereby save further fabric resources. This will not be done if either phase increment or phase offset is Streaming and only when Optimization Goal is Area. When this optimization is performed, the initial value of the `PINC` and/or `POFF` register must be zero. This is enforced by the GUI by setting the initial value of `PINC` and/or `POFF` to zero and disabling entry.
- **Latency Options:** Select whether Latency should be configured automatically by the GUI or manually:
  - **Auto:** Will cause the DDS to be pipelined for optimal performance (taking into account the Optimization Goal).
  - **Configurable:** Where optimal performance is beyond requirements, Latency may be set to configurable and a smaller value of latency selected. This will reduce the number of pipeline stages and will generally result in resource savings. A minimum value of latency is imposed, where a cycle of latency arises from each of the following sources:

    - Streaming phase increment (that is, use of PINC_IN)
    - Block ROM within SIN/COS LUT (can be avoided by selecting Distributed ROM).
    - Block ROM within second order Taylor Series Correction (used for SFDR above 120dBs).

- **Optional Pins:** Certain inputs and outputs may be disabled to save resources.
    - **Has Phase Out:** When checked the core will have the PHASE_OUT output port. This is provided in synchronism with SINE and COSINE outputs.
    - **Clock Enable:** When checked the core will have a CE port.
    - **Synchronous Clear:** When checked the core will have an SCLR port.
    - **RDY:** When checked the core will have the RDY output port that validates the SINE and COSINE outputs.
    - **RFD:** When checked the core will have an RFD port. This is for completeness. The DDS is always ready for data.
    - **Channel Pins:** For multi-channel operation, check to obtain an output that provides the index of the channel currently on the outputs. Only available on DDS and Phase Generator parts.
- **Parameter Entry Pages:** The following pages appear for entry of parameters when either Phase Increment or Phase Offset are either Fixed or Programmable. If Programmable, the initial value of the register is specified through the Parameter Entry Pages. If an XtremeDSP register is used, as described under DSP Use, the initial value of phase increment and/or offset is assumed to be zero.

  System Parameters:

    - **Output Frequencies:** This page appears when Parameter Selection is set to System Parameters and Phase Increment Programmability is Fixed or Programmable. For each channel, an independent frequency (MHz) can be entered into the table. The allowable range is displayed as 0 to Fs (where Fs is the frequency per channel). Values from Fs/2 to Fs will alias to -Fs/2 to 0 respectively, so can be used to input negative frequencies.
    - **Phase Offset Angles:** This page appears when Parameter Selection is set to System Parameters and Phase Offset is set to Fixed or Programmable. This table allows the phase offset to be specified for each channel as a fraction of a cycle. The valid range is -1.0 to 1.0. For example enter 0.5 for 180 degrees (that is, π radians). This range is greater than a single cycle, but is allowed, as negative values will map to equivalent positive values.

  Hardware Parameters:

    - **Phase Angle Increment Values:** This page appears when Parameter Selection is set to Hardware Parameters and Phase Increment Programmability is Fixed or Programmable. Values must be entered in binary. The range is 0 to the weight of the accumulator, that is, $2^{\text{Phase Width}}-1$, which corresponds to a single cycle. The angle in radians can be obtained by converting the unsigned fractional number to decimal and multiplying by 2π. Entries will be extended to Phase Width bits by zero padding to the left.
    - **Phase Offset Values:** This page appears when Parameter Selection is set to Hardware Parameters and Phase Offset is set to Fixed or Programmable. Values must be entered in binary. The range is 0 to the weight of the accumulator, that is, $2^{\text{Phase Width}}-1$, which corresponds to a single cycle. The angle in radians can be obtained by converting the unsigned fractional number to decimal and multiplying by 2π. Entries will be extended to Phase Width bits by zero padding to the left.
- **Summary (2 pages):** The final two pages of the GUI are devoted to feedback fields.
    - **Summary (Page 1):** This page presents the resolved values of the selected part. For instance, these fields indicate the result of automatic memory type and latency allocation. They also indicate the expected SFDR and frequency resolution for the DDS when hardware parameters are used for input, or vice versa. There are also resource estimates (XtremeDSP slices and 18kbit block RAM primitives).
    - **Summary (Page 2):** This is only presented when Phase Increment and/or Phase Offset are fixed or programmable, and provides a summary of the hexadecimal values used to obtain a particular frequency or phase offset. The actual value of frequency and phase (the latter as a fraction of a cycle) is also given as a floating-point number.

System Generator for DSP Graphical User Interface

This section describes the System Generator for DSP GUI and details the parameters that differ from the CORE Generator GUI. The DDS Compiler core may be found in the Xilinx Blockset in the DSP section. The block is called "DDS Compiler v4.0." See the System Generator for DSP Help page for the "DDS Compiler v4.0" block for more information on parameters not mentioned here.

The System Generator for DSP GUI offers the same parameters as the CORE Generator GUI. However, whereas in CORE Generator the Hardware Parameters are hidden when System Parameter entry is selected, in System Generator for DSP GUI the Hardware Parameters are simply disabled. Likewise, System Parameters are disabled when Hardware Parameter entry is selected.

Using the DDS Compiler IP Core

Simulation Models

The core has a number of options for simulation models:

- VHDL RTL-based simulation model
- Verilog UniSim-based structural simulation model

The models required may be selected in the CORE Generator software project options.

Xilinx recommends that simulations utilizing UniSim-based structural models are run using a resolution of 1ps. Some Xilinx library components require a 1ps resolution in either functional or timing simulation. The UniSim-based structural models may produce incorrect results if simulated with a resolution other than 1ps. See the "Register Transfer Level (RTL) Simulation Using Xilinx Libraries" section in *Chapter 6* of the Synthesis and Simulation Design Guide. This document is part of the ISE® Software Manuals set available at www.xilinx.com/support/software_manuals.htm.

XCO File Parameters

The XCO parameters are summarized in Table 5.

Table 5: **XCO Parameters**

GUI Field	XCO Parameter	XCO Values(1)	Description
Component name	Component_Name	string	The name of the CORE Generator instance
Configuration Options	PartsPresent	**Phase_Generator_and_SIN_COS_LUT**, Phase_Generator_only, SIN_COS_LUT_only	Allows for parts of DDS to be instanced separately
Parameter Selection	Parameter_Entry	**System_Parameters**, Hardware_Parameters	
Number of Channels	Channels	Integer, **1** to 16	
System Clock	DDS_Clock_Rate	0.01 to 550, default is **100**	MHz
Spurious Free Dynamic Range	Spurious_Free_Dynamic_Range	18(2)to 150, default is **36**	dB
Frequency Resolution	Frequency_Resolution	Frequency per Channel divided by $2^{\text{Phase Width}}$ to Frequency per Channel/2^{48}, default is **0.4**Hz	Hz
Noise Shaping	Noise_Shaping	**Auto**, None, Phase_Dithering, Taylor_Series_Corrected	

Table 5: **XCO Parameters** ***(Cont'd)***

GUI Field	XCO Parameter	XCO Values[1]	Description
Output Width	Output_Width	3[3]to 26, default is **6**	Defines the output width, hence precision
Phase Width	Phase_Width	3 to 48. Virtex-4 is limited to a width of 36 bits. Default is **16**	Defined width of phase buses hence frequency resolution
Phase Increment Programmability	Phase_Increment	**Fixed**, Programmable, Streaming	
Phase Offset Programmability	Phase_offset	**None**, Fixed Programmable, Streaming	
Output Selection	Output_Selection	Sine, Cosine, **Sine_and_Cosine**	
Negative Sine	Negative_Sine	**false**, true	
Negative Cosine	Negative_Cosine	**false**, true	
Amplitude Mode	Amplitude_Mode	**Full_Range**, Unit_Circle	Selects maximum possible amplitude or exact power-of-two amplitude
Memory Type	Memory_Type	**Auto**, Distributed_ROM, Block_ROM	
Optimization Goal	Optimization_Goal	**Auto**, Area, Speed	
DSP48 Use	DSP48_Use	**Minimal**, Maximal	
Latency Options	Latency_Configuration	**Auto**, Configurable	
	Latency	0,1,...15	
Has Phase Out	Has_Phase_Out	false, **true**	
Clock Enable	Clock_Enable	**false**, true	
Synchronous Clear	SCLR_Pin	**false**, true	
RDY	RDY	**false**, true	
RFD	RFD	**false**, true	
Channel Pins	Channel_Pin	**false**, true	
Output Frequencies	Output_Frequency1, Output_Frequency2,... Output_Frequency16	**0.0** to Fs (Fs=DDS_Clock_Rate/ Channels)	MHz. Frequencies above the Nyquist limit [Fs/2 to Fs] will alias to negative frequencies [-Fs/2 to 0] respectively.
Phase Offset Angles	Phase_Offset_Angles1, Phase_Offset_Angles2,... Phase_Offset_Angles16	real (-1.0 to +1.0), default is **0.0**	-0.5 is -180 degrees, +0.5 is +180 degrees
Phase Angle Increment Values	PINC1, PINC2, ..., PINC16	**0** to 2**Phase_Width -1	Unsigned binary
Phase Angle Offset Values	POFF1, POFF2, ..., POFF16	**0** to 2**Phase_Width -1	Unsigned binary

1. Default value highlighted in bold.
2. Note that SDFR must be greater than 18 dBs if Phase Dithering is to be used.
3. The minimum value for Phase Width and Output Width is 4 if Phase Dithering is to be used.

Core Timing

Programming Interface

Figure 18 shows a timing sequence for a single-channel DDS core with programming interface. The prefix 0x has been used to indicate that numbers are in hexadecimal. In this example, the DDS has SCLR, CE, and PHASE_OUT ports, and programmable phase increment (PINC) and a phase offset (POFF). Registers are implemented for both PINC and POFF within the core, and a programming interface provided.

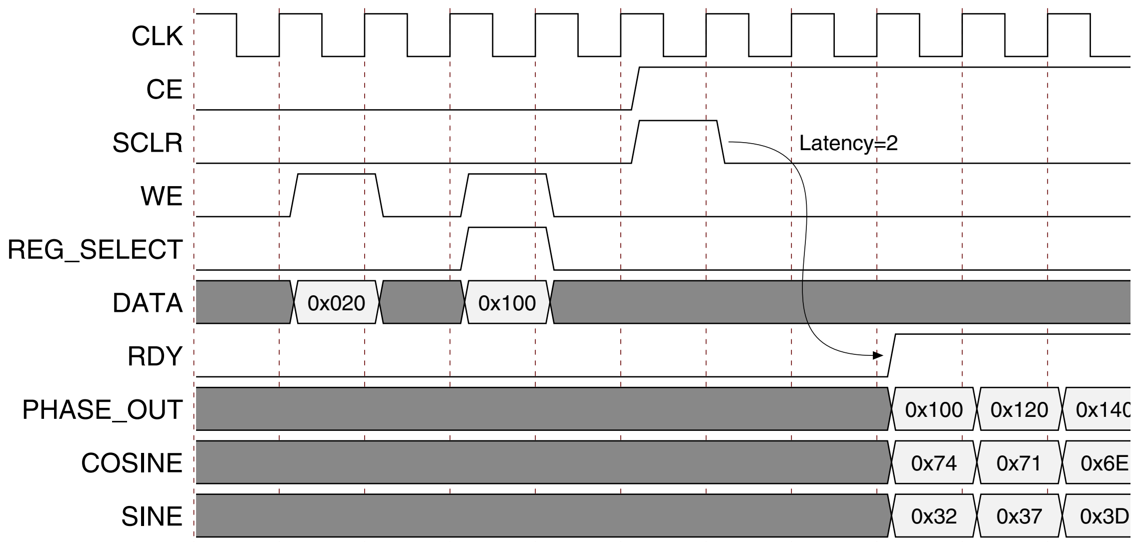


Figure 18: **DDS Timing: Single Channel Programmable**

In this example, PINC memory is first written. This is achieved by supplying the phase increment value on the DATA port and addressing the PINC memory by setting REG_SELECT = 0. Since this example is single-channel, the ADDR port is not present so need not be considered (otherwise, the channel number would have been placed on the ADDR bus). The write is performed on the positive clock edge when WE is active; that is, WE = 1. After the PINC memory is loaded, a value is written to the POFF memory. This requires REG_SELECT = 1 and WE = 1.

CE does not have to be active to write either the PINC or POFF memories (and indeed, here it is shown inactive). The DDS starts operating after the clock enable is applied (CE = 1). Since CE is an optional pin, DDS configurations that do not include this pin will begin operating after the FPGA is configured and the system clock is active.

After a start-up latency (measured from the assertion of CE[(1)]) 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. In this case, SCLR is also asserted, then deasserted. The latency of the core determines the delay from the deassertion of SCLR to the assertion of RDY. As SCLR is not gated by CE, it would be possible to delay CE by 1 cycle so that CE = 0 when SCLR = 1 and obtain the same output.

The assertion by the core of RDY indicates the first valid output sample. As illustrated in Figure 18, valid samples begin appearing at the output ports when RDY goes High. In this case, the latency, measured from the deassertion of SCLR, to output is 2 cycles.

1. Assuming this port is present.

SCLR, when present, will reset the control path (in this case just the RDY logic), the accumulator value and dither circuit (if present). It does not affect the rest of the datapath. As such, the core may output stale data until RDY is asserted. If necessary, outputs can be qualified by RDY.

The value of phase for the n^{th} cycle is given by:

$$\text{PHASE\_OUT}(n) = \text{PINC}*n + \text{POFF}$$

In this example, PINC = 0x020, and POFF = 0x100, so PHASE_OUT = 0x100, 0x020 + 0x100, 0x020*2+ 0x100....

To synchronize the output, SCLR may be applied, during or after programming of PINC and/or POFF. As PINC and POFF programming may have up to 2 cycles of latency, SCLR should be deasserted a minimum of 1 cycle after the last register write (as in this example).

It is possible, to write the PINC and POFF registers during DDS operation in order to modulate the frequency and phase of the DDS. However, the latency of those register value changes depends upon the configuration of the core. The behavioral model can be used to establish this timing.

Streaming Interface

Phase increment and/or offset may be set to streaming to provide a direct way to modulate frequency and/or phase offset. When this is done, the phase increment and offset are supplied through the input ports PINC_IN and POFF_IN respectively. There is a subtle change in behavior when applying the phase increment as a stream, and that is the phase for the n^{th} cycle is given by:

$$\text{PHASE\_OUT}(n) = \sum_{i=0}^{n} \text{PINC\_IN}(n) + \text{POFF}(n)$$

That is, the phase increment value is used immediately to calculate phase, and the first output is PINC_IN(0) + POFF_IN(0), rather than POFF_IN(0) as when the phase increment is programmable.

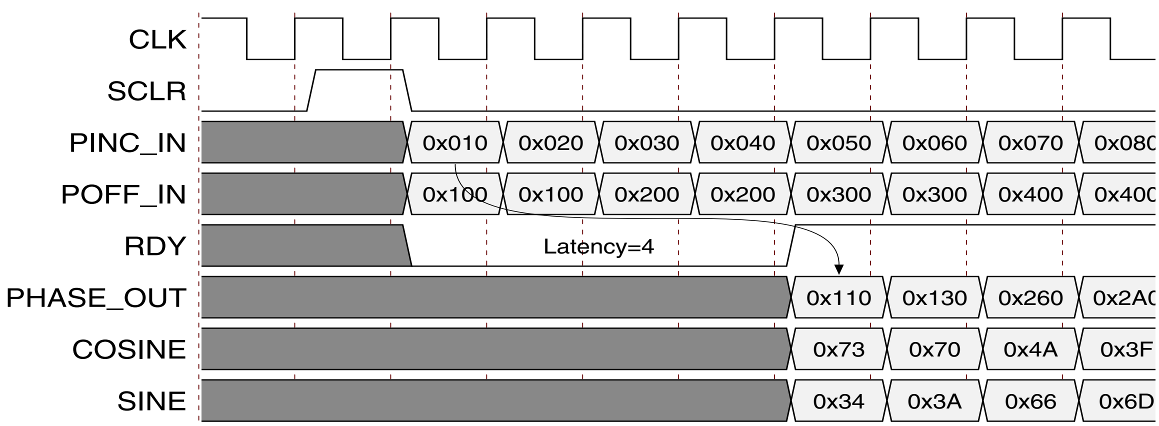


Figure 19: **DDS Timing: Single Channel Streaming**

Multi-Channel

When configured for more than 1 channel, the DDS, or Phase Generator part, will generate outputs for each channel in a time-multiplexed fashion. As such, the output for a particular channel will be given every N-cycles, where N is the number of channels selected when the core was customized. The outputs for channel 0 are given first. The CHANNEL output can be enabled through the GUI to provide the index of the channel at the output.

SCLR causes the channel counter to be reset, and so provides a means of synchronizing the input and output of a multi-channel DDS or Phase Generator. For streaming input, the phase relating to channel 0 should be applied on the first cycle after SCLR.

Latency

The latency of the core can be specified through the GUI, or be automatically set to the optimum value based upon the Optimization Goal.

For streaming inputs (PHASE_IN, PINC_IN and POFF_IN) the latency specifies the number of cycles between input and its associated output (see Streaming Interface for an example).

For non-streaming inputs, the latency is measured relative to SCLR, and is the time from the first cycle SCLR is deasserted, to the associated output (see Programming Interface for an example).

When Latency = 0, RDY will be activated by SCLR. This indicates that once SCLR is de-activated the output is valid.

There is further latency in the path to program PINC and POFF registers. It takes one cycle to write the data to the register, and a second cycle when the Optimization Goal is Speed, to pipeline the bus signals.

Migrating to Version 4.0

The CORE Generator core update feature may be used to update an existing DDS Compiler XCO file from a previous version to version 4.0. The core may then be regenerated to create a new netlist. See the CORE Generator documentation on this feature. There have been significant changes to version 4.0, and the version information file available through CORE Generator should be consulted before using the update feature.

Performance and Resource Utilization

Tables 6 through 13 show the performance of the DDS Compiler core in terms of resource usage and maximum achieved operating frequency.

All examples are for the DDS, have SCLR and CE ports, SINE and COSINE output ports without negation, full range amplitude, block RAM memory type and programmable frequency. All other parameters are specified in the tables. As indicated by the name of the case in the second table for each family, the case aims to provide a given SFDR.

These results were obtained with ISE v11.3 tools. The resource count and speed of the core can change depending on the surrounding circuitry of your design. Therefore, these figures should be taken only as a guide.

The maximum clock frequency results were obtained by double-registering input and output ports (using IOB flip-flops) to reduce dependence on I/O placement. The first level of registers used a separate clock signal to measure the path from the input registers to the first output wrapper register through the core.

The resource usage results do not include the aforementioned wrapper registers and represent the true logic used by the core. LUT counts include SRL16s or SRL32s (according to device family).

The tool settings to achieve these results were as follows:

```
map -ol high
par -ol high
```

Note: The tool settings can have a significant effect on area use and speed. The Xilinx Xplorer script can be used to find the optimal settings.

The Virtex-6 FPGA test cases in Table 6 used an XC6VLX75T-FF484 (-1 speed grade) device and ISE speed file version "ADVANCED 1.01g 2009-07-27."

Table 6: **Virtex-6 Family Performance and Resource Utilization**

Description	Small	Medium	Large	Taylor	Taylor Multi-Channel
Phase Width	8	23	30	20	20
Output Width	6	16	17	18	18
Noise shaping	None	None	Dithering	Taylor	Taylor
Channels	1	1	1	1	16
Use_DSP48	Minimal	Minimal	Maximal	Maximal	Maximal
Optimization	Speed	Area	Speed	Speed	Speed
Latency	2	2	Auto	Auto	Auto
Phase Offset	None	None	Fixed	None	None
Phase Angle Width	8	12	15	11	11
LUT FF pairs	17	88	144	87	131
LUTs	16	87	140	85	124
FFs	25	80	226	172	218
Block RAM36/18s	0/1	0/1	4/0	0/1	0/1
DSP48E1s	0	0	2	4	4
Frequency (MHz)	400	276	400	400	400

Table 7: **Virtex-6 Family Performance and Resource Utilization**

Description	SFDR70		SFDR84		SFDR110		SFDR140	
Phase Width	12		12		12		25	
Output Width	12		14		20		25	
Noise Shaping	None		Dithering		Taylor		Taylor	
Channels	1		1		1		1	
Optimization Goal	Speed		Speed		Speed		Speed	
Latency	Auto		Auto		Auto		Auto	
Phase Offset	None		None		None		None	
Phase Angle Width	12		12		11		11	
Use DSP48	Minimal	Maximal	Minimal	Maximal	Minimal	Maximal	Minimal	Maximal
LUT FF pairs	93	86	120	103	123	91	206	139
LUTs	76	71	118	98	120	89	201	134
FFs	131	131	167	155	201	180	325	271
Block RAM36/18s	0/1	0/1	0/1	0/1	1/0	1/0	1/1	1/1
DSP48E1s	0	1	0	2	3	4	5	6
Frequency (MHz)	400	400	400	400	400	400	400	400

The Virtex-5 FPGA test cases in Table 8 used an XC5VLX50T-FF1136 (-1 speed grade) device and ISE speed file version "PRODUCTION 1.65 2009-07-27, STEPPING level 0.

Table 8: **Virtex-5 Family Performance and Resource Utilization**

Description	Small	Medium	Large	Taylor	Taylor Multi-Channel
Phase Width	8	23	30	20	20
Output Width	6	16	17	18	18
Noise Shaping	None	None	Dithering	Taylor	Taylor
Channels	1	1	1	1	16
Use DSP48	Minimal	Minimal	Maximal	Maximal	Maximal
Optimization Goal	Speed	Area	Speed	Speed	Speed
Latency	2	2	Auto	Auto	Auto
Phase Offset	None	None	Fixed	None	None
Phase Angle Width	8	12	15	11	11
LUT FF pairs	25	100	237	172	231
LUTs	8	75	101	19	64
FFs	25	80	226	172	218
Block RAM36/18s	0/1	0/1	4/0	0/1	0/1
DSP48Es	0	0	2	4	4
Frequency (MHz)	450	259	450	450	433

Table 9: **Virtex-5 Family Performance and Resource Utilization**

Description	SFDR70		SFDR84		SFDR110		SFDR140	
Phase Width	12		12		12		25	
Output Width	12		14		20		25	
Noise Shaping	None		Dithering		Taylor		Taylor	
Channels	1		1		1		1	
Optimization Goal	Speed		Speed		Speed		Speed	
Latency	Auto		Auto		Auto		Auto	
Phase Offset	None		None		None		None	
Phase Angle Width	12		12		11		11	
Use DSP48	Minimal	Maximal	Minimal	Maximal	Minimal	Maximal	Minimal	Maximal
LUT FF pairs	131	131	174	161	201	180	325	271
LUTs	58	46	104	80	60	19	108	29
FFs	131	131	167	155	201	180	325	271
Block RAM36/18s	0/1	0/1	0/1	0/1	1/0	1/0	1/1	1/1
DSP48Es	0	1	0	2	3	4	5	6
Frequency (MHz)	450	450	450	450	450	450	448	447

The Spartan-6 FPGA test cases in Table 10 used an XC6SLX45-FGG484 (-2 speed grade) device and ISE speed file version "ADVANCED 1.01e 2009-07-27."

Table 10: **Spartan-6 Family Performance and Resource Utilization**

Description	Small	Medium	Large Dither with Offset	Taylor	Taylor Multi-Channel
Phase Width	8	23	30	20	20
Output Width	6	16	17	18	18
Noise shaping	None	None	Dithering	Taylor	Taylor
Channels	1	1	1	1	16
Use_DSP48	Minimal	Minimal	Maximal	Maximal	Maximal
Optimization	Speed	Area	Speed	Speed	Speed
Latency	2	2	Auto	Auto	Auto
Phase Offset	None	None	Fixed	None	None
Phase Angle Width	8	12	15	11	11
LUT FF pairs	17	81	146	87	130
LUTs	16	80	123	85	125
FFs	25	80	226	172	218
Block RAM18/9s	0/1	1/0	8/0	1/0	1/0
DSP48A1s	0	0	3	4	4
Frequency (MHz)	250	179	250	250	250

Table 11: **Spartan-6 Family Performance and Resource Utilization**

Description	SFDR70		SFDR84		SFDR110		SFDR140	
Phase Width	12		12		12		25	
Output Width	12		14		20		25	
Noise Shaping	None		Dithering		Taylor		Taylor	
Channels	1		1		1		1	
Optimization Goal	Speed		Speed		Speed		Speed	
Latency	Auto		Auto		Auto		Auto	
Phase Offset	None		None		None		None	
Phase Angle Width	12		12		11		11	
Use DSP48	Minimal	Maximal	Minimal	Maximal	Minimal	Maximal	Minimal	Maximal
LUT FF pairs	93	84	118	99	123	91	205	143
LUTs	75	73	111	94	120	89	201	130
FFs	131	131	167	155	201	180	325	271
Block RAM18/9s	1/0	1/0	1/0	1/0	1/1	1/1	2/1	2/1
DSP48A1s	0	1	0	2	3	4	5	6
Frequency (MHz)	250	250	250	250	250	250	250	250

The Spartan-3A DSP FPGA test cases in Table 12 used an XC3SD3400A-FG676 (-4 speed grade) device and ISE speed file version “PRODUCTION 1.33 2009-07-27.

Table 12: **Spartan-3A DSP Family Performance and Resource Utilization**

Description	Small	Medium	Large Dither with Offset	Taylor	Taylor Multi-Channel
Phase Width	8	23	30	20	20
Output Width	6	16	17	18	18
Noise Shaping	None	None	Dithering	Taylor	Taylor
Channels	1	1	1	1	16
Use DSP48	Minimal	Minimal	Maximal	Maximal	Maximal
Optimization Goal	Speed	Area	Speed	Speed	Speed
Latency	2	2	Auto	Auto	Auto
Phase Offset	None	None	Fixed	None	None
Phase Angle Width	8	12	15	11	11
Slices	25	52	139	87	154
LUTs	8	76	150	55	148
FFs	37	80	226	172	218
Block RAM18s	1	1	8	1	1
DSP48As	0	0	3	4	4
Frequency (MHz)	250	142	223	250	245

Table 13: **Spartan-3A DSP Family Performance and Resource Utilization**

Description	SFDR70		SFDR84		SFDR110		SFDR140	
Phase Width	12		12		12		25	
Output Width	12		14		20		25	
Noise Shaping	None		Dithering		Taylor		Taylor	
Channels	1		1		1		1	
Optimization Goal	Speed		Speed		Speed		Speed	
Latency	Auto		Auto		Auto		Auto	
Phase Offset	None		None		None		None	
Phase Angle Width	12		12		11		11	
Use DSP48	Minimal	Maximal	Minimal	Maximal	Minimal	Maximal	Minimal	Maximal
Slices	80	80	104	98	102	91	178	148
LUTs	81	69	139	115	100	59	158	79
FFs	131	131	169	157	201	180	343	289
Block RAM18s	1	1	1	1	2	2	3	3
DSP48As	0	1	0	2	3	4	5	6
Frequency (MHz)	216	212	221	214	250	250	244	247

Known Issues

Sub-Harmonic Frequencies

The equations for SFDR rely on the assumption that rounding errors from the finite precision of phase and amplitude are incoherent. This assumption is violated for values of Phase Increment that are not mutually prime with the weight of the Phase Accumulator. The anomalies, such as spurii, will be more obvious for larger common factors between the Phase Increment Value (PINC) and the weight of the accumulator (2^{Phase_Width}). This is because such values may not access every location in the SIN/COS Lookup table, so the rounding errors are not randomly spread. To avoid this, do not use values of Output Frequency that are simple rational fractions of the frequency per channel, Fs, such as 3/8, 1/64.

Design Examples

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 preceding requirements can be entered into the GUI directly without having to calculate low-level core details. The GUI also provides feedback of the hardware parameters so the translation of system-level parameters to low-level parameters can been seen. Alternatively, as of v4.0, hardware parameters may be entered directly.

The binary point in the POFF and PINC registers is in a fixed point, so if less precision is required on the POFF register, the data values must be shifted up to be right justified in the POFF register.

Example 1

For a single-channel DDS with 1 MHz system clock, frequency resolution of 1 Hz, Phase Width is 20-bits. To synthesize an output of 23.4 kHz, an Output Frequency value of 0.0234MHz must be entered into the GUI, which then returns a value of 5FD8 in hexadecimal, which is 24536 in decimal.

This will give a synthesized frequency of $24536/2^{20}$ *1 MHz = 23399.35 Hz.

If the application requires this to be modulated by one of 8 phase offsets, the phase offset bus need only be 3 bits precision, but these must be the top 3 bits of the DATA input. Hence, the phase offset of 1/8 of a cycle would be entered as 0.125 in the GUI. This returns a value of 20000(hex). This could be entered on the 3-bit bus as 001(binary). The remaining 17 bits of the DATA bus may be disregarded, as they will default to zero.

Example 2 (DDS Requiring Negative Frequencies)

For a 4-channel DDS with 100MHz System Clock, Frequency Resolution of 1Hz the Phase Width is 25-bits. Frequencies of -3MHz, -1MHz, 1 MHz and 3MHz are required. Fs is the frequency per channel which is System Clock/Number of Channels, that is, 25MHz. The negative frequencies alias to every Fs Hz. The legal range to enter in the GUI is 0 to Fs, so the entered frequencies for this example must be 22MHz (Fs-3MHz), 24MHz (Fs-1MHz), 1MHz and 3MHz respectively.

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

See the *IP Release Notes Guide* XTP025 for further information on this core. On the first page there is a link to "All DSP IP." The relevant core can then be selected from the displayed list.

For each core, there is a link to the master Answer Record that contains the Release Notes and Known Issues list. The following information is listed for each version of the core:

- New Features
- Bug Fixes
- Known Issues

Ordering Information

This LogiCORE IP module is included at no additional cost with the Xilinx ISE Design Suite software and is provided under the terms of the Xilinx End User License Agreement. Use the CORE Generator software included with the ISE Design Suite to generate the core. For more information, please visit the core page.

Information about additional Xilinx LogiCORE modules is available at the Xilinx IP Center. For pricing and availability of other Xilinx LogiCORE modules and software, please contact your local Xilinx sales representative.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/28/06	1.0	Initial Xilinx release.
11/30/06	1.1	Updated for core release version 1.1.
05/17/07	2.0	Updated for core release version 2.0 and for support to Spartan-3A DSP FPGAs.
03/24/08	2.1	Updated for core release version 2.1.
06/24/09	3.0	Updated for core release version 3.0 and support for Virtex-6 and Spartan-6.
09/16/09	4.0	Updated for core release version 4.0.
12/02/09	4.1	Replaced SFDR50 with SFDR140 case in Performance section and corrected frequency in Example 2.
04/19/10	4.2	Added Simulator Support section.
03/01/11	4.3	Support added for Virtex-7 and Kintex-7. ISE Design Suite 13.1

Notice of Disclaimer