HB0078
Handbook
CoreDDS v3.1

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 3.0

Added PolarFire® SoC support.

1.2 Revision 2.0

The following is a summary of the changes in revision 2.0 of CoreDDS v3.0.

- Introduced GUI
- Added Taylor series phase correction
- Improved dither generator randomness
- Added frequency and phase modulation ports

1.3 Revision 1.0

The First publication of this document.

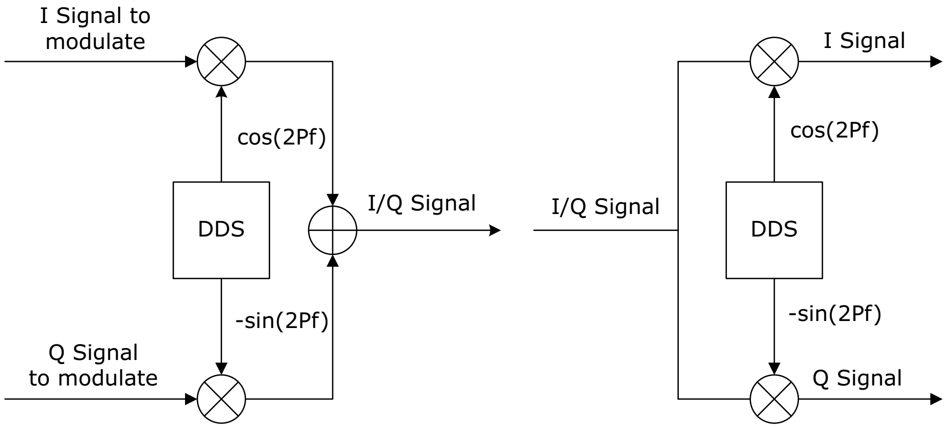
2 Overview

Microsemi direct digital synthesizer (DDS) core implements the efficient sine and/or cosine wave sample generator. The DDS or numerically controlled oscillator, the digital counterpart to analog oscillator is one of the most popular DSP functions. It is used in a broad range of communications applications including software defined radios to implement modulators/demodulators.

The DDS stores sine wave samples, in a memory based LUT and reads them out to generate series of sine and/or cosine wave digital samples. The output wave frequency depends on the sample reading speed and number of samples per wave period. You may think of the DDS functionality as storing the full wave period of samples and reading them sequentially one by one. In reality though the required LUT capacity can be too big so, the real world DDS use certain techniques to keep the storage size reasonable while not compromising the generated wave quality.

An example of a DDS-based I/Q modulator/demodulator system is shown in the following figure. On the modulator, the I channel is mixed with cosine wave, the Q channel with negative sine wave supplied by the DDS module. On the demodulator side, mixing the I/Q complex modulated signal with the cosine and sine waves restores the I and Q signals.

Figure 1 • **DDS-Based I/Q Modulator and Demodulator**



2.1 Key Features

CoreDDS supports the following features.

- Sine, cosine, or quadrature generator
- Up to 32 bits phase accumulator
- Up to 2^{20} effective LUT depth
- ¼ wave memory saving architecture
- Output sample bit resolution 4-32 bits
- Optional phase dithering for spurious-free dynamic range (SFDR) improvement
- Optional trigonometric wave correction for SFDR improvement
- User configurable output polarity
- Optional built-in dynamically controlled phase and frequency offsets
- Controllable pipeline latency

2.2 Core Version

This handbook applies to CoreDDS v3.1.

2.3 Supported Families

The following FPGA families are supported by the CoreDDS:

- PolarFire® SoC
- PolarFire®
- RTG4™
- SmartFusion®2
- IGLOO®2

2.4 Utilization and Performance

CoreDDS has been implemented in the RT4G150 device using speed grade -1 and PolarFire MPF300 at speed grade -1. A summary of the implementation data is provided in the following table.

Table 1 • **CoreDDS Device Utilization and Performance**

Parameter Settings	PH_ACC_BITS	24	24	18	24	10	12	24	21	21
	QUANTIZER_BITS	15	15	9	15	8	9	15	14	14
	OUTPUT_BITS	17	25	12	17	9	10	17	17	17
	URAM_MAXDEPTH	0	0	0	0	512	512	0	0	0
	PH_INC_MODE	0	0	0	1	0	0	1	1	1
	PH_INC	567	600	567	x	4	11	x	x	X
	SIN_ON	1	1	1	1	1	1	1	1	1
	COS_ON	1	1	1	1	0	1	0	0	0
	FREQ_OFFSET_BITS	x	x	x	10	x	x	x	x	7
	PH_OFFSET_MODE	0	0	0	1	0	0	1	1	2
	PH_OFFSET_CONST	x	x	x	x	x	x	21	21	X
	PH_CORRECTION	0	0	0	0	0	1	2	0	1
RTG4 RT4G150										
Resources Utilized	4LUT	4215	8864	585	4225	248	936	4342	2481	2528
	DFF	495	684	187	516	135	452	660	346	416
	RAM64x18	0	0	0	0	2	9	0	0	1
	RAM1K18	8	12	2	8	0	0	8	4	4
	MACC	0	0	0	0	0	0	3	0	0
Max Clock Rate, MHz		250	233	250	234	254	236	237	250	250
PolarFire MPF300										
Resources Utilized	4LUT	4011	9007	556	4048	276	728	4140	2632	2679
	DFF	495	612	187	526	139	440	654	338	382
	uSRAM	0	0	0	0	4	17	0	0	1
	RAM1K20	8	10	2	8	0	0	8	4	4
	MACC	0	0	0	0	0	0	3	0	0
Max Clock Rate, MHz		417	359	429	424	533	515	414	422	414

Note: FPGA resources and performance data for the PolarFire SoC family is similar to PolarFire family.

The common configuration settings of CoreDDS used to obtain the data of the Table 1 are listed in the following table.

Table 2 • **CoreDDS Utilization and Performance Configuration**

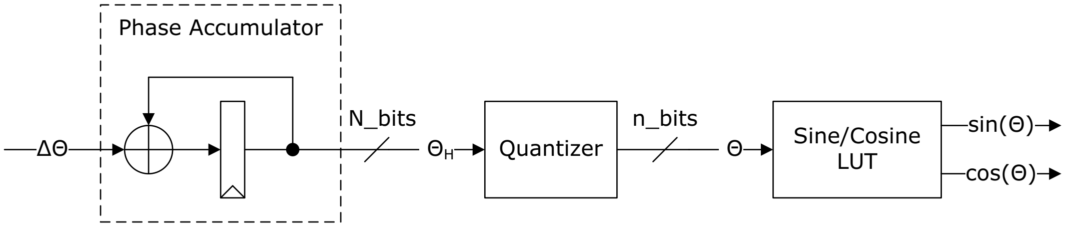
Parameter	Value
SIN_POLARITY	0
COS_POLARITY	0
LATENCY	Fully Pipelined
HDL type	Verilog

3 Functional Descriptions

3.1 Theory of Operation

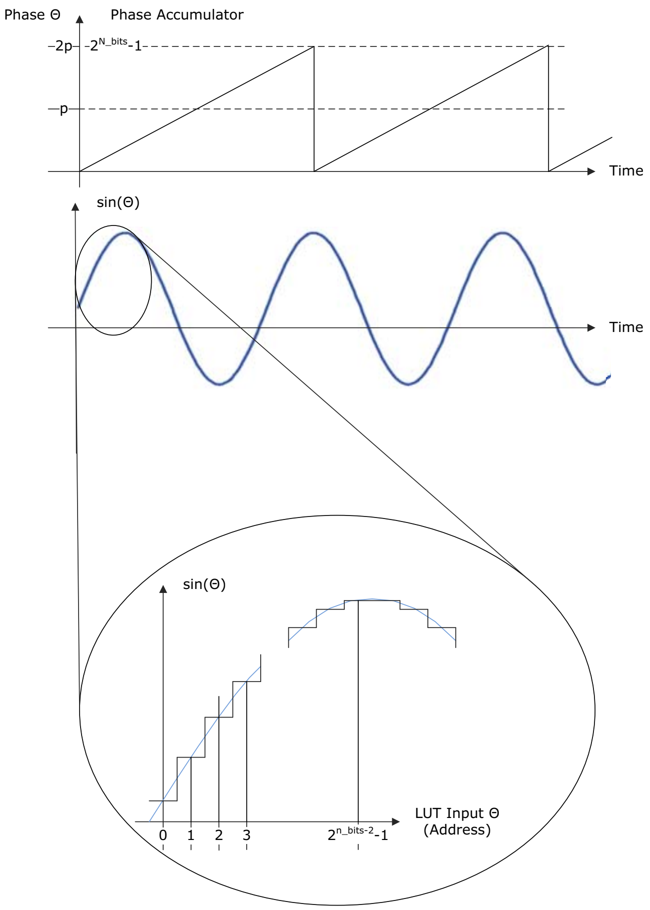
The DDS contains two major components—a phase accumulator that serves as a digital phase generator, and a phase-to-waveform converter (sine/cosine lookup table (LUT)). Practical implementations also include a phase quantizer, which helps to achieve a compromise between high resolution of the phase accumulator and limited LUT depth.

Figure 2 • **Simplified DDS Block Diagram**



The following figure shows the generation of linear phase by the phase accumulator and a LUT-based conversion of the phase into digital sine samples.

Figure 3 • **Direct Digital Synthesis Principle**



3.1.1 Output Frequency

The phase accumulator bit resolution is N_bits, so that the maximum phase is presented by the number 2^{N_bits}. At every clock period T_{clk} the phase increases by the phase increment ΔΘ. Maximum phase is reached in T_0 time interval:

$$T_0 = \frac{2^{N\_bits} \times T_{clk}}{\Delta\theta}$$

The equation can be rewritten to reflect the output frequency as the function of the phase increment, phase accumulator bit resolution and clock frequency F_{clk}:

$$F_o = \frac{\Delta\theta \times F_{clk}}{2^{N\_bits}}$$

For example, if the DDS parameters are: ΔΘ = 100, F_{clk} = 200 MHz, and N_bits = 14.

The output frequency (F_O) = 100*200 MHz / 2^{14} = 1.220703125 MHz

The phase increment value necessary to generate output frequency F_O is

$$\Delta\theta = \frac{2^{N\_bits} \times F_O}{F_{clk}}$$

The phase increment is an unsigned integer value. For example, if the DDS parameters are: F_{clk} = 100 and MHz N_bits = 18.

Then to generate the output sine wave frequency of 10 MHz, the required phase increment is

$$\Delta\theta = \frac{2^{18} \times 10 \times 10^6}{100 \times 106} = 26214.4$$

After rounding the phase increment to the closest integer number of 26214, the actual output frequency can be calculated using the Eq (2):

$$F_O = \frac{26214 \times 100 \quad \text{MHz}}{2^{18}} \approx 9.99985 \quad \text{MHz}$$

3.1.2 Frequency Resolution

The DDS frequency resolution ΔF is defined by two values: the clock frequency F_{clk} and the phase accumulator bit resolution N_bits:

$$\Delta F = \frac{F_{clk}}{2^{N\_bits}}$$

For example, if the F_{clk} = 80 MHz and N_bits = 24, the frequency resolution is

$$\Delta F = \frac{100 \times 106}{224} = 4.76837158203125 \quad \text{Hz}$$

3.1.3 Quantization Effects

The phase quantization from N_bits to n_bits (see Figure 2, page 5) intended for reducing the LUT depth, produces unwanted spurious spectral components in the generated sine/cosine wave. The limited width of the LUT that is, the bit resolution of a sine and/or cosine output samples also impacts the output wave purity. Often the output signal quality is measured as a difference between the desired sine/cosine wave and the maximum level of spurs (Lionel Cordesses, Direct Digital Synthesis: A tool for periodic wave generation. Streamlining Digital Signal Processing: A trick of the trade guidebook. Edited by Rychard G.Lyons, IEEE Press, John Wiley 2007). The difference is called spurious-free dynamic range (SFDR) and estimates as 6*n_bits dB.

CoreDDS explores a few techniques to improve the SFDR without sacrificing too much RAM capacity to implement large LUT.

The LUT implementation takes advantage of the sine wave symmetry storing only quarter period waveforms. In effect, the n_bits value for a given RAM capacity gets increased by 2. The core makes its own decision when to use a full-wave LUT contents or switch to the quarter-wave technique.

Another option is a trigonometric correction of the output wave, which approximates the sine wave using the first-order Taylor series:

$$\sin(\varphi + \Delta\varphi) \cong \sin(\varphi) + \Delta\varphi \times \cos(\varphi)$$

$$\cos(\varphi + \Delta\varphi) \cong \cos(\varphi) + \Delta\varphi \times \sin(\varphi)$$

Another method to improve SFDR aims at reducing the maximum spurs by spreading them over the available bandwidth. The method adds a small pseudo-random noise signal called dither to the phase accumulator value prior to quantizing it. The SFDR then improves by 12 dB:

$$\mathrm{SFDR} \approx 6 \times \mathrm{n\_bits} + 12 \quad \mathrm{dB}$$

3.1.4 Modulation Controls

CoreDDS supports optional inputs for the waveform frequency and/or phase modulation.

The frequency modulation varies the synthesizer output frequency. The option can be used to tune the generated waveforms to a certain frequency, like in phased-locked loop. Another application of the digitally modulated frequency is the frequency shift keying.

Phase modulation can be used to implement an initial phase shift in the generated sine wave, or phase shift keying modulation/demodulation.

4 Interface Description

4.1 Configuration Parameters

CoreDDS has parameters (Verilog) or generics (VHDL) for configuring the RTL code. These parameters and generics are described in the following table. All parameters and generics are integer types.

Table 3 • **CoreDDS Parameter Descriptions**

Parameter	Valid Range	Default	Description
PH_ACC_BITS	4 – 32	24	Bit width of the phase accumulator, N_bits
PH_INC_MODE	0, 1	0	Control over phase increment 0 – Constant phase increment defined by the parameter PH_INC value 1 – External variable phase increment intended to modulate frequency of the generated wave. The phase increment is defined by the signal value on the FREQ_OFFSET port
PH_INC	1 – $2^{PH\_ACC\_BITS-1}$ - 1	2^{20}	Constant phase increment. The value is neglected if the PH_INC_MODE differs of 0
SIN_ON	0, 1	1	Generate sine wave. 0 – do not generate the sine wave 1 – generate the sine wave At least one of the two parameters SIN_ON and COS_ON has to be set to 1
COS_ON	0, 1	1	Generate cosine wave. 0 – do not generate the cosine wave 1 – generate the cosine wave At least one of the two parameters SIN_ON and COS_ON has to be set to 1
SIN_POLARITY	0, 1	0	Sine wave output polarity. 0 – Positive 1 - Negative
COS_POLARITY	0, 1	0	Cosine wave output polarity. 0 – Positive 1 - Negative
FREQ_OFFSET_BITS	3 – (PH_ACC_BITS-1)	3	Bit width of the FREQ_OFFSET port. Neglected if the PH_INC_MODE=0
PH_OFFSET_MODE	0 – 2	0	Control over phase offset 0 – No phase offset (Phase offset = 0) 1 – Constant phase offset (Phase offset = PH_OFFSET_CONST) 2 – External variable phase offset defined by a value on the PH_OFFSET port
PH_OFFSET_CONST	1 – $2^{PH\_ACC\_BITS}$ -1	1	Constant phase offset. The value is neglected if the PH_OFFSET_MODE differs from 1
PH_OFFSET_BITS	3 – PH_ACC_BITS	3	Bit width of the PH_OFFSET port. The core neglects the value if the PH_OFFSET_MODE differs of 2

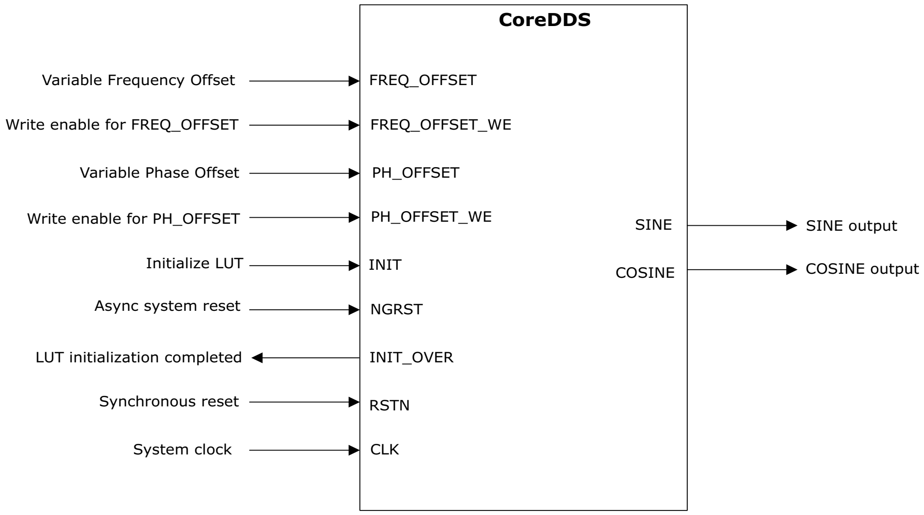
Table 3 • **CoreDDS Parameter Descriptions *(continued)***

ph_CORRECTION	0 – 2	0	Phase correction mode: 0 – None 1 – Dithering 2 - Trigonometric
QUANTIZER_BITS	3 – 20	8	Bit width of the quantizer output that defines the LUT depth. Effective LUT depth = 2QUANTIZER_BITS. The QUANTIZER_BITS value cannot exceed the PH_ACC_BITS
OUTPUT_BITS	4 – 32	18	Bit width of the output wave samples If trigonometric correction is on (PH_CORRECTION=2), the OUTPUT_BITS range is 4 to 18
LATENCY	0 – 3	3	Level of design pipelining: 0 – Minimal number of pipeline layers 1 – Small number of pipeline layers 2 – Moderate number of pipeline layers 3 – Fully pipelined design The bigger pipeline level, the bigger the latency and the better core performance is. The core UI displays a numeric latency value for every LATENCY setting expressed in number of clock cycles
URAM_MAXDEPTH	0, 4, 8, 16, 32, 64, 128, 256, 512	0	The largest RAM depth to be implemented with micro-RAM. Once the RAM depth required for a user-selected QUATIZER_BITS exceeds the URAM_MAXDEPTH, large hard RAM blocks will be used

4.2 CoreDDS Ports

The following figure shows the ports and in/out signals for the CoreDDS.

Figure 4 • **Core I/O Ports**



Note: All ports are identical for all supported families.

The following table lists the port signals and their description.

Table 4 • **CoreDDS Port Descriptions**

Port Name	Type	Port Width, Bits	Description
FREQ_OFFSET	In	FREQ_OFFSET_BITS	Unsigned external variable frequency offset influences the output frequency. It is used as an increment for the phase accumulator. The signal value becomes the phase accumulator increment when the PREQ_OFFSET_WE is active. The port signal is ignored if the PH_INC_MODE is set for the constant Phase Increment (PH_INC_MODE=0)
FREQ_OFFSET_WE	In	1	Write enable signal locks the FREQ_OFFSET value in the data register
PH_OFFSET	In	PHASE_OFFSET_BITS	Unsigned external variable phase offset. Defines the offset to the phase accumulator output consequently controlling the output waveform offset. The signal becomes the phase offset when the input PH_OFFSET_WE is active. The port signal is ignored if the PH_OFFSET_MODE is set for Zero or constant Phase Offset (PH_OFFSET_MODE!=2)
PH_OFFSET_WE	In	1	Write enable signal locks the PH_OFFSET value in the data register.
SINE	Out	OUTPUT_BITS	Sine wave output
COSINE	Out	OUTPUT_BITS	Cosine wave output
INIT	In	1	Initialize the core LUT. The optional port can be used to initialize or re-initialize the Sine/Cosine LUT at any time. During the initialization process the core output is not valid
INIT_OVER	Out	1	A clock-wide pulse indicates the LUT initialization is over. Since that moment the core outputs valid waveform samples
CLK	In	1	Clock with active rising edge
RSTN	Out	1	Optional synchronous reset signal. Active high. The signal resets the Phase Accumulator and pipeline registers
NGRST	Out	1	Asynchronous reset. Active low. The signal is expected to be active during FPGA power-on process. On the NGRST signal the core starts automatic LUT initialization, resets phase accumulator and pipeline registers

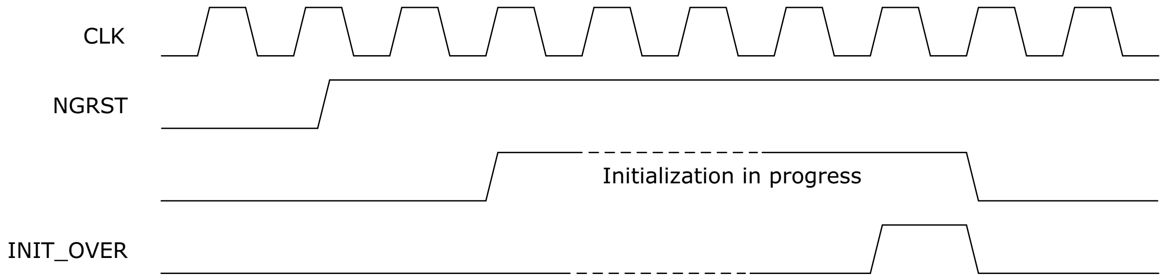
5 Timing Diagrams

5.1 Sine/Cosine LUT Initialization

Initialization, Generation, Offsets, Trigonometric phase correction

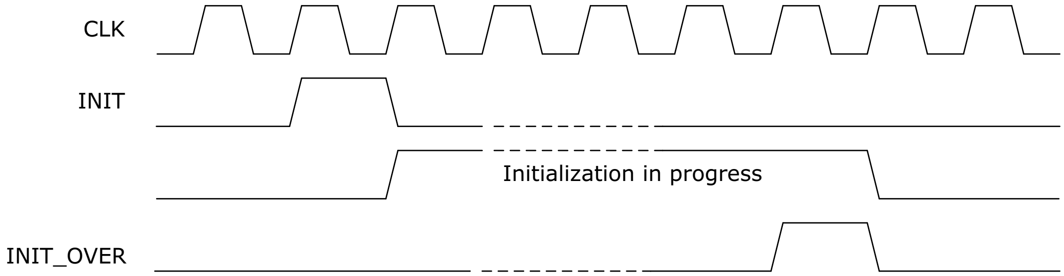
Upon powering on the FPGA, the core automatically initializes the LUT after the signal NGRS deactivates that is, goes high (Error! Reference source not found.). Once the initialization completes, the core generates the INIT_OVER flag. Since that moment the core is fully ready and starts generating the sine and/or cosine wave samples.

Figure 5 • **LUT Initialization after NGRST**



The LUT can be re-initialized at any time if desired. Since the NGRST signal is normally reserved for power on period, the port INIT can be used to start the re-initialization process (Figure 6). The INIT signal must be synchronous with regard to the CLK and last at least one clock cycle. Upon generating the INIT_OVER flag, the core starts generating valid output samples.

Figure 6 • **LUT Re-initialization after INIT Signal**

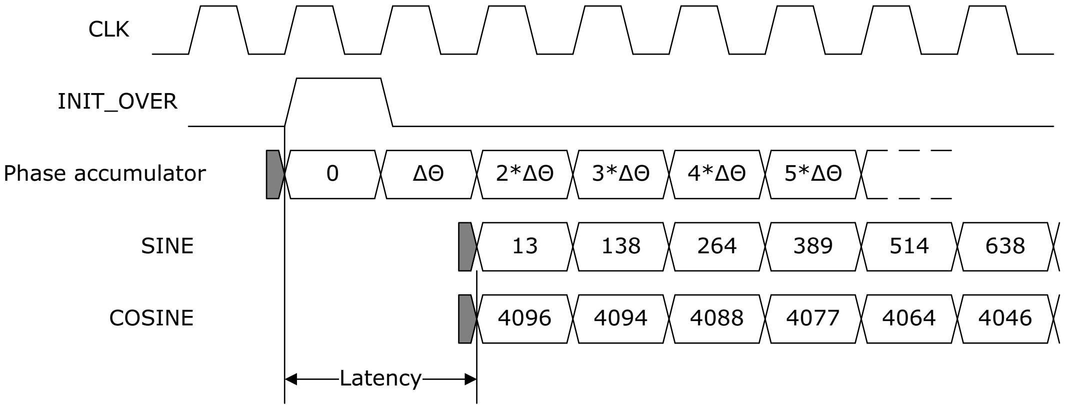


5.2 Sine/Cosine Sample Generation and RSTN

The sample generation starts automatically once the initialization is accomplished. The INIT_OVER pulse resets the phase accumulator as shown on Error! Reference source not found. Then the phase accumulator gets incremented on every positive clock edge. The output samples however appear after certain latency. You can control the latency by setting the core parameter LATENCY. The user interface indicates the numerical value of the latency expressed in clock cycles for every particular CoreDDS configuration and the LATENCY parameter setting. The example of the Figure 7 shows the latency of two clock cycles with regard to the INIT_OVER pulse.

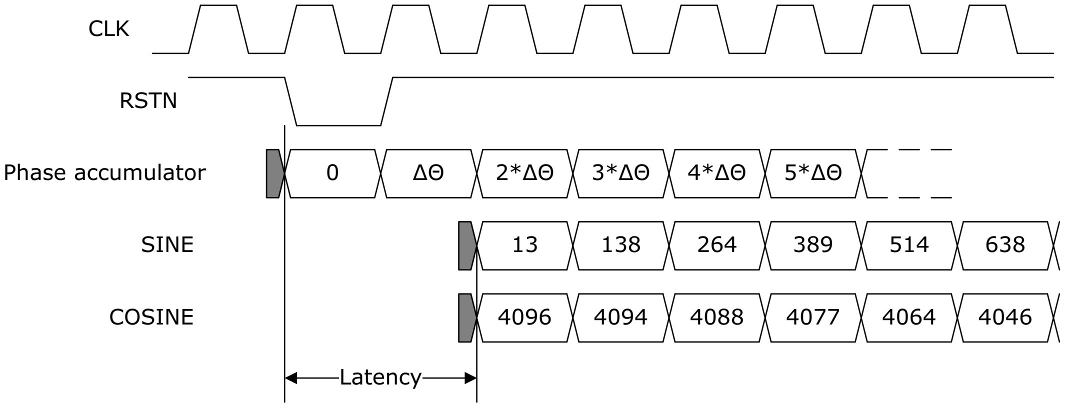
The following figure shows the case when frequency or phase modulations are off.

Figure 7 • **Waveform Sample Generation**



Reset the phase accumulator at any time after the INIT_OVER signal by issuing the RSTN signal, as shown in the following figure.

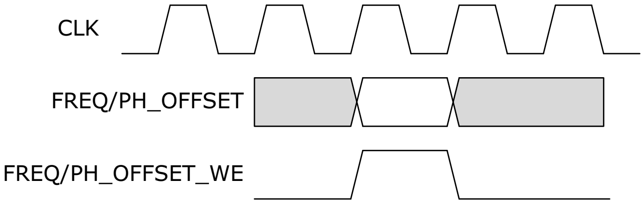
Figure 8 • **Reset Phase Accumulator**



5.3 Modulation Control

The frequency and phase offset registers lock the external offset values on the corresponding write enable signal. Immediately after the write enable pulse, the core starts using the new offset value.

Figure 9 • **Registering Optional Frequency and Phase Offsets**

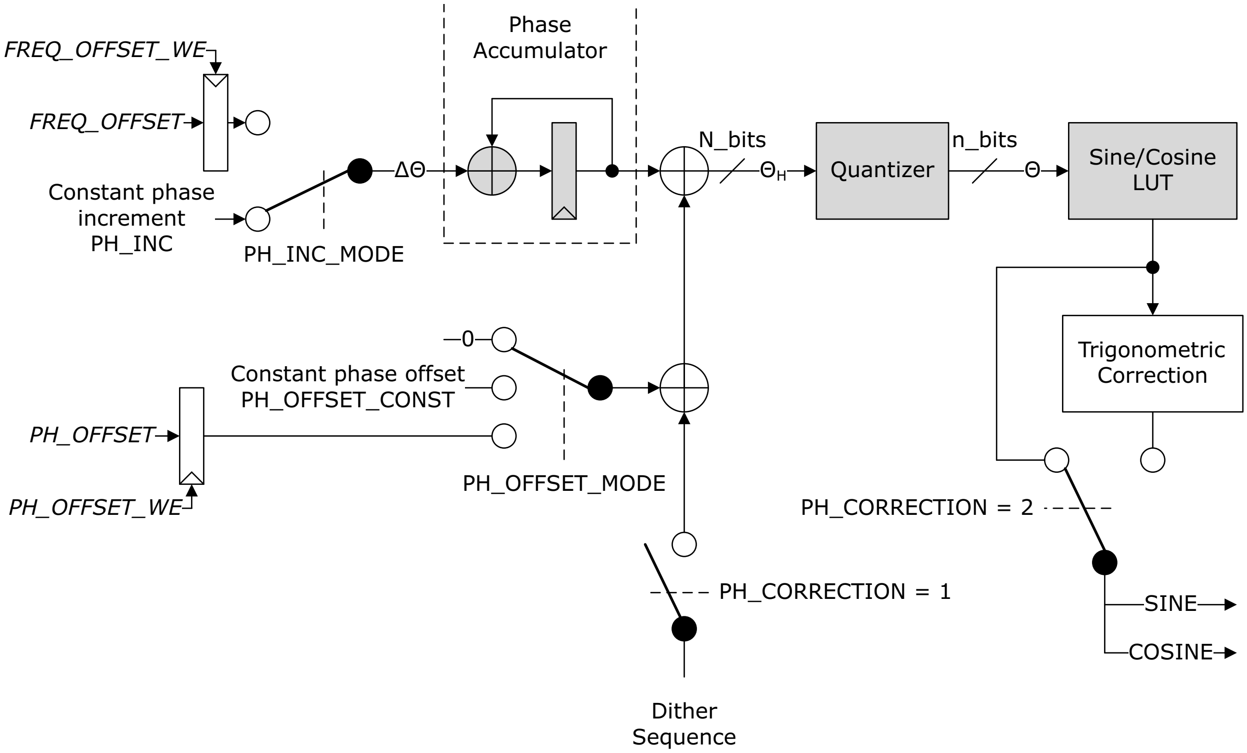


6 Implementation Details

6.1 Implementation Block Diagram

Port names are written in condensed italic, the core parameters in regular font. Optional components are shown as white rectangles. In addition to the components of the Figure 2, page 5, the implementation contains optional input registers that store frequency and/or phase offset values. The offset value gets stored in the register when it is accompanied by a corresponding write enable pulse.

Figure 10 • **Implementation Block Diagram**



The core enables the dither sequence generator if the PH_CORRECTION parameter is set to 1. When the parameter value is set to 2, the core enables trigonometric correction module.

6.2 Latency

Depending on the configuration, the core automatically infers a number of pipeline registers in critical paths to improve the design performance. You can control the number of pipelines inferred and consequently the core latency by setting the LATENCY parameter value.

Going from minimal latency through small and moderate to fully pipelined, increases the actual latency but improves the core performance. The core user interface displays the actual latency value for every configuration. The displayed latency value is expressed in number of clock cycles.

6.3 Dither Sequence

The dithering sequence module implements a 21-bit pseudo-random generator (LFSR), which generates up to four consecutive bits at every clock. When PH_CORRECTION is set to 1, the core adds the random bits to the part of the phase accumulator that eventually gets truncated by the DDS quantizer.

7 Tool Flows

7.1 Licensing

A license is not required to use this IP Core with Libero SoC.

7.2 SmartDesign

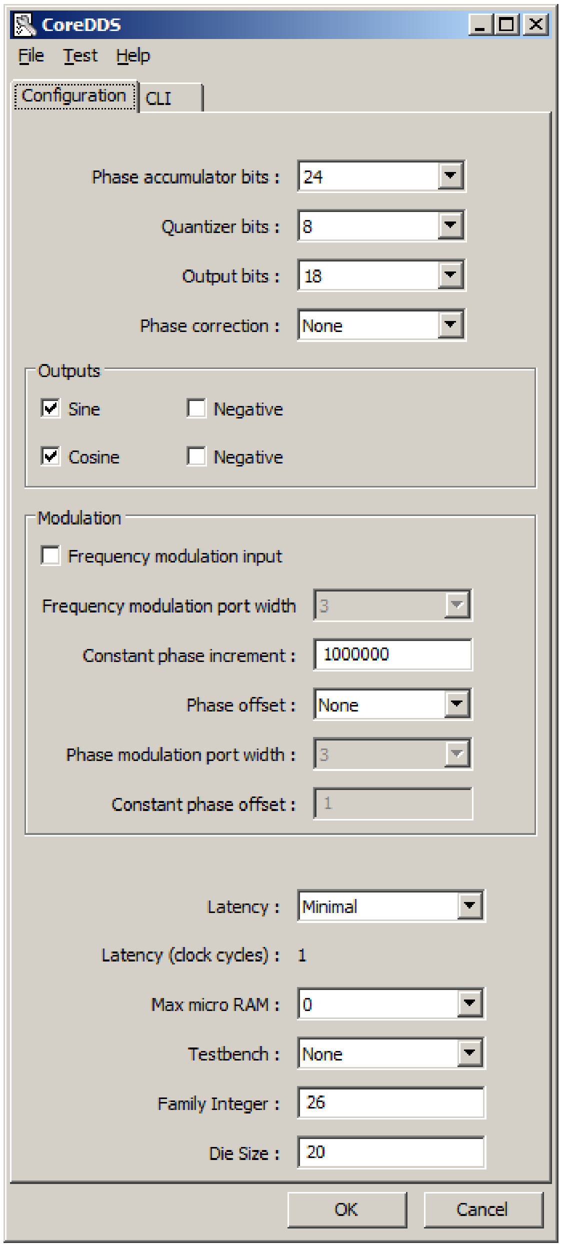
CoreDDS is available for download in the Libero SoC IP catalog through the web repository. Once it is listed in the catalog, the core can be instantiated using the SmartDesign flow. To know how to create SmartDesign project using the IP cores, refer to *Libero SoC documents page* and use the latest SmartDesign user guide.

After configuring and generating the core instance, basic functionality can be simulated using the test-bench supplied with CoreDDS. The testbench parameters automatically adjust to the CoreDDS configuration. CoreDDS can be instantiated as a component of a larger design.

7.2.1 Configuring CoreDDS in SmartDesign

The core can be configured using the configuration GUI within SmartDesign. The following figure shows an example of the GUI.

Figure 11 • **CoreDDS User Interface**



Phase accumulator bits : 24

Quantizer bits : 8

Output bits : 18

Phase correction : None

Outputs

☑ Sine ☐ Negative

☑ Cosine ☐ Negative

Modulation

☐ Frequency modulation input

Frequency modulation port width 3

Constant phase increment : 1000000

Phase offset : None

Phase modulation port width : 3

Constant phase offset : 1

Latency : Minimal

Latency (clock cycles) : 1

Max micro RAM : 0

Testbench : None

Family Integer : 26

Die Size : 20

7.3 Simulation Flows

The User Testbench for CoreDDS is included in the release.To run simulations, select the User Testbench flow within SmartDesign. The User Testbench is selected through the Core Configuration GUI.

When SmartDesign generates the core, it will install the user testbench files.

To run the user testbench, set the design root to the CoreDDS instantiation in the Libero design hierarchy pane and run Pre-Synthesis design simulation.

Note: When simulating the VHDL version of the core you might want to get rid of the IEEE.NUMERIC_STD library warnings. To do so add the following two lines to the automatically generated run.do file:

- Set NumericStdNoWarnings 1
- Set StdArithNoWarnings 1

7.4 Synthesis in Libero

To run synthesis on the CoreDDS, set the design root to the IP component instance and run the synthesis tool from the Libero design flow pane

7.5 Place-and-Route in Libero

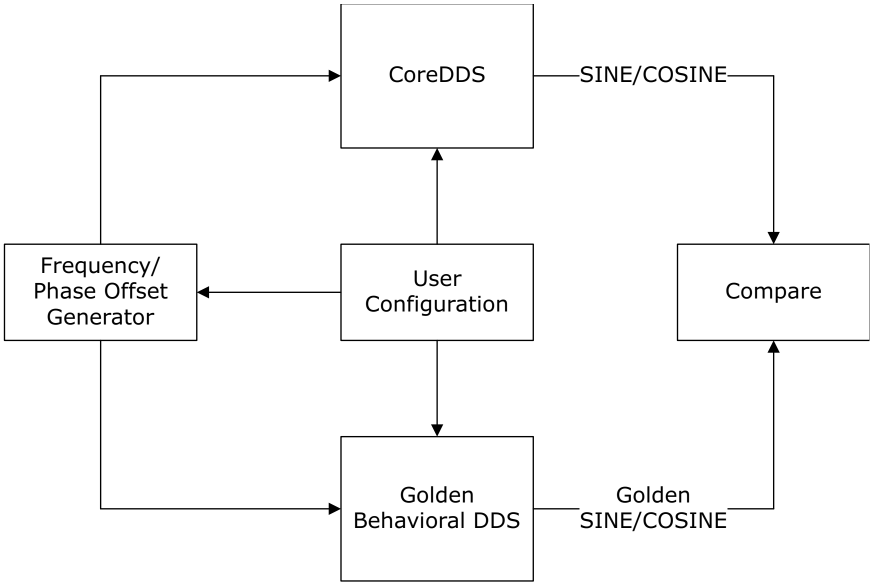
After the design has been synthesized and compiled, run Place-and-route tool. CoreDDS requires no special place-and-route settings.

8 Testbench Operation and Modification

8.1 User Testbench

The following figure shows the testbench block diagram. Both the Golden DDS and CoreDDS are configured identically and receive the same test signal. The testbench compares the output signals of the Golden module and the actual CoreDDS.

Figure 12 • **CoreDDS User Testbench**

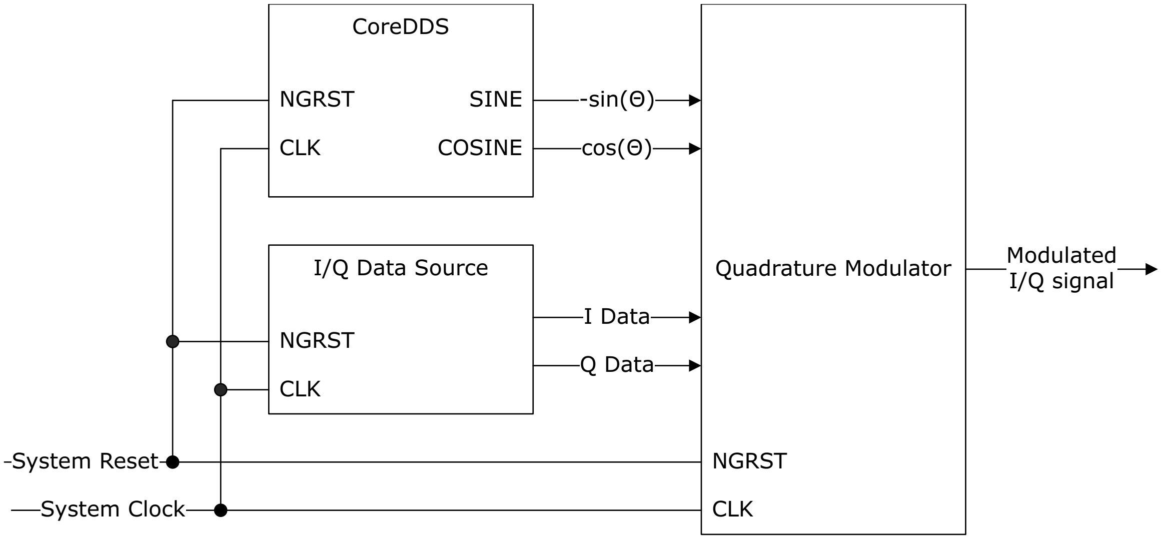


The testbench provides examples of how to use a generated DDS module. The Testbench can be modified as per the requirements.

9 System Integration

The following figure shows an example of using the core. Upon system power on and the core initialization, the DDS starts generating quadrature carrier samples indefinitely. The sine and cosine samples drive the quadrature modulator where the I and Q data get converted into a single modulated quadrature signal.

Figure 13 • **Example of a Quadrature Modulator**


CoreDDS
NGRST
SINE
CLK
COSINE
-sin(Θ)
cos(Θ)
I/Q Data Source
NGRST
CLK
I Data
Q Data
Quadrature Modulator
Modulated I/Q signal
NGRST
CLK
-System Reset
System Clock