Numerically Controlled Oscillator IP Core User's Guide

June 2010
IPUG36_02.5

Table of Contents

Numerically Controlled Oscillators (NCO), also called Direct Digital Synthesizers (DDS), offer several advantages over other types of oscillators in terms of accuracy, stability and reliability. NCOs provide a flexible architecture that enables easy programmability such as on-the-fly frequency/phase. NCOs are used in many communications systems including:

- Digital up/down converters used in 3G wireless and software radio systems
- Digital PLLs
- RADAR systems
- Drivers for optical or acoustic transmissions
- Multilevel FSK/PSK modulators/demodulators

Lattice provides a parameterizable NCO IP core that supports multiple channels and a Quadrature Amplitude Modulation (QAM) mode, in addition to other usual configurations. The resource utilization and performance trade-off can be tuned by configuring different parameters of the IP core to obtain the optimal Spurious Free Dynamic Range (SFDR) result. The Lattice NCO core offers a variety of memory reduction schemes and mechanisms for SFDR improvement.

Quick Facts

Table 1-1 through Table 1-9 give quick facts about the NCO IP core for LatticeEC™, LatticeECP™, LatticeECP2™, LattticeSC™, LatticeSCM™, LatticeXP™, LatticeECP2M™, LatticeXP2™, and Lattice ECP3™ devices.

Table 1-1. NCO IP core for LatticeEC Devices Quick Facts

		NCO IP Configuration		
		Constant FSK/PSK with 32bit phase resolution, 1 channel	**Variable FSK/PSK with 32bit phase resolution, 1 channel**	**Variable FSK/PSK with 32bit phase resolution, 16 channe**
Core Requirements	FPGA Families Supported	LatticeEC		
	Minimal Device Needed	LFEC3E	LFEC1E	LFEC6E
Resource Utilization	Targeted Device	LFEC20E-5F672C		
	LUTs	1800	300	3800
	sysMEM EBRs	3	2	5
	Registers	800	300	1900
	MULT18X18ADDSUB	N/A		
Design Tool Support	Lattice Implementation	Diamond® 1.0 or ispLEVER® 8.1		
	Synthesis	Synopsys® Synplify® Pro for Lattice D-2009.12L-1		
	Simulation	Aldec® Active-HD® 8.2 Lattice Edition II		
		Mentor Graphics® ModelSim® SE 6.3F		

Table 1-2. NCO IP core for LatticeECP Devices Quick Facts

		NCO IP Configuration		
		Constant FSK/PSK with 32bit phase resolution, 1 channel	**Variable FSK/PSK with 32bit phase resolution, 1 channel**	**Variable FSK/PSK with 32bit phase resolution, 16 channe**
Core Requirements	FPGA Families Supported	LatticeECP		
	Minimal Device Needed	LFECP6E		
Resource Utilization	Targeted Device	LFECP20E-5F672C		
	LUTs	100	300	400
	sysMEM EBRs	3	2	5
	Registers	100	300	500
	MULT18X18ADDSUB	2	0	4
Design Tool Support	Lattice Implementation	Diamond 1.0 or ispLEVER 8.1		
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1		
	Simulation	Aldec Active-HDL 8.2 Lattice Edition II		
		Mentor Graphics ModelSim SE 6.3F		

Table 1-3. NCO IP core for LatticeECP2 Devices Quick Facts

		NCO IP Configuration		
		Constant FSK/PSK with 32bit phase resolution, 1 channel	**Variable FSK/PSK with 32bit phase resolution, 1 channel**	**Variable FSK/PSK with 32bit phase resolution, 16 channe**
Core Requirements	FPGA Families Supported	LatticeECP2		
	Minimal Device Needed	LFE2-6E		
Resource Utilization	Targeted Device	LFE2-50E-7F672C		
	LUTs	100	300	300
	sysMEM EBRs	3	1	3
	Registers	100	300	500
	MULT18X18ADDSUB	2	0	4
Design Tool Support	Lattice Implementation	Diamond 1.0 or ispLEVER 8.1		
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1		
	Simulation	Aldec Active-HDL 8.2 Lattice Edition II		
		Mentor Graphics ModelSim SE 6.3F		

Table 1-4. NCO IP core for LatticeSC Devices Quick Facts

		NCO IP Configuration		
		Constant FSK/PSK with 32bit phase resolution, 1 channel	**Variable FSK/PSK with 32bit phase resolution, 1 channel**	**Variable FSK/PSK with 32bit phase resolution, 16 channe**
Core Requirements	FPGA Families Supported	LatticeSC		
	Minimal Device Needed	LFSC3GA15E		
Resource Utilization	Targeted Device	LFSC3GA25E-7F900C		
	LUTs	2200	300	5200
	sysMEM EBRs	3	1	3
	Registers	1100	300	2600
	MULT18X18ADDSUB	N/A		
Design Tool Support	Lattice Implementation	Diamond 1.0 or ispLEVER 8.1		
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1		
	Simulation	Aldec Active-HDL 8.2 Lattice Edition II		
		Mentor Graphics ModelSim SE 6.3F		

Table 1-5. NCO IP core for LatticeSCM Devices Quick Facts

		NCO IP Configuration		
		Constant FSK/PSK with 32bit phase resolution, 1 channel	**Variable FSK/PSK with 32bit phase resolution, 1 channel**	**Variable FSK/PSK with 32bit phase resolution, 16 channe**
Core Requirements	FPGA Families Supported	LatticeSCM		
	Minimal Device Needed	LFSCM3GA15EP1		
Resource Utilization	Targeted Device	LFSCM3GA25EP1-7F900C		
	LUTs	2200	300	5200
	sysMEM EBRs	3	1	3
	Registers	1100	300	2600
	MULT18X18ADDSUB	N/A		
Design Tool Support	Lattice Implementation	Diamond 1.0 or ispLEVER 8.1		
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1		
	Simulation	Aldec Active-HDL 8.2 Lattice Edition II		
		Mentor Graphics ModelSim SE 6.3F		

Table 1-6. NCO IP core for LatticeXP Devices Quick Facts

		NCO IP Configuration		
		Constant FSK/PSK with 32bit phase resolution, 1 channel	**Variable FSK/PSK with 32bit phase resolution, 1 channel**	**Variable FSK/PSK with 32bit phase resolution, 16 channe**
Core Requirements	FPGA Families Supported	LatticeXP		
	Minimal Device Needed	LFXP3E		
Resource Utilization	Targeted Device	LFXP20E-5F484C		
	LUTs	1800	300	3800
	sysMEM EBRs	3	2	5
	Registers	800	300	1900
	MULT18X18ADDSUB	N/A		
Design Tool Support	Lattice Implementation	Diamond 1.0 or ispLEVER 8.1		
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1		
	Simulation	Aldec Active-HDL 8.2 Lattice Edition II		
		Mentor Graphics ModelSim SE 6.3F		

Table 1-7. NCO IP core for LatticeECP2M Devices Quick Facts

		NCO IP Configuration		
		Constant FSK/PSK with 32bit phase resolution, 1 channel	**Variable FSK/PSK with 32bit phase resolution, 1 channel**	**Variable FSK/PSK with 32bit phase resolution, 16 channe**
Core Requirements	FPGA Families Supported	LatticeECP2M		
	Minimal Device Needed	LFE2M20E		
Resource Utilization	Targeted Device	LFE2M-35E-7F484C		
	LUTs	100	300	300
	sysMEM EBRs	3	1	3
	Registers	100	300	500
	MULT18X18ADDSUB	2	0	4
Design Tool Support	Lattice Implementation	Diamond 1.0 or ispLEVER 8.1		
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1		
	Simulation	Aldec Active-HDL 8.2 Lattice Edition II		
		Mentor Graphics ModelSim SE 6.3F		

Table 1-8. NCO IP core for LatticeXP2 Devices Quick Facts

		NCO IP Configuration		
		Constant FSK/PSK with 32bit phase resolution, 1 channel	**Variable FSK/PSK with 32bit phase resolution, 1 channel**	**Variable FSK/PSK with 32bit phase resolution, 16 channe**
Core Requirements	FPGA Families Supported	LatticeXP2		
	Minimal Device Needed	LFXP2-5E		
Resource Utilization	Targeted Device	LFXP2-17E-7F484CES		
	LUTs	100	300	300
	sysMEM EBRs	3	1	3
	Registers	100	300	500
	MULT18X18ADDSUB	2	0	4
Design Tool Support	Lattice Implementation	Diamond 1.0 or ispLEVER 8.1		
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1		
	Simulation	Aldec Active-HDL 8.2 Lattice Edition II		
		Mentor Graphics ModelSim SE 6.3F		

Table 1-9. NCO IP core for LatticeECP3 Devices Quick Facts

		NCO IP Configuration		
		Constant FSK/PSK with 32bit phase resolution, 1 channel	**Variable FSK/PSK with 32bit phase resolution, 1 channel**	**Variable FSK/PSK with 32bit phase resolution, 16 channe**
Core Requirements	FPGA Families Supported	LatticeECP3		
	Minimal Device Needed	LFE3-35EA		
Resource Utilization	Targeted Device	LFE3-95E-7FN672CES		
	LUTs	100	300	400
	sysMEM EBRs	3	1	3
	Registers	100	300	500
	MULT18X18C	4	0	8
Design Tool Support	Lattice Implementation	Diamond 1.0 or ispLEVER 8.1		
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1		
	Simulation	Aldec Active-HDL 8.2 Lattice Edition II		
		Mentor Graphics ModelSim SE 6.3F		

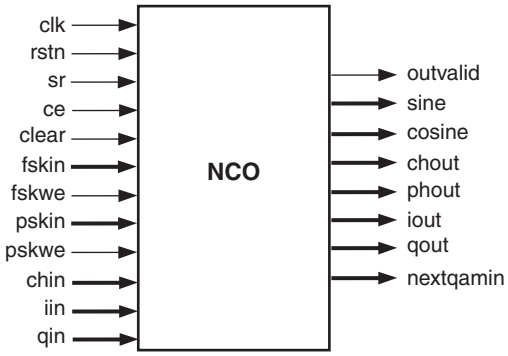
Features

- Supports single or multi channel operation up to 16 channels
- Run time variable phase increment input $\Delta\theta$ and phase offset input ϕ
- Up to 32-bit user-configurable phase resolution
- Up to 20-bit user-configurable quantizer resolution

- Up to 32-bit user-configurable amplitude resolution
- User-configurable memory saving architectures – 1/2 wave, 1/4 wave or full wave
- User-selectable sum of angles (SOA) optimization for memory saving
- Up to 4-bit user-selectable phase dithering correction
- User-selectable trigonometric correction for SFDR improvement
- Option for truncating or rounding the quantizer output when neither dithering nor trigonometric correction is used.
- User-selectable QAM mode support
- Provides high-SFDR up to 115 dB
- Provides sine, cosine or quadrature outputs.
- User configurable output polarity

This chapter provides a functional description of the NCO IP core. Figure 2-1 shows a top-level inteface diagram for the NCO IP core.

Figure 2-1. Top-level Interface Diagram for NCO IP Core



Principle of NCO

The NCO generates a sine waveform using the concept of direct digital synthesis. In direct digital synthesis, the samples of the sine wave are stored in memory and are read out to generate the output sine wave. The frequency of the output sine wave is controlled by the clock speed and appropriate skipping of intermediate data points. In the simplest scenario, the sampled data for one full wave period is stored in memory and is directly used for the output. However, other enhanced methods are frequently used to reduce the memory size requirements. For example, only a half or a quarter cycle of the waveform could be stored in memory and the memory address and output sign could be manipulated to get the full cycle waveform. Another useful technique for memory reduction is to consider the input angle as the sum of a coarse angle and a fine angle and compute the output from coarse and fine look-up tables using the sum of angles trigonometric identity.

The simplest full wave NCO is considered first to explain the concepts and bring out the notations. The full wave corresponding to one period of the sine wave is divided into N segments. The incremental angle for each segment, denoted as $\Delta\theta$, is equal to $2\pi/N$ and the phase values corresponding to one period are given by:

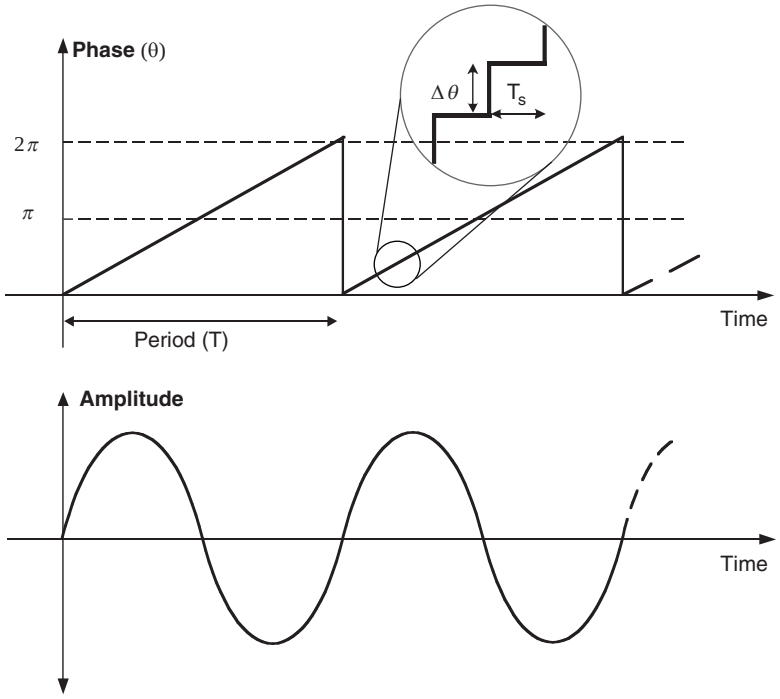
$$\theta_i = \frac{2\pi i}{N} \quad i = 0, 1, 2, ..., N-1 \tag{1}$$

The output values corresponding to the phase sequence, given in Equation 2, are stored in the look-up table. Figure 2 shows the mapping of angle to sine waveform.

$$d_i = \sin\theta_i = \sin\left[\frac{2\pi i}{N}\right] \tag{2}$$

The phase index, i is generated either sequentially or in increments and used to address the memory look-up table. The output of the look-up table is the sine wave sample. The index increment can be any value greater than zero, including fractional values.

Figure 2-2. Mapping of Linear Phase Angle to Sine Wave

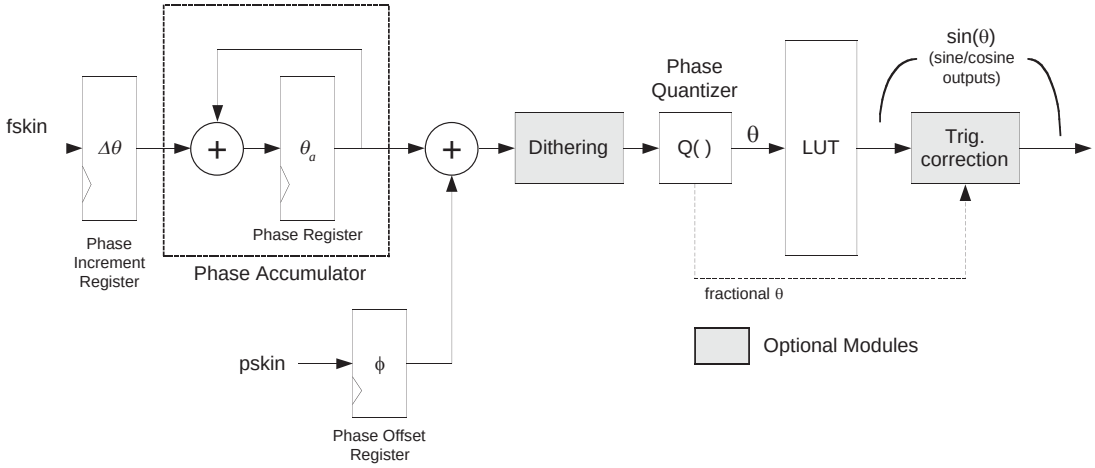


Lattice NCO Implementation

The key elements of a simple NCO are the phase accumulator and waveform look-up table. The phase accumulator adds a constant phase increment stored in the phase increment register to the accumulated phase at every clock cycle. The accumulated phase provides addresses for the look-up table. The accumulated phase is usually quantized before addressing the look-up table to allow for fractional phase index increments.

The Lattice NCO implementation is shown in Figure 2-3. This figure shows a single channel NCO, with FSK (Frequency Shift Keying) and PSK (Phase Shift Keying) inputs and a full wave look-up table. It also shows optional modules for dithering and trigonometric correction. The functional blocks of NCO IP are described in the following sections.

Figure 2-3. Lattice NCO Implementation



Phase Increment Register (for FSK)
The phase increment register stores the phase value ($\Delta\theta$) that gets added up to the accumulated phase at every clock cycle. The phase increment linearly decides the frequency of the output signal. Hence, this input can be used for frequency shift keying (FSK) modulation. The phase increment is either fixed or read dynamically from an input port, `fskin`, depending on how the NCO is configured. The output frequency is always a fraction of the clock frequency of the system.

Phase Accumulator
The phase accumulator computes the phase angle value that is used to address the look-up tables used for the output sine signal generation. The phase angle at any cycle is equal to the phase angle at the last cycle plus the phase increment. For cycle i, $\theta_i = \theta_{i-1} + \Delta\theta$. The width of the accumulator is specified by the user parameter, "Phase resolution". For a given accumulator width, phase resolution is highest when the phase increment is equal to 1 and reduces for values greater than 1.

Phase Shift Keying
A constant phase input is added to the accumulated phase before addressing the look-up table. This is useful for implementing phase shift keying (PSK) modulation of the NCO output. The user can choose no phase offset, a fixed phase offset or a variable phase offset (PSK). The variable offset is applied through the PSK input (`pskin`). Any phase offset that is added causes a shift in the phase angle and a corresponding linear phase shift in the output sine signal.

Quantizer
The output of phase accumulator (or the optional PSK or dithering module) drives the quantizer. The quantizer scales down the accumulator output to reduce the size of the look-up table. Assuming the look-up able has integer resolution, the quantizer provides a mechanism for fractional phase increments. The Quantizer output width decides the depth of the look-up table and is normally less than the accumulator output width. This allows high precision accumulation operation while using less memory.

Look-up Table
The central part of the NCO is the look-up table which stores the values of the sine wave corresponding to equally spaced phase angles in the (0,2π) interval. If the `Wave size` parameter is equal to "half" or "quarter", sine wave samples corresponding to (0,π) or (0, π/2) respectively are stored in the look-up table. As the cosine of an angle can be derived from the sine of a shifted angle, the cosine value, if required, is read from the same look-up table by manipulating the address. The depth of the look-up table is always a power of 2 and is determined by the user-defined parameter `Quantizer resolution`. The width of the look-up table is, in most cases, equal to the output width. The look-up table is implemented using block or distributed memories, which is selected by the user parameter `Memory type`. The memory is addressed by the phase angle index, which is generated by the accumulator and quantizer blocks.

Half-wave storage reduces the memory requirement by half, but uses slightly more logic and increases the latency by one cycle. Except for very small look-up table configurations, the user may better choose half-wave storage to reduce memory usage. The user can also choose a quarter-wave storage to reduce memory by another half (half of what is needed for half-wave storage). In the quarter-wave case, however, the latency increases by 1 cycle and additional logic is used compared to half-wave implementation.

Sum-of-Angles Memory Reduction

As the sine wave samples are stored in memory in direct digital synthesis NCOs, increasing the phase resolution of the output leads to corresponding increase in the size of the look-up table. The amount of memory required can be greatly reduced by making use of the "sum of angles" trigonometric identity and by using additional multipliers and adders after the memory output. This is achieved by dividing the angle space into coarse sub-divisions and then writing the phase angle as a sum of the nearest coarse angle and an additive corrective angle (fine angle).

Consider Equation 1 that relates phase angle to an integer angle index. The phase angle resolution of *N* that is used in that equation can be achieved by the following method. Define the following two sets of angles: coarse and fine, by choosing *C* and *F* to satisfy the equation *C*F=N.*

Coarse angle set:

$$c_j = \frac{2\pi j}{C} \quad j = 0, 1, 2, ..., C-1 \tag{3}$$

Fine angle set:

$$f_k = \frac{2\pi k}{CF} \quad k = 0, 1, 2, ..., F-1 \tag{4}$$

Any phase angle θ_i of Equation 1 can be written as a sum of an angle in the coarse set and one in the fine set as:

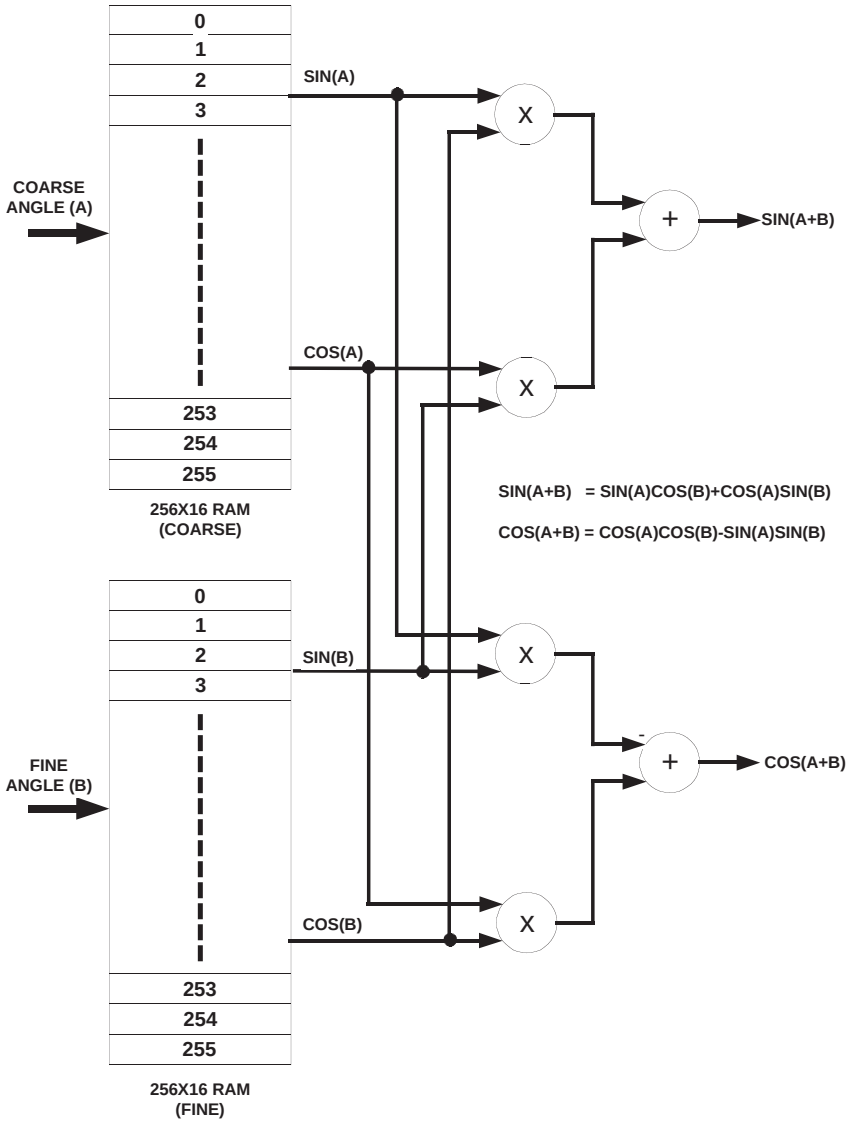
$$\theta_i = c_x + f_y \quad \text{where } x = i \text{ div } F \text{ and } y = i \bmod F. \tag{5}$$

The sine and cosine values of θ_i can be computed using the sine and cosine values of c_x and f_y using the following trigonometric identity:

$$\sin(\theta_i) = \sin(c_x + f_y) = \sin(c_x)*\cos(f_y) + \cos(c_x)*\sin(f_y)$$
$$\cos(\theta_i) = \cos(c_x + f_y) = \cos(c_x)*\cos(f_y) - \sin(c_x)*\sin(f_y)$$

The look-up tables need only to store the sine and cosine values for coarse and fine phase angle sets only. An implementation of the sum of angles scheme is shown in Figure 2-4.

Figure 2-4. Sum of Angles Memory Reduction



This sum of angle scheme uses four multipliers and two adders after the look-up table. However, the memory used is much less compared with the full-wave scheme without sum of angles reduction. For a typical example of 16-bit quantizer resolution, sum of angles scheme can lead to more than 98% memory saving, compared to the full-wave implementation.

Improving Quality of Output

A common measure of the output quality of NCO is the Spurious Free Dynamic Range (SFDR). This roughly indicates the degree of power separation between the main lobe and the next strongest side lobe in the power spectral density plot. The SFDR can be improved using either phase dithering or trigonometric correction. Phase dithering diffuses the concentration of phase quantization noise by adding a small random value to the accumulated phase before quantization. Trigonometric correction serves to improve the SFDR in a more deterministic way by adding a correction factor computed from the discarded LSB bits, to the output. The SFDR for the NCO output without dithering or trigonometric correction is approximately equal to 6*`Quantizer resolution`.

Phase Dithering

Truncation in phase address output from accumulator, results in amplitude errors at the output of the sine or cosine waveforms. These errors are periodic in nature regardless of the frequency of operation. Due to the periodicity of these errors in time, they appear as spurious frequencies in the frequency spectrum. This reduces the SFDR of the output signal. In order to improve the SFDR, random phase error is introduced in the least significant bits of the look up table address. Introduction of this randomness minimizes the periodicity of the errors in time domain, resulting in reduced strength of the spurious frequencies in frequency spectrum. This SFDR improvement is achieved at the cost of reduced signal-to-noise ratio (SNR) at the output. Phase dithering is implemented by adding a random number to the phase address output of the accumulator before it is given to the quantizer. The word length of the random number is user programmable based on the parameter `Dithering bits`.

Trigonometric Correction

If the SFDR requirements are more stringent and cannot be met by the phase dithering option, then trigonometric phase correction implementation should be used. This implementation improves SFDR by 46 dB over the no-phase-correction implementation and by 34 dB over the phase dithering implementation. The phase correction is implemented on the output samples from the look-up table memories as shown in Figure 2-5. In this implementation the truncated LSBs of the phase accumulator are used for phase correction using trigonometric properties as explained below:

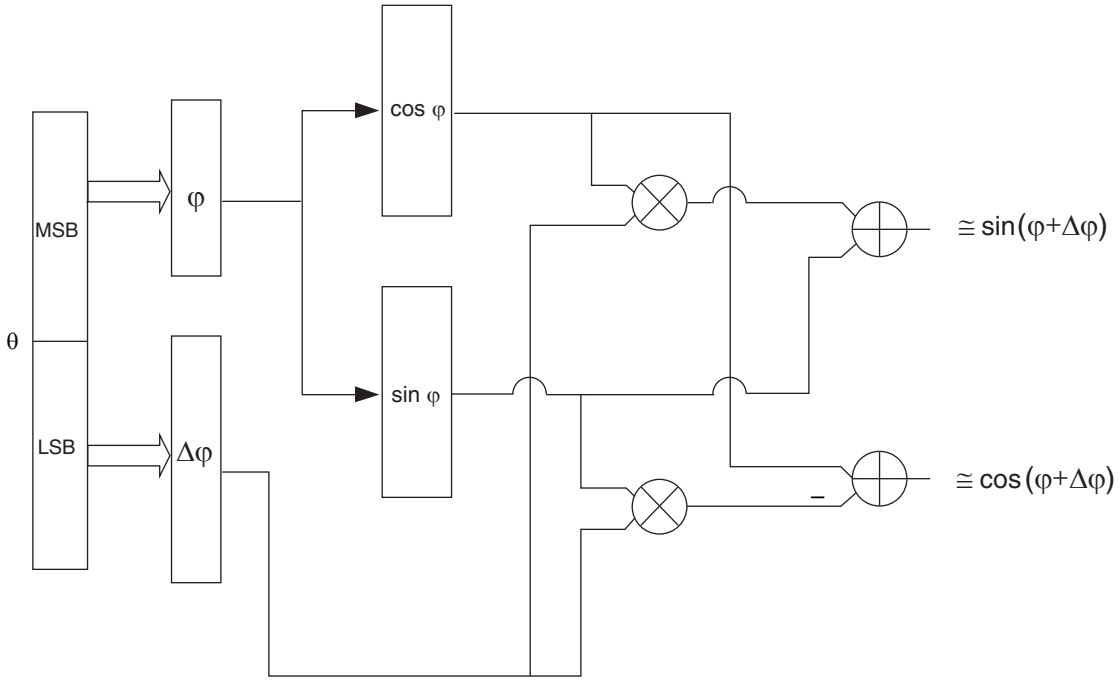
For any φ and a $\Delta\varphi < \pi/2$,

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

$$\cos(\varphi + \Delta\varphi) \cong \cos(\varphi) - \Delta\varphi * \sin(\varphi)$$

This implementation requires two additional multipliers and two adders and one constant multiplier as shown in the figure.

Figure 2-5. Trigonometric Phase Correction



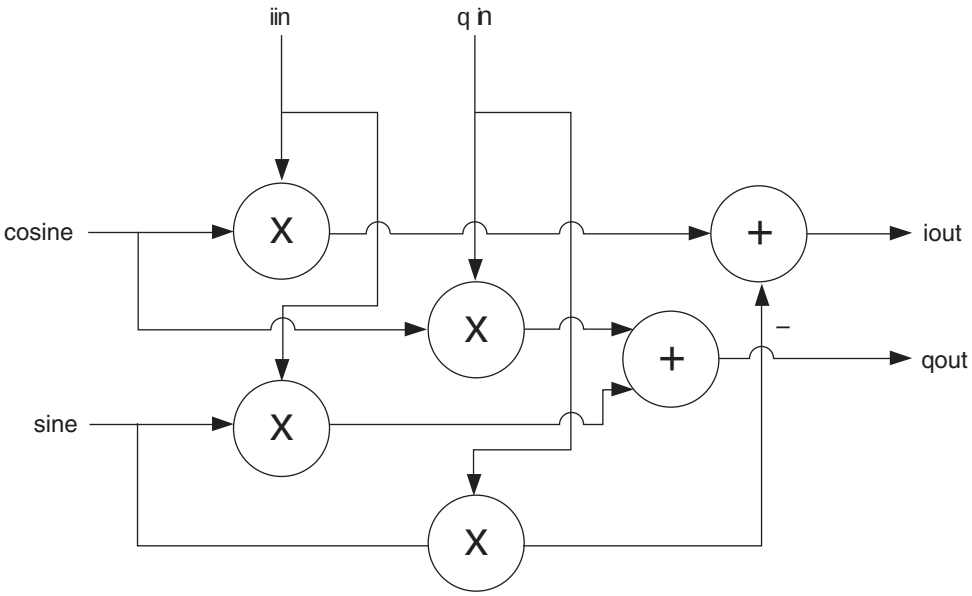
Multi-channel NCO

In multi-channel operation, the NCO can generate outputs for up to 16 channels. All these outputs will timeshare the output ports. The multi-channel implementation of the NCO contains all the functional blocks required by the single channel NCO. In addition to those blocks, it uses memories for storing the phase increment values, phase offset values and phase accumulator output values for each channel.

Quadrature Amplitude Modulation (QAM)

In addition to frequency and phase modulations, this NCO can also be used for quadrature amplitude modulation (QAM). These modulations are ubiquitous in wireless and wireline communications systems. Four multipliers and two adders are required for implementing QAM. The QAM implementation is shown in Figure 2-6.

Figure 2-6. QAM Implementation



Signal Descriptions

Table 2-1. Interface Signal Descriptions

Port	Bits	I/O	Description
All Configurations			
clk	1	I	System clock (reference clock for input and output data).
rstn	1	I	System wide asynchronous active low reset signal.
sine	4 - 32	O	Sine output data in 2's complement form. This port can be optionally omitted if either cosine port or QAM ports (iout and qout) are selected.
cosine	4 - 32	O	Cosine output data in 2's complement form. This port can be optionally omitted if either sine port or QAM ports (iout and qout) are selected.
A For FSK Mode only (when the parameter FSK input = "Variable")			
fskin	3 - 31	I	Frequency shift keying input data. This unsigned value becomes the phase increment factor for the phase accumulator and decides the output frequency. The value at this port is read only when fskwe is high.
fskwe	1	I	Write enable strobe for fskin data.
For PSK Mode only (when the parameter PSK input = "Variable")			
pskin	3 - 32	I	Phase shift keying input data. This unsigned value is used as offset to accumulated phase and is normally used to implement phase shift keying modulation. The value at this port is read only when pskwe is high.
pskwe	1	I	Write enable strobe for pskin data.
For Multi-channel Mode only (when the parameter Multi channel = "Yes")			
chin	1 - 4	I	This port is used when the number of channels is more than one and either or both PSK input and FSK input parameters are configured as "Variable". The value in chin port associates the channel number for the current fskin or pskin ports. The width of this port depends upon the number of channels and is equal to the next higher integer value of $\log_2$ of (Number of Channels).
chout	1 - 4	O	This output is present if the NCO operates in multi-channel mode. The value at this port indicates the channel number for which data samples are given at the output currently. The width of this port depends upon the number of channels and is equal to the next higher integer value of $\log_2$ of (Number of Channels).
For QAM Mode only (when the parameter QAM Mode = "Yes")			
iin	4 - 18	I	I input for Quadrature Amplitude Modulation. The width is defined by the parameter QAM input port width.
qin	4 - 18	I	Q input for Quadrature Amplitude Modulation. The width is defined by the parameter QAM input port width.
iout	4 - 32	O	I component of the QAM output. is equal to Output width plus QAM input port width.
qout	4 - 32	O	Q component of the QAM output. The width of this port is equal to Output width plus QAM input port width. The qout port is a user-selectable output.
nextqamin	1-4	O	This output port gives the channel number for the next QAM input signal (iin or qin). This optional signal is available only when Multi-channel is selected.
Optional I/Os			
ce	1	I	Clock enable signal. This signal has the highest priority after rstn. The NCO operation freezes for as long as ce is held low. This optional signal should be selected only when required as it leads to increased core size.
sr	1	I	Synchronous reset signal. When asserted all internal registers are reset. The optional signal ce, if used, must be held high, for sr to be effective. This optional signal should be selected only when required as it leads to increased core size.
clear	1	I	Accumulator clear signal. If high, it clears the phase accumulator and restarts the sine output from zero or the programmed phase offset (PSK offset).

Table 2-1. Interface Signal Descriptions (Continued)

Port	Bits	I/O	Description
phout	3-32	O	Phase output. This optional output provides the phase value corresponding to the current sine or cosine output (in unsigned format).
outvalid	1	O	Output valid. This optional output signal signifies the presence of a valid output at the output data busses (sine and/or cosine).

Latency

The latency for NCO varies depending on different parameter settings. It is displayed in the summary page of the NCO IP GUI. Latency for NCO is defined as the number of clock cycles required for changes to either fskin or pskin to be reflected in sine or cosine outputs. When both FSK input and PSK input are defined as "constant", then the latency is the number of clock cycles for valid outputs (sine or cosine) to appear after rstn is de-asserted. For the sample configuration in the timing diagram Figure 2-7, the latency is three cycles.

Timing Diagrams

The I/O timing diagrams for single channel and multi-channel NCOs are given in Figure 2-7 and Figure 2-8 respectively.

Figure 2-7. Timing Diagram for Single Channel NCO

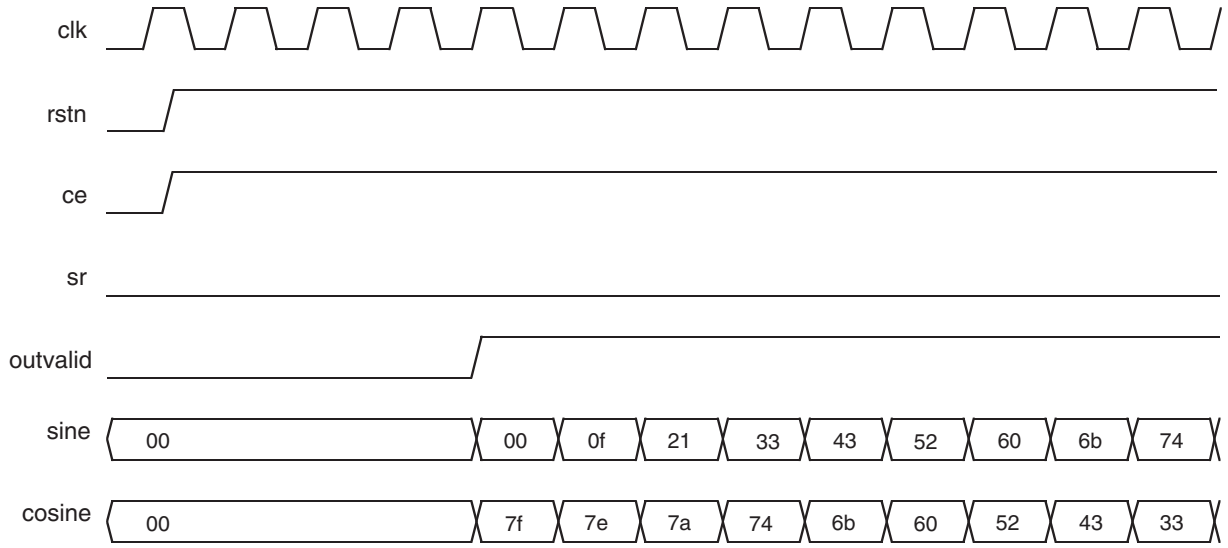
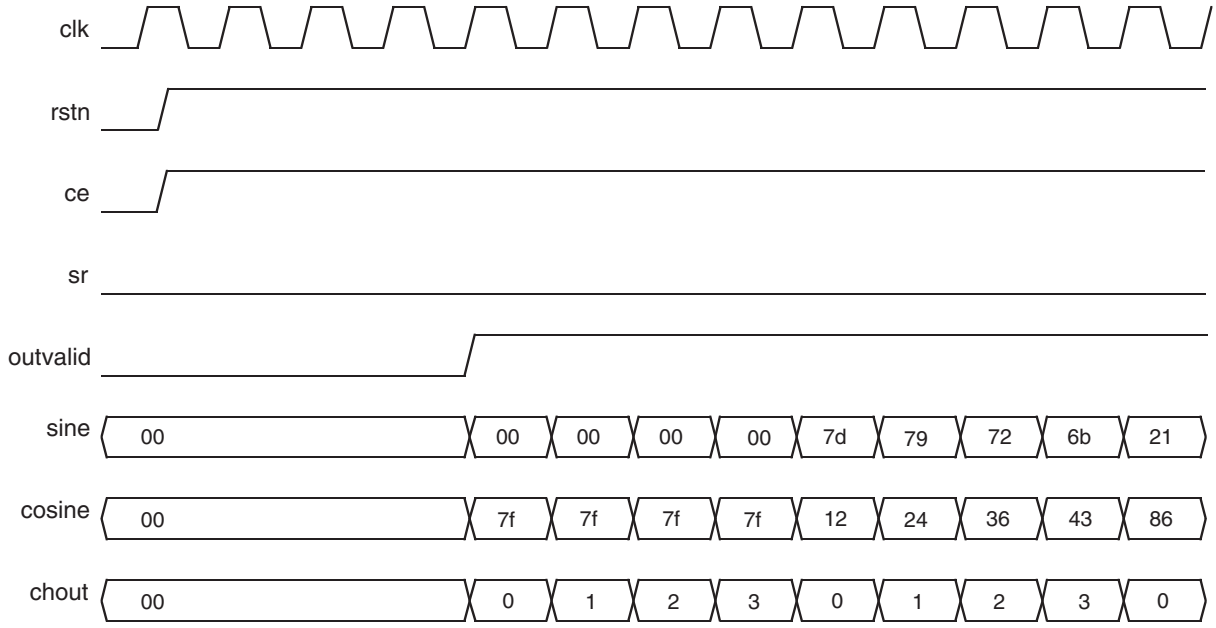


Figure 2-8. Timing Diagram for Multi-Channel NCO



Parameter Settings

The IPexpress™ tool is used to create IP and architectural modules in the Diamond and ispLEVER software. Refer to "IP Core Generation" on page 27 for a description on how to generate the IP.

Table 3-1 provides the list of user configurable parameters for the NCO IP core. The parameter settings are specified using the NCO IP core Configuration GUI in IPexpress. The numerous NCO parameter options are partitioned across multiple GUI tabs as shown in this chapter.

Table 3-1. NCO IP Core Configuration Parameters

Parameter	Range/Options	Default Value
Multi-channel Mode		
Multi-channel	Yes, No	No
Number of channels	2 to 16	16
Wave Characteristics		
Wave size	full, half or quarter	full
Sum of angles	Yes, No	Yes
Phase resolution	3 to 32 (for non Trigonometric correction); 32 (for Trigonometric correction).	32
Quantizer resolution	3 to 16 if sum of angles is not used; 6 to 20, if sum of angles is used; 12, if Trig. correction is used. The maximum is limited by `Phase resolution` in all cases.	16
Output width	18, if Trigonometric correction; 4 to 18, if sum of angles or QAM mode; otherwise 4 to 32.	18
Phase Correction		
Phase correction	None, Dithering, Trigonometric	None
Rounding type	Truncation, Nearest	Truncation
Dithering bits	1 to 4	4
QAM Mode		
QAM mode	Yes, No	No
`QAM input port width`	4 to 18	16
FSK Mode		
FSK input	Constant, Variable	Constant
Phase increment	1 to 2^(`Phase resolution`-1)	1073741824
FSK input port width	3 to (`Phase resolution` -1)	16
PSK Mode		
PSK input	None, Constant, Variable}	None
Phase offset	1 to 2^(`Phase resolution`)	1
PSK input port width	3 to `Phase resolution`	16
Memory Type		
Memory type	Block memory, Distributed memory	Block memory
DSP Block		
Use DSP block	Yes, No	Yes

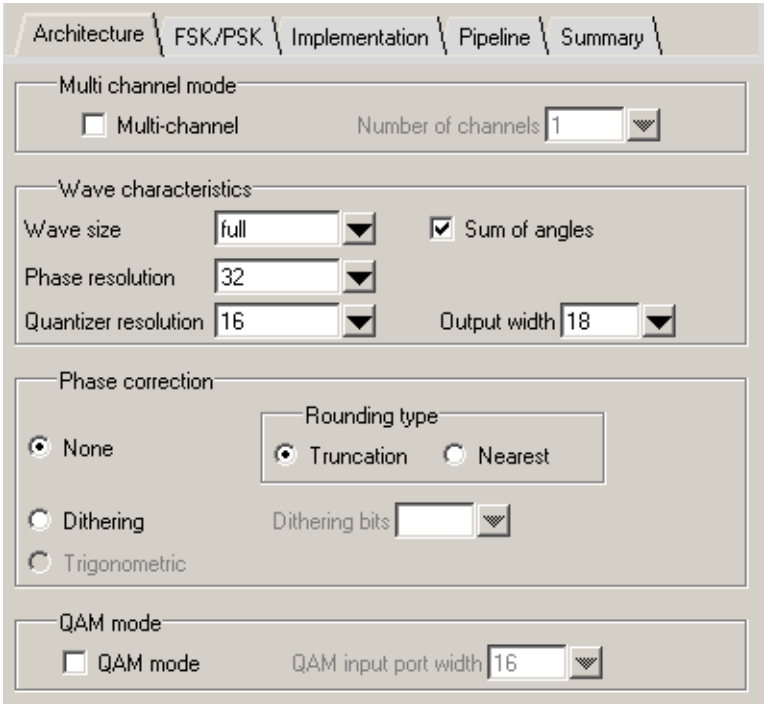
Table 3-1. NCO IP Core Configuration Parameters (Continued)

Parameter	Range/Options	Default Value
Data Output Ports		
Sine	Yes, No	Yes
Cosine	Yes, No	Yes
Sine Polarity	{Positive, Negative}	Positive
Cosine Polarity	{Positive, Negative}	Positive
Optional I/O Ports		
ce	Yes, No	No
sr	Yes, No	No
clear	Yes, No	No
phout	Yes, No	No
outvalid	Yes, No	Yes
qout	Yes, No	No
Pipeline Options		
Register after phase shift adder	Yes, No	No
Register after phase dithering block	Yes, No	No
Register after phase quantizer	Yes, No	No
Memory output register	Yes, No	Yes
Additional memory data register for half and quarter waves	Yes, No	No

Architecture Tab

Figure 3-1 shows the contents of the Architecture tab.

Figure 3-1. Architecture Tab



Multi-channel Mode

Multi-channel
Determines whether multiple channels are supported.

Number of Channels
Denotes the number of NCO channels. Valid only if Multi-channel is selected.

Wave Characteristics

Wave Size
Determines how much of the sine wave is stored in the look-up table.

Sum of Angles
Determines whether sum of angles method is used for memory reduction.

Phase Resolution
Phase Resolution: Maximum phase resolution of the NCO expressed in bits. This also defines the accumulator width.

Quantizer Resolution
Phase quantizer resolution: The output of the phase accumulator is quantized to this resolution before addressing the trigonometric look-up table. This also determines the depth of the trigonometric look-up table. The maximum value supported is 16 bits without sum of angles usage and 20 bits if the sum of angles method is employed. This resolution must be less than or equal to Phase resolution.

Phase Correction

Phase Correction
Phase correction method for SFDR improvement. “Trigonometric” option is not available if Sum of angles is selected.

Rounding Type
Rounding type used for quantizing the phase accumulator output. This is valid only if Phase correction is “None.”

Dithering Bits
Number of dithering bits. This is used only if Phase correction is “Dithering.”

QAM Mode

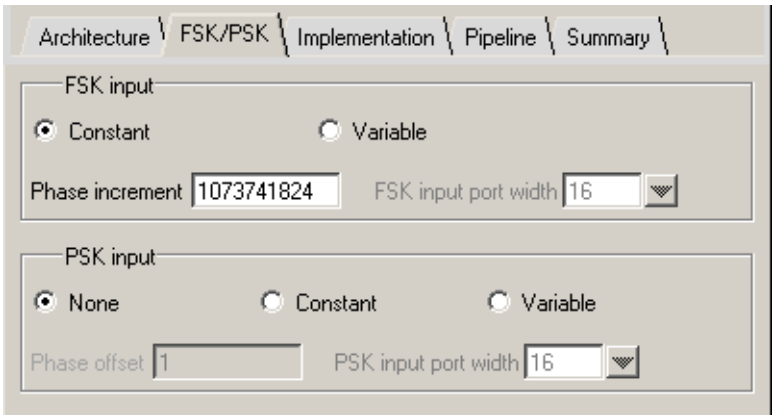
QAM Mode
This parameter indicates whether Quadrature Amplitude Modulation functionality is required. If “Yes,” QAM input and output ports are added to the IP and the parameter QAM input port width must be defined by user.

QAM Input Port Width
Width of the QAM input port.

FSK/PSK Tab

Figure 3-2 shows the contents of the FSK/PSK tab.

Figure 3-2. FSK/PSK Tab



FSK Mode

FSK Input

This parameter defines whether the FSK input is a constant or a variable. If “Variable,” FSK input ports are added and the parameter FSK input port width must be defined by the user. If “Constant,” the Phase increment parameter must be defined.

Phase Increment

Phase increment value. This value determines the phase increment that is added to the phase accumulator at every clock. This decides the frequency of the output waveform. In multi-channel modes, a phase increment must be specified for each channel.

FSK Input Port Width

Width of the fskin port. This must be less than the parameter Phase resolution.

PSK Mode

PSK Input

This parameter determines if Phase Shift Keying input is used and if used, whether it is a constant or variable. If “Constant,” a fixed value defined by Phase offset is used for the increment. If “Variable,” PSK input ports are added and the user must define the parameter PSK input port width.

Phase Offset

Phase offset value. Determines the phase offset that is added to the accumulated phase at every clock. This decides the phase of the output waveform. In multi-channel modes, a phase offset must be specified for each channel.

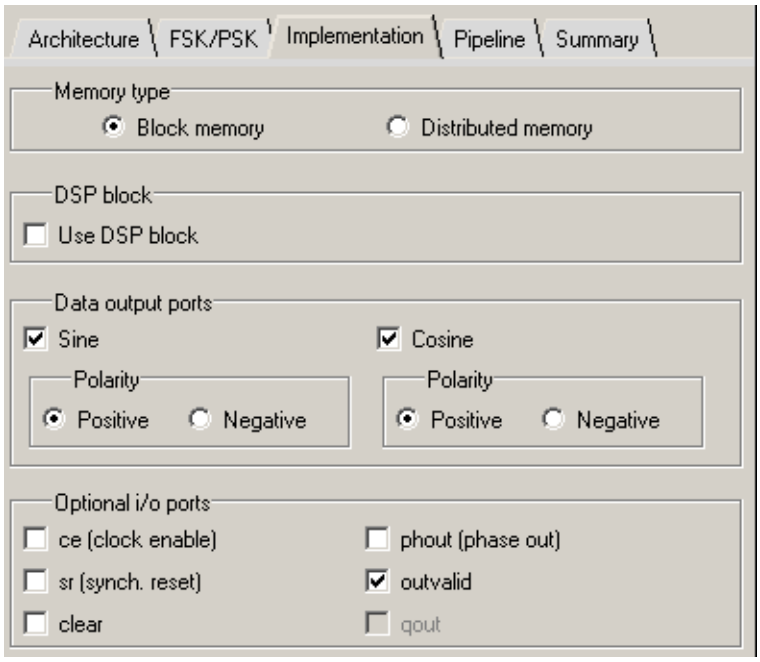
PSK Input Port Width

Width of the pskin port. This must be equal to or less than the parameter Phase resolution.

Implementation Tab

Figure 3-3 shows the contents of the Implementation tab.

Figure 3-3. Implementation Tab



Memory Type

Memory Type

This parameter defines whether block or distributed memories are used. It provides the user with additional flexibility of memory/logic resource utilization.

DSP Block

Use DSP Block

This parameter defines whether DSP blocks are used. This option is available only for Trigonometric correction, Sum of angles and QAM modes.

Data Output Ports

Sine

This parameter determines whether the sine output port is available in the core. If QAM mode is “No” and Cosine is “No,” then Sine must be “Yes.”

Cosine

This parameter determines whether the cosine output port is available in the core. If QAM mode is “No” and Sine is “No,” then Cosine must be be “Yes.”

Sine Polarity

This parameter defines polarity of the sine output. It could be positive or negative.

Cosine Polarity

This parameter defines polarity of the cosine output. It could be positive or negative.

Optional I/O Ports

ce
Determines whether the input port ce (clock enable) is present.

sr
Determines whether the input port sr (synchronous reset) is present.

clear
Determines whether the input port clear is present. This signal clears the phase accumulator (or presets the accumulator with the fixed phase offset, if provided).

phout
This option determines whether the optional phase output is required. If "Yes," the output port phout is added.

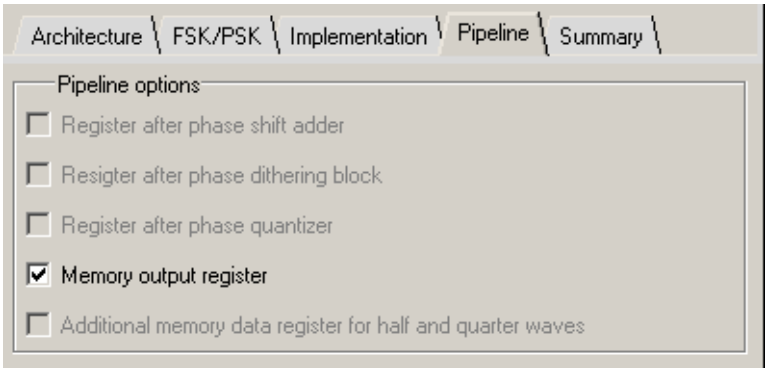
outvalid
This option determines whether the output port outvalid is present.

qout
This option determines whether qout port is present. This port is available only if QAM mode is selected.

Pipeline Tab

Figure 3-4 shows the contents of the Pipeline tab.

Figure 3-4. Pipeline Tab



Pipeline Options

Register After Phase Shift Adder
This option places a register after the phase shift adder if PSK input is "Variable." This prevents any performance degradation due to phase adder, but the output is delayed by one more clock cycle.

Register After Phase Dithering Block
This option places a register after the dithering block if "Dithering" is chosen for phase correction. This prevents any performance degradation due to phase dithering, but the output is delayed by one more clock cycle.

Register After Phase Quantizer
This option places a register after the phase quantizer if Wave size is "quarter." This prevents any performance degradation in quarter-wave modes, but the output is delayed by one more clock cycle.

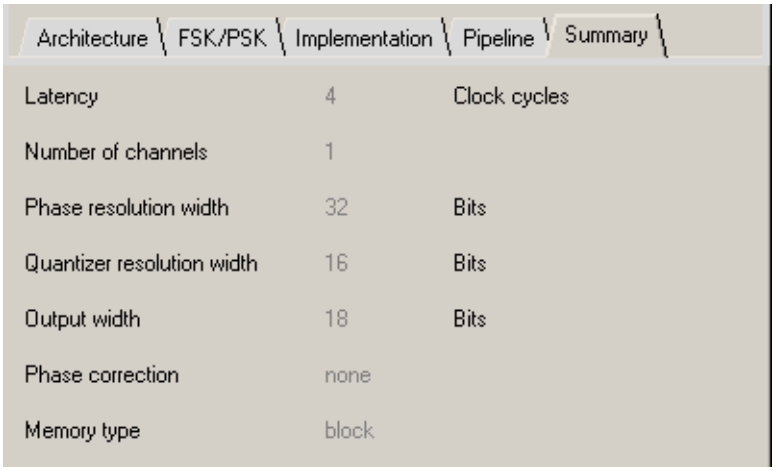
Memory Output Register
This option selects the optional memory output register in the sysMEM™ EBR block RAMs. This improves the performance of the trigonometric look-up tables, especially when multiple sysMEM EBR blocks are used for the look-up-table. The output is delayed by one more clock cycle if this option is chosen.

Additional Memory Data Register for Half and Quarter Waves
This option places an additional register in the memory data path. This prevents any performance degradation in half-wave or quarter-wave modes. The output is delayed by one more clock cycle if this option is chosen.

Summary Tab

Figure 3-4 shows the contents of the Summary tab. This tab presents a summary of various aspects of the generated NCO IP core based on the specified parameters.

Figure 3-5. Summary Tab



IP Core Generation

This chapter provides information on how to generate the NCO IP core using the IPexpress tool included in the Diamond and ispLEVER software, and how to include the core in a top-level design.

Licensing the IP Core

An IP core- and device-specific license is required to enable full, unrestricted use of the NCO IP corein a complete, top-level design. Instructions on how to obtain licenses for Lattice IP cores are given at:

http://www.latticesemi.com/products/intellectualproperty/aboutip/isplevercoreonlinepurchas.cfm

Users may download and generate the NCO IP core and fully evaluate the core through functional simulation and implementation (synthesis, map, place and route) without an IP license. The NCO IP corealso supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core that operate in hardware for a limited time (approximately four hours) without requiring an IP license. See "Hardware Evaluation" on page 32 for further details. However, a license is required to enable timing simulation, to open the design in the Diamond or ispLEVER EPIC tool, and to generate bitstreams that do not include the hardware evaluation timeout limitation.

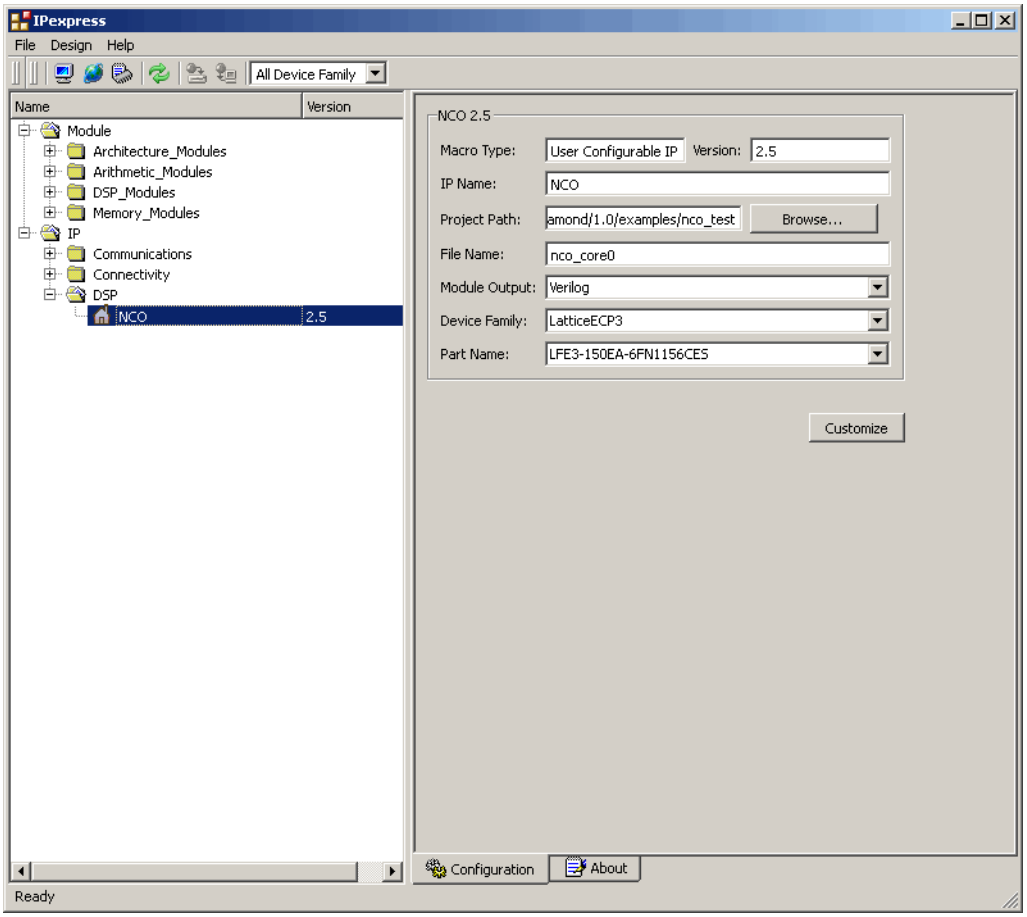
Getting Started

The NCO IP core is available for download from Lattice's IP server using the IPexpress tool. The IP files are automatically installed using ispUPDATE technology in any customer-specified directory. After the IP core has been installed, the IP core will be available in the IPexpress GUI dialog box shown in Figure 4-1.

The ispLEVER IPexpress tool GUI dialog box for the NCO IP core is shown in Figure 4-1. To generate a specific IP core configuration the user specifies:

- **Project Path** – Path to the directory where the generated IP files will be loaded.
- **File Name** – "username" designation given to the generated IP core and corresponding folders and files.
- **(Diamond) Module Output** – Verilog or VHDL.
- **(ispLEVER) Design Entry Type** – Verilog HDL or VHDL.
- **Device Family** – Device family to which IP is to be targeted (e.g. LatticeSCM, Lattice ECP2M, LatticeECP3, etc.). Only families that support the particular IP core are listed.
- **Part Name** – Specific targeted part within the selected device family.

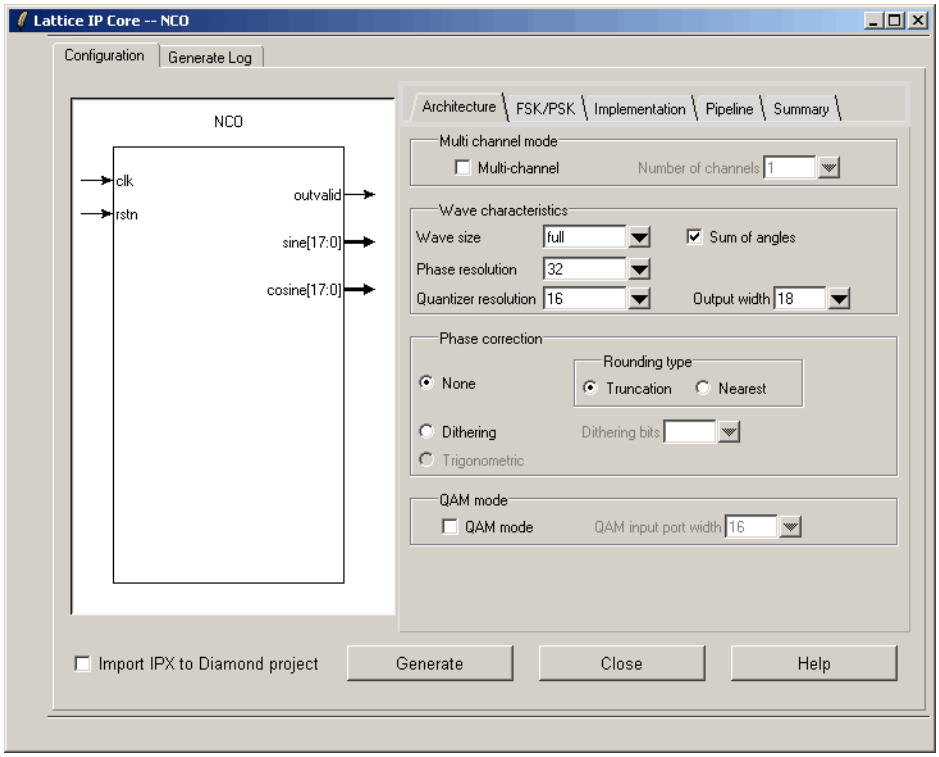
Figure 4-1. The IPexpress Tool Dialog Box (Diamond Version)



Note that if the IPexpress tool is called from within an existing project, Project Path, Module Output (Design Entry in ispLEVER), Device Family and Part Name default to the specified project parameters. Refer to the IPexpress tool online help for further information.

To create a custom configuration, the user clicks the **Customize** button in the IPexpress tool dialog box to display the NCO IP coreConfiguration GUI, as shown in Figure 4-2. From this dialog box, the user can select the IP parameter options specific to their application. Refer to “Parameter Settings” on page 20 for more information on the NCO IP coreparameter settings.

Figure 4-2. The IPexpress Tool Dialog Box - Configuration GUI (Diamond Version)



IPexpress-Created Files and Top Level Directory Structure

When the user clicks the **Generate** button in the IP Configuration dialog box, the IP core and supporting files are generated in the specified "Project Path" directory. The directory structure of the generated files is shown in Figure 4-3.

Figure 4-3. LatticeECP3 NCO IP core Directory Structure

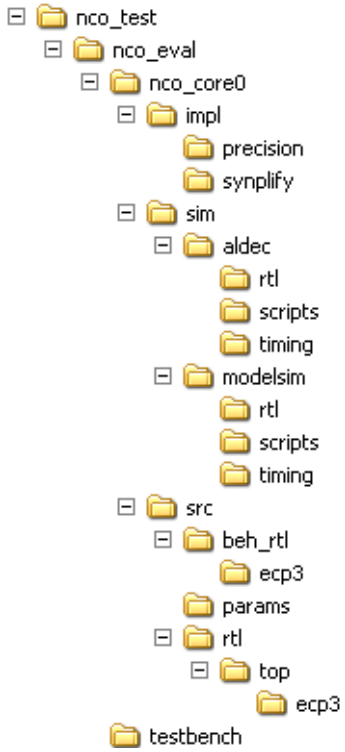


Table 4-1 provides a list of key files created by the IPexpress tool. The names of most of the created files are customized to the user's module name specified in the IPexpress tool. The files shown in Table 4-1 are all of the files necessary to implement and verify the NCO IP core in a top-level design.

Table 4-1. File List

File	Description
<*username*>_inst.v	This file provides an instance template for the IP.
<*username*>.v	This file provides a wrapper for the NCO core for simulation.
<*username*>_beh.v	This file provides a behavioral simulation model for the NCO core.
<*username*>_bb.v	This file provides the synthesis black box for the user's synthesis.
<*username*>.ngo	The ngo files provide the synthesized IP core.
<*username*>.lpc	This file contains the IPexpress tool options used to recreate or modify the core in the IPexpress tool.
<*username*>.ipx	IPexpress package file (Diamond only). This is a container that holds references to all of the elements of the generated IP core required to support simulation, synthesis and implementation. The IP core may be included in a user's design by importing this file to the associated Diamond project.
pmi_*.ngo	One or more files implementing synthesized memory modules used in the IP core.
*.mem	ROM initialization files.

Table 4-1. File List (Continued)

File	Description
<*username*>_generate.tcl	Created when GUI “Generate” button is pushed, invokes generation, may be run from command line.
<*username*>_generate.log	IPexpress scripts log file.
<*username*>_gen.log	IPexpress IP generation log file

Instantiating the Core

The generated NCO IP core package includes black-box (<*username*>_bb.v) and instance (<*username*>_inst.v) templates that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file that can be used as an instantiation template for the IP core is provided in \<*project_dir*>\nco_eval\<*username*>\src\rtl\top. Users may also use this top-level reference as the starting template for the top-level for their complete design.

Running Functional Simulation

Simulation support for the NCO IP core is provided for Aldec Active-HDL (Verilog and VHDL) simulator, Mentor Graphics ModelSim simulator. The functional simulation includes a configuration-specific behavioral model of the NCO IP core. The test bench sources stimulus to the core, and monitors output from the core. The generated IP core package includes the configuration-specific behavior model (<*username*>_beh.v) for func-tional simulation in the “Project Path” root directory. The simulation scripts supporting ModelSim evaluation simulation is provided in \<*project_dir*>\nco_eval\<*username*>\sim\modelsim\scripts. The simulation script supporting Aldec evaluation simulation is provided in \<*project_dir*>\nco_eval\<*username*>\sim\aldec\scripts. Both ModelSim and Aldec simulation is supported via test bench files provided in \<project_dir>\nco_eval\testbench. Models required for simulation are provided in the corresponding \models folder.

Users may run the Aldec evaluation simulation by doing the following:

1. Open Active-HDL.
2. Under the Tools tab, select **Execute Macro**.
3. Browse to folder \<*project_dir*>\nco_eval\<*username*>\sim\aldec\scripts and execute one of the "do" scripts shown.

Users may run the Modelsim evaluation simulation by doing the following:

1. Open ModelSim.
2. Under the File tab, select **Change Directory** and choose the folder <*project_dir*>\nco_eval\<*username*>\sim\modelsim\scripts.
3. Under the Tools tab, select **Execute Macro** and execute the ModelSim “do” script shown.

Note: *When the simulation completes, a pop-up window will appear asking “Are you sure you want to finish?” Answer “No” to analyze the results (answering “Yes” closes ModelSim).*

Synthesizing and Implementing the Core in a Top-Level Design

The NCO IP core itself is synthesized and provided in NGO format when the core is generated through IPexpress. You may combine the core in your own top-level design by instantiating the core in your top-level file as described in "Instantiating the Core" on page 31 and then synthesizing the entire design with either Synplify or Precision RTL Synthesis.

The following text describes the evaluation implementation flow for Windows platforms. The flow for Linux and UNIX platforms is described in the Readme file included with the IP core.

The top-level file <*userame*>_top.v is provided in \<*project_dir*>\nco_eval\<*username*>\src\rtl\top. Push-button implementation of the reference design is supported via the project file <*username*>.ldf (Diamond) or .syn (ispLEVER) located in \<*project_dir*>\nco_eval\<*username*>\impl\(synplify or precision).

To use this project file in Diamond:

1. Choose **File > Open > Project**.
2. Browse to \<*project_dir*>\nco_eval\<*username*>\impl\synplify (or precision) in the Open Project dialog box.
3. Select and open <*username*>_.ldf. At this point, all of the files needed to support top-level synthesis and implementation will be imported to the project.
4. Select the **Process** tab in the left-hand GUI window.
5. Implement the complete design via the standard Diamond GUI flow.

To use this project file in ispLEVER:

1. Choose **File > Open Project.**
2. Browse to \<*project_dir*>\nco_eval\<*username*>\impl\synplify (or precision) in the Open Project dialog box.
3. Select and open <username>.syn. At this point, all of the files needed to support top-level synthesis and implementation will be imported to the project.
4. Select the device top-level entry in the left-hand GUI window.
5. Implement the complete design via the standard ispLEVER GUI flow.

Hardware Evaluation

The NCO IP core supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core that operate in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. It may also be used to evaluate the core in hardware in user-defined designs.

Enabling Hardware Evaluation in Diamond:

Choose **Project > Active Strategy > Translate Design Settings**. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default.

Enabling Hardware Evaluation in ispLEVER:

In the Processes for Current Source pane, right-click the **Build Database** process and choose **Properties** from the dropdown menu. The hardware evaluation capability may be enabled/disabled in the Properties dialog box. It is enabled by default.

Updating/Regenerating the IP Core

By regenerating an IP core with the IPexpress tool, you can modify any of its settings including: device type, design entry method, and any of the options specific to the IP core. Regenerating can be done to modify an existing IP core or to create a new but similar one.

Regenerating an IP Core in Diamond

To regenerate an IP core in Diamond:

1. In IPexpress, click the **Regenerate** button.
2. In the Regenerate view of IPexpress, choose the IPX source file of the module or IP you wish to regenerate.
3. IPexpress shows the current settings for the module or IP in the Source box. Make your new settings in the T**arget** box.
4. If you want to generate a new set of files in a new location, set the new location in the **IPX Target File** box. The base of the file name will be the base of all the new file names. The IPX Target File must end with an .ipx extension.
5. Click **Regenerate.** The module's dialog box opens showing the current option settings.
6. In the dialog box, choose the desired options. To get information about the options, click **Help**. Also, check the About tab in IPexpress for links to technical notes and user guides. IP may come with additional information. As the options change, the schematic diagram of the module changes to show the I/O and the device resources the module will need.
7. To import the module into your project, if it's not already there, select **Import IPX to Diamond Project** (not available in stand-alone mode).
8. Click **Generate**.
9. Check the Generate Log tab to check for warnings and error messages.
10. Click **Close**.

The IPexpress package file (.ipx) supported by Diamond holds references to all of the elements of the generated IP core required to support simulation, synthesis and implementation. The IP core may be included in a user's design by importing the .ipx file to the associated Diamond project. To change the option settings of a module or IP that is already in a design project, double-click the module's .ipx file in the File List view. This opens IPexpress and the module's dialog box showing the current option settings. Then go to step 6 above.

Regenerating an IP Core in ispLEVER

To regenerate an IP core in ispLEVER:

1. In the IPexpress tool, choose **Tools > Regenerate IP/Module**.
2. In the Select a Parameter File dialog box, choose the Lattice Parameter Configuration (.lpc) file of the IP core you wish to regenerate, and click **Open**.
3. The Select Target Core Version, Design Entry, and Device dialog box shows the current settings for the IP core in the Source Value box. Make your new settings in the Target Value box.

4. If you want to generate a new set of files in a new location, set the location in the LPC Target File box. The base of the .lpc file name will be the base of all the new file names. The LPC Target File must end with an .lpc extension.
5. Click **Next**. The IP core's dialog box opens showing the current option settings.
6. In the dialog box, choose desired options. To get information about the options, click **Help**. Also, check the About tab in the IPexpress tool for links to technical notes and user guides. The IP core might come with additional information. As the options change, the schematic diagram of the IP core changes to show the I/O and the device resources the IP core will need.
7. Click **Generate**.
8. Click the **Generate Log** tab to check for warnings and error messages.

Support Resources

This chapter contains information about Lattice Technical Support, additional references, and document revision history.

Lattice Technical Support

There are a number of ways to receive technical support.

Online Forums

The first place to look is Lattice Forums (http://www.latticesemi.com/support/forums.cfm). Lattice Forums contain a wealth of knowledge and are actively monitored by Lattice Applications Engineers.

Telephone Support Hotline

Receive direct technical support for all Lattice products by calling Lattice Applications from 5:30 a.m. to 6 p.m. Pacific Time.

- For USA & Canada: 1-800-LATTICE (528-8423)
- For other locations: +1 503 268 8001

In Asia, call Lattice Applications from 8:30 a.m. to 5:30 p.m. Beijing Time (CST), +0800 UTC. Chinese and English language only.

- For Asia: +86 21 52989090

E-mail Support

- techsupport@latticesemi.com
- techsupport-asia@latticesemi.com

Local Support

Contact your nearest Lattice Sales Office.

Internet

www.latticesemi.com

References

LatticeECP/EC

- HB1000, *LatticeECP/EC Family Handbook*

LatticeECP2M

- HB1003, *LatticeECP2M Family Handbook*

LatticeECP3

- HB1009, *LatticeECP3 Family Handbook*

LatticeSC/M

- DS1004, *LatticeSC/M Family Data Sheet*

LatticeXP

- HB1001, *LatticeXP Family Handbook*

LatticeXP2

- DS1009, *Lattice XP2 Datasheet*

Revision History

Date	Document Version	IP Version	Change Summary
—	—	1.0	Previous Lattice releases.
August 2006	02.1	2.0	NCO version 2.0, with LatticeECP/EC, Lattice ECP2, LatticeSC, and LatticeXP support for IPexpress.
December 2006	02.2	2.1	Updated appendices and added support for the LatticeECP2M family.
June 2008	02.3	2.2	Updated appendices.
April 2009	02.4	2.3	Updated appendices and added support for the LatticeECP3 family.
June 2010	02.5	2.5	Added support for Diamond software.
			Divided document into chapters. Added table of contents.
			Added Quick Facts table in Chapter 1, “Introduction.”
			Added new content in Chapter 4, “IP Core Generation.”

Appendix A:

Resource Utilization

This appendix gives resource utilization information for Lattice FPGAs using the NCO IP core. The IP configurations shown in this chapter were generated using the IPexpress software tool. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the Diamond and ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and Diamond and ispLEVER help systems. For more information on the Diamond or ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

LatticeEC Devices

The utilization data shown in Table A-1 is derived from the parameter settings listed in Table A-2.

Table A-1. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	Slices	LUTs	Registers	I/Os	18x18 Multipliers	sysMEM EBRs	fMAX (MHz)
1	1000	1710	715	39	NA	3	116
2	194	211	262	104	NA	2	230
3	2235	3730	1833	217	NA	5	123

1. Performance and utilization data are generated targeting an LFEC20E-5F672C device using Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeEC family.

Ordering Part Number

The Ordering Part Number (OPN) for the NCO targeting LatticeEC devices is NCO-DDS-E2-U2.

Table A-2. Parameter Settings of the Evaluation Packages

Parameter Name (in Documentation)	Parameter Name (in .lpc file)	Config 1	Config 2	Config 3
FSK input	fsk_in	Constant	Variable	Variable
FSK Phase increment	phase_increment_value	1073741824	n/a	n/a
FSK input port width	fsk_width	n/a	31	27
PSK input	psk_in	Absent	Variable	Variable
PSK Phase offset	phase_offset_value	n/a	n/a	n/a
PSK input port width	psk_width	n/a	32	28
Memory type	memory_type	Block	Block	Block
Sine Polarity	sin_polarity	Positive	Positive	Positive
Cosine Polarity	cos_polarity	Positive	Positive	Positive
Phase correction	phase_correction	None	None	Dither
Dithering bits	dither_level	n/a	n/a	4
Rounding type	rounding_method	Truncation	Truncation	n/a
QAM input port width	qam_input_width	n/a	n/a	18
Wave size	wave_storage	Full	Quarter	Full
Number of channels	num_channels	1	1	16
Phase resolution	accumulator_width	32	32	28
Quantizer resolution	quantizer_width	16	12	20
Output width	output_width	18	18	18
sr	sync_reset_port	Absent	Absent	Absent
ce	clock_enable_port	Absent	Absent	Absent

Table A-2. Parameter Settings of the Evaluation Packages (Continued)

Parameter Name (in Documentation)	Parameter Name (in .lpc file)	Config 1	Config 2	Config 3
outvalid	output_valid_port	Present	Present	Present
phout	ph_out_port	Absent	Absent	Absent
Sine	sin_port	Present	Present	Present
Cosine	cos_port	Present	Present	Present
QAM mode	qam_mode	Absent	Absent	Present
qout	q_out_port	Absent	Absent	Present
clear	clear_port	Absent	Absent	Present
Memory output register	mor_latency	Present	Present	Present
Register after phase quantizer	quantizer_latency	Absent	Present	Absent
Additional memory data register	mpu_latency	Absent	Present	Absent
Sum of angles	sum_of_angles	Present	Absent	Present
Register after phase dithering block	dither_latency	Absent	Absent	Present
Register after phase shift adder	var_phase_offset_latency	Absent	Present	Present
Use DSP block	dsp_block	n/a for LatticeEC, Present for LatticeECP	n/a	n/a for LatticeEC, Present for LatticeECP

LatticeECP Devices

The utilization data shown in Table A-3 is derived from the parameter settings listed in Table A-2 on page 37.

Table A-3. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	Slices	LUTs	Registers	I/Os	18x18 Multipliers[2]	sysMEM EBRs	fMAX (MHz)
1	27	7	44	39	4	3	215
2	194	211	262	104	0	2	215
3	305	316	494	217	8	5	226

1. Performance and utilization data are generated targeting an LFECP20E-5F672C device using Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP family.
2. One DSP block provides two MULT18X18ADDSUBs.

Ordering Part Number

The Ordering Part Number (OPN) for the NCO targeting LatticeECP devices is NCO-DDS-E2-U2.

LatticeECP2 Devices

The utilization data shown in Table A-4 is derived from the parameter settings listed in Table A-2 on page 37.

Table A-4. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	Slices	LUTs	Registers	I/Os	MULT18X18 ADDSUB[2]	sysMEM EBRs	f_{MAX} (MHz)
1	24	5	44	39	4	3	368
2	200	222	262	104	0	1	370
3	287	282	494	217	8	3	325

1. Performance and utilization data are generated targeting an LFE2-50E-7F672C device using Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP2 family.
2. One DSP block provides two MULT18X18ADDSUBs.

Ordering Part Number

The Ordering Part Number (OPN) for the NCO targeting LatticeECP2 devices is NCO-DDS-P2-U2.

LatticeECP2M Devices

The utilization data shown in Table A-5 is derived from the parameter settings listed in Table A-2 on page 37.

Table A-5. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	Slices	LUTs	Registers	I/Os	18x18 Multipliers[2]	sysMEM EBRs	f_{MAX} (MHz)
1	24	5	44	39	4	3	347
2	200	222	262	104	0	1	361
3	287	282	494	217	8	3	256

1. Performance and utilization data are generated targeting an LFE2M-35E-7F484C device using Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP2M family.
2. One DSP block provides two MULT18X18ADDSUBs.

Ordering Part Number

The Ordering Part Number (OPN) for the NCO targeting LatticeECP2M devices is NCO-DDS-PM-U2.

LatticeECP3 Devices

The utilization data shown in Table A-6 is derived from the parameter settings listed in Table A-2 on page 37.

Table A-6. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	Slices	LUTs	Registers	I/Os	18x18 Multipliers	sysMEM EBRs	fMAX (MHz)
1	25	6	44	39	4	3	340
2	163	220	262	104	0	1	340
3	302	310	494	217	8	3	320

1. Performance and utilization data are generated targeting an LFE3-95E-7FN672CES device using Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP3 family.

Ordering Part Number

The Ordering Part Number (OPN) for the NCO targeting LatticeECP3 devices is NCO-DDS-E3-U2.

LatticeSC/M Devices

The utilization data shown in Table A-7 is derived from the parameter settings listed in Table A-2 on page 37.

Table A-7. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	Slices	LUTs	Registers	I/Os	18x18 Multipliers	sysMEM EBRs	fMAX (MHz)
1	1472	2194	1071	39	N/A	3	242
2	152	210	262	104	NA	1	375
3	3583	5153	2549	217	NA	3	239

1. Performance and utilization data are generated targeting an LFSC3GA25E-7F900C device using Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeSC family.

Ordering Part Number

The Ordering Part Number (OPN) for the NCO targeting LatticeSC devices is NCO-DDS-SC-U2.

LatticeXP Devices

The utilization data shown in Table A-8 is derived from the parameter settings listed in Table A-2 on page 37.

Table A-8. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	Slices	LUTs	Registers	I/Os	18x18 Multipliers	sysMEM EBRs	fMAX (MHz)
1	1000	1710	715	39	N/A	3	112
2	194	211	262	104	N/A	2	219
3	2235	3730	1833	217	N/A	5	115

1. Performance and utilization data are generated targeting an LFXP20E-5F484C device using Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the
LatticeXP family.

Ordering Part Number

The Ordering Part Number (OPN) for the NCO targeting LatticeXP devices is NCO-DDS-XM-U2.

LatticeXP2 Devices

The utilization data shown in Table A-9 is derived from the parameter settings listed in Table A-2 on page 37.

Table A-9. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	Slices	LUTs	Registers	I/Os	18x18 Multipliers[2]	sysMEM EBRs	fMAX (MHz)
1	24	5	44	39	4	3	314
2	200	222	262	104	0	1	314
3	287	282	494	217	8	3	314

1. Performance and utilization data are generated targeting an LFXP2-17E-7F484C device using Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeXP2 family.
2. One DSP block provides two MULT18X18ADDSUBs.

Ordering Part Number

The Ordering Part Number (OPN) for the NCO targeting LatticeXP2 devices is NCO-DDS-X2-U2.

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Development Software ***category:***

Click to view products by Lattice ***manufacturer:***

Other Similar products are found below :

RAPPID-560XBSW RAPPID-567XFSW DG-ACC-NET-CD SRP004001-01 SW006021-1NH SW163052 SYSWINEV21 Core429-SA SW500006-HPA CWP-BASIC-FL W128E13 CWP-PRO-FL SYSMACSE210L SYSMACSE203L AD-CCES-NODE-1 NT-ZJCAT1-EV4 CWA-BASIC-FL RAPPID-567XKSW CWA-STANDARD-R SW89CN0-ZCC CWA-LS-DVLPR-NL VDSP-21XX-PCFLOAT RAPPID-563XMSW IPS-EMBEDDED SWR-DRD-L-01 SDAWIR-4532-01 SYSMAC-SE201L MPROG-PRO535E AFLCF-08-LX-CE060-R21 WS02-CFSC1-EV3-UP SYSMAC-STUDIO-EIPCPLR LIB-PL-PC-N-1YR-DISKID SYSMACSE2XXL LS1043A-SWSP-PRM 1120270005 1120270006 MIKROBASIC PRO FOR FT90X (USB DONGLE) MIKROC PRO FOR AVR (USB DONGLE LICENSE) MIKROC PRO FOR FT90X (USB DONGLE) MIKROBASIC PRO FOR AVR (USB DONGLE LICEN MIKROBASIC PRO FOR FT90X MIKROC PRO FOR DSPIC30/33 (USB DONGLE LI MIKROC PRO FOR FT90X MIKROC PRO FOR PIC32 (USB DONGLE LICENSE 52202-588 MIKROPASCAL PRO FOR ARM (USB DONGLE LICE MIKROPASCAL PRO FOR FT90X MIKROPASCAL PRO FOR FT90X (USB DONGLE) MIKROPASCAL PRO FOR PIC32 (USB DONGLE LI SW006021-2H