NCO Compiler MegaCore Function

Solution Brief 49 **September 2000, ver. 1.0**

Target Applications:
Data Storage and Retrieval Systems, Modulators, Demodulators, and Digital PLLs

Family:
APEX™ 20K, ACEX™, FLEX® 10, FLEX 8000, and FLEX 6000

Ordering Code:
PLSM-NCO

Vendor:

101 Innovation Drive
San Jose, CA 95134
http://www.altera.com
Tel. (408) 544-7000

Features

- Optimized for multiple device architectures (including APEX™ 20K, ACEX™, FLEX® 10K, FLEX 8000, and FLEX 6000 devices)
- Multiple implementation architectures
  - ROM-based implementation using device embedded array blocks (EABs), embedded system blocks (ESBs), or external ROM
  - Coordinate rotation digital computer (CORDIC)-based implementation using logic, EABs, or ESBs
- Variable width-frequency modulation input
- User-defined frequency resolution
- User-defined angular and magnitude precision
- Dynamically generated MATLAB files
  - Standard M-Files and Simulink S-functions
  - Clock cycle and bit-accurate models
- Reference designs provided for a digital phase-locked loop (PLL), direct digital synthesis (DDS), and quadrature amplitude modulation (QAM) / quadrature phase shift keying (QPSK) modulator

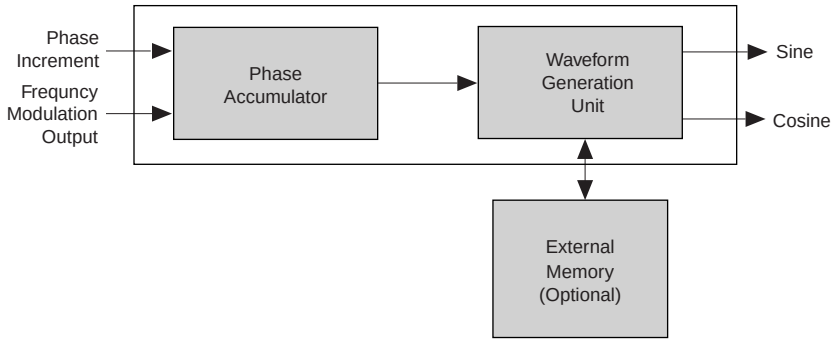
General Description

Digital numerically controlled oscillators (NCOs) generate a digital representation of sine and cosine waves. NCOs are typically used as building blocks in digital signal processing systems such as modulators, demodulators, digital PLLs, and symbol recovery circuits. NCOs can be used to generate a carrier or to modulate a signal onto a carrier.

The Altera® digital signal processing (DSP) MegaCore® function provides a solution for integrating NCOs into a digital communications system. Optimized for APEX 20K, ACEX, FLEX 10K, FLEX 8000, and FLEX 6000 devices, the NCO Compiler greatly enhances your productivity by allowing you to focus efforts on developing the system's custom logic.

Figure 1 shows the NCO Compiler functional block diagram.

Figure 1. NCO Compiler Functional Block Diagram



Functional Description

The NCO Compiler MegaCore function consists of a phase accumulator that calculates the value of a phase, and a waveform generation unit. Both the width of the accumulator and the phase-increment data input to the accumulator effect the frequency and resolution of the output sine wave. In the ROM version, the phase accumulator output addresses a ROM containing data describing a sinusoidal waveform. The ROM address width determines the angular precision of the sinusoid generated by the waveform generation unit. The NCO Compiler supports both ROM and CORDIC architectures.

ROM Architecture

The ROM containing the sine/cosine wave can be large or small. For the small ROM architecture option, only the first 90 degrees of the sinusoidal waveform are stored in ROM. The rest of the sine and cosine cycle is derived from the first 90 degrees. For the large ROM architecture option, the entire sine and cosine waves are held in two ROM memories.

CORDIC Architecture

A CORDIC algorithm calculates trigonomic functions such as sine and cosine. The CORDIC-based NCO function computes the sine and cosine of an input phase value by iteratively shifting the phase angle to approximate the cartesian coordinate values for the input angle. At the end of the CORDIC iteration, the x and y coordinates of an angle represent the cosine and sine of that angle.

Architecture Tradeoffs

The NCO ROM-based architecture permits flexible design implementation and very high spectral purity in the resulting sinusoid. Increasing the ROM address width generates a sinusoid of higher purity and spurios-free dynamic range (SFDR), yet may require too much memory for a given design. ROM implementations are best used for applications that require higher frequency and lower precision. A ROM block stores the sine or cosine values and outputs every clock cycle, operating at clock rates of 70 to 160 MHz.

For applications requiring high-precision waveforms, a CORDIC architecture is more area-efficient than the equivalent ROM implementation. CORDIC implementations, in which the sine and cosine values are created by the CORDIC algorithm, are most effective for applications that require lower frequency and higher precision. CORDIC implementations use a very small ROM block and one clock cycle for every bit of data (e.g., eight data bits require eight clock cycles to calculate the result).

Performance and Implementation

Table 1 provides performance statistics for ROM-based NCO functions.

Table 1. NCO Function Performance

Memory Type	Output	Accumulator Width	Address Width	Magnitude Precision	Logic Cells	EABs/ESBs	Speed (MHz)
Large, internal	Dual	24	9	16	56	8	135.0
Small, internal	Dual	32	11	17	264	2	119.0
Small, internal	Dual	24	12	9	171	2	107.5

MegaWizard Plug-In Manager

The MegaWizard® Plug-In Manager, which you can use within the Quartus™ or MAX+PLUS® II software or as a standalone application, allows you to create or modify the design files that contain megafunction variations. You can customize the NCO Compiler function to meet the needs of your application by defining the parameters within the MegaWizard Plug-In Manager. Table 2 highlights the parameters for the NCO Compiler function.

Table 2. NCO Compiler Parameters

Parameter	Description
Accumulator precision	Controls the frequency resolution
Angular precision	Determines the address width of the waveform generation unit
Magnitude precision	Selects the bit width of the outputs
CORDIC architecture	Chooses to implement the ROM in internal EABs/ESBs or as logic
Modulation input	Modulates the sinusoid and indicates the frequency modulation input width
Outputs	Chooses sine or both sine and cosine

The MegaWizard Plug-In generates the following files:

- An AHDL Text Design File (**.tdf**), VHDL Design File (**.vhd**) and Verilog Design File (**.v**), used to instantiate an NCO function in your design
- Symbol File (**.sym**), used to instantiate the function into a schematic design
- MATLAB models to simulate the NCO functionality
- Simulink models to drag and drop into your system design
- Verilog HDL models used for simulation in other EDA tools

Obtaining and Installing the NCO Compiler Function

You can download the NCO Compiler and other MegaCore functions from the Altera web site at **http://www.altera.com**, or obtain them from your local Altera representative.

Generating an NCO Function

To create an NCO function using the MegaWizard Plug-In, choose the **MegaWizard Plug-In Manager** (File menu) in the Quartus or MAX+PLUS II software, or start the standalone version of the MegaWizard Plug-In Manager. When the MegaWizard Plug-In Manager dialog box opens, specify that you want to create a new custom megafunction. Select the NCO Compiler function from the DSP MegaCore drop-down list in the **Available Megafunctions** dialog box.

To learn more about Altera's NCO Compiler, refer to the *NCO Compiler MegaCore Function User Guide.*

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com