Electrical and Computer Engineering
EE3623 Embedded Systems Design Laboratory

Implementation of Sine and Cosine Functions using the CORDIC Xilinx LogiCORE Block

Dennis Silage, PhD

The Coordinate Coordinate Rotation Digital Computer (CORDIC) is a simple and efficient algorithm to calculate hyperbolic and trigonometric functions, vector rotations and translation and the square root. The CORDIC is commonly used when no hardware multiplier is available as the only operations it requires are addition, subtraction, bit shifting and table lookup. However, when a hardware multiplier is available, power series calculations for these functions are generally faster than the CORDIC algorithm.

The modern CORDIC algorithm was first described in 1959 by Jack E. Volder but it is similar to techniques published by Henry Briggs as early as 1624. John Stephen Walther at Hewlett-Packard further generalized the algorithm.

A complete description of the Xilinx LogiCORE block CORDIC algorithm is in the data sheet *DS249.pdf* posted on Blackboard. As part of this Laboratory you are to read and study this *spec sheet* for the project implementation. This Laboratory consists of an exercise in the specifications of a LogiCORE block and the implementation of the specific LogiCORE block as a nested module in a Xilinx ISE project.

The CORDIC Graphical User Interface (GUI) contains four screens for configuration:

1. Screen 1: Used to configure the functional selection and architecture of the CORDIC algorithm LogiCORE.
2. Screen 2: Used to configure the phase and magnitude data formats, optional control signals, and synchronization.
3. Screen 3: Provides options for configuring the rounding mode and input-outputs.
4. Screen 4: Provides options for advanced configuration parameters: coarse rotation, iterations, internal precision, compensation scaling and displays the latency and controls the inclusion of LogiCORE instantiation.

The representation of numbers in the CORDIC LogiCORE block is new to you but common for the representation of *floating point* numbers in binary format and known as the *Q Format Signed Numbers*.

A QN format number is an N bit 2.s complement binary number: a sign bit followed by an N bit mantissa (fraction). QN format can be used to express numbers in the range -1 to $(1 - 2^{-N})$.

An XQN format number is a QN format number left shifted by X bits. XQN format can be used to express numbers in the range: (-2^{X}) to $(2^{X} - 2^{(X-N)})$.

1Q9 Format

	SB	D8	D7	D6	D5	D4	D3	D2	D1
+1	0	1	0	0	0	0	0	0	0
-1	1	1	0	0	0	0	0	0	0
+Pi/4	0	0	1	1	0	0	1	0	0
-Pi/4	1	1	0	0	1	1	0	1	1
^ <---Binary Point									

2Q9 Format

	SB	D8	D7	D6	D5	D4	D3	D2	D1
+1	0	0	1	0	0	0	0	0	0
-1	1	1	1	0	0	0	0	0	0
+Pi	0	1	1	0	0	1	0	0	1
-Pi	1	0	0	1	1	0	1	1	1
^ <---Binary Point									

For example in 1Q10 format: 0100000000 => 01.00000000 => +1.0
1100000000 => 11.00000000 => - 1.0
For example in 2Q10 format: 0010000000 => 001.0000000 => +1.0
1110000000 => 111.0000000 => - 1.0

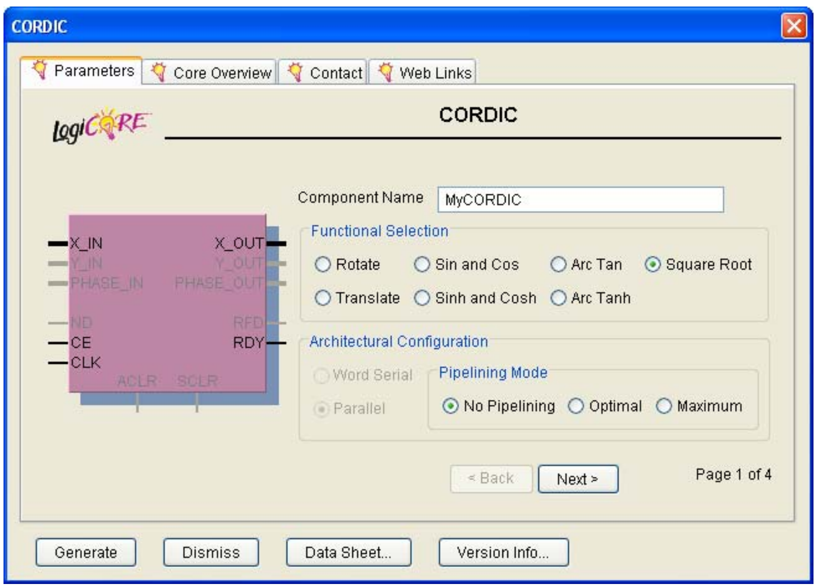
The task for this two week Laboratory for Thursday February 26th and March 5th, due after Spring Break on March 19th is as follows:

Implement the CORDIC algorithm LogiCORE block for the sine and cosine of an input phase angle *Pin* in the approximate radian range ± π/4 (± 45º) using 10 bits of resolution and are to be a 2Q10 number. For example: 0001100100. => 000.1100100 => 0.781 radians (approximately). You are to use the rotary switch to increment and decrement the input phase angle between these limits and display the radian angle in decimal (for the example +0.781) on line 1 of the LCD.

The sine and cosine output from the CORDIC LogiCORE block are to be 1Q10 numbers. For example if Pin = 0.781: sine: 0010110100. => 00.10110100 => 0.703

and cosine: 0010110110. => 00.10110110 => 0.711. You are to display the sine and cosine of the phase angle on line 2 of the LCD.

Signal	Description
PHASE_IN	Input Angle θ Range: -Pi <= PHASE_IN <= Pi
X_OUT	Output Cos(θ) Range: -1 <= X_OUT <= 1
Y_OUT	Output Sin(θ) Range: -1 <= Y_OUT <= 1



A sequence of design steps would be to first generate the appropriate phase angle in 2Q10 format as a binary number then convert it for display within the specified limits on the LCD (which would consititute a partial Laboratory). Next, implement the sine and cosine functions using the CORDIC LogiCORE block and their display on the LCD. Additional asynchronous control signals, such as the pushbuttons, can be used to initiate the process. A synchronous clock CCLK is available.

Obviously, you should use a calculator to verify that the sine and cosine of the input phase angle is correct. To enhance your project, you may want to use the Laboratory logic analyzer to measure the latency of the calculation from CE to RDY and report the results. Does the latency vary with the input phase angle?