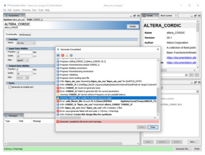
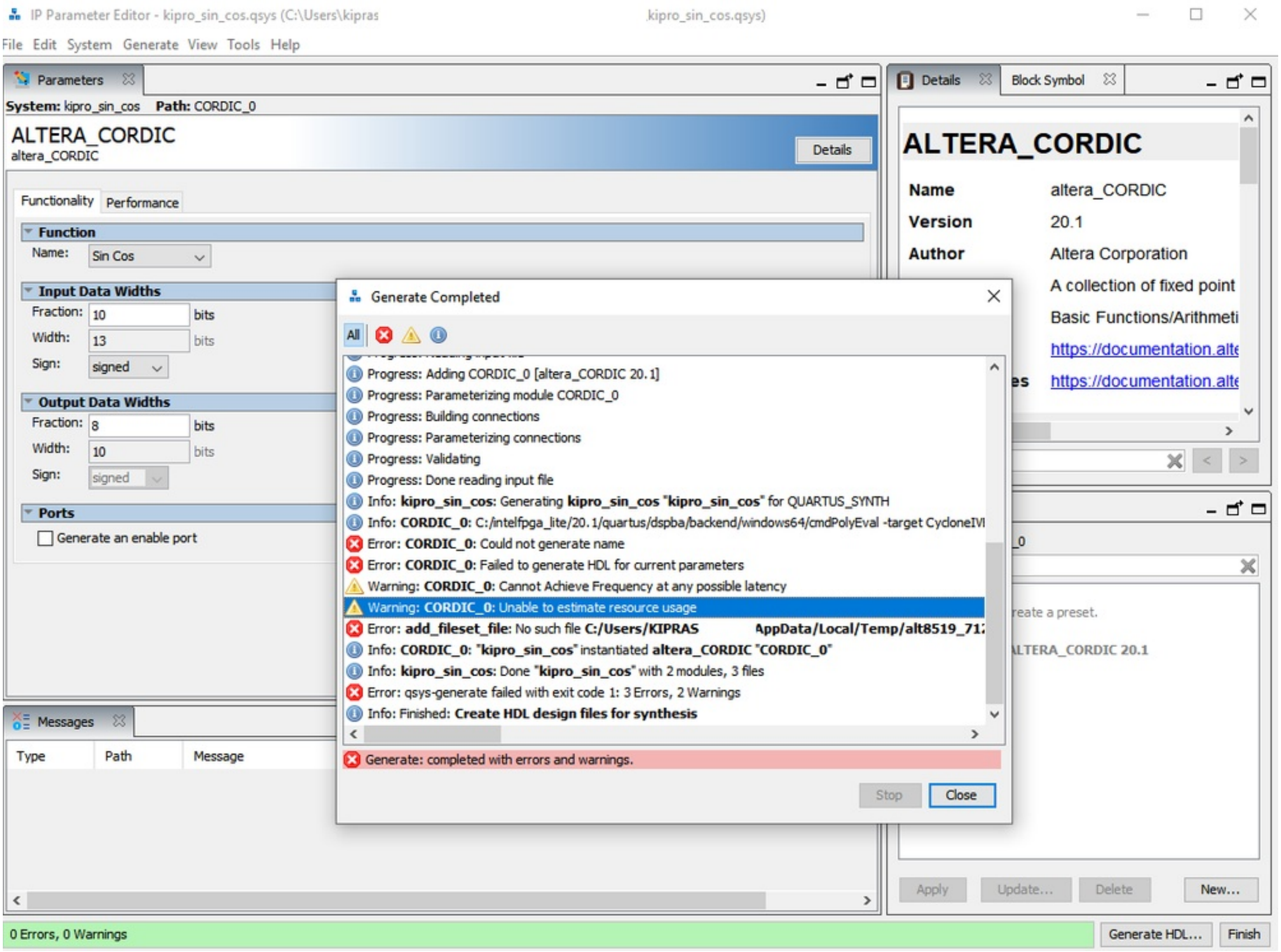


intel ALTERA_CORDIC IP Core User Guide

Home » Intel » intel ALTERA_CORDIC IP Core User Guide

Contents

intel ALTERA_CORDIC IP Core



IP Parameter Editor - kipro_sin_cos.qsys (C:\Users\kipras kipro_sin_cos.qsys)
File Edit System Generate View Tools Help
Parameters
Details
Block Symbol
Functionality
Performance
Messages
Generate HDL...
Finish
Apply
Update...
Delete
New...
Stop
Close

System: kipro_sin_cos Path: CORDIC_0

ALTERA_CORDIC
altera_CORDIC

Function
Name: Sin Cos

Input Data Widths
Fraction: 10 bits
Width: 13 bits
Sign: signed

Output Data Widths
Fraction: 8 bits
Width: 10 bits
Sign: signed

Ports
□ Generate an enable port

Name	altera_CORDIC
Version	20.1
Author	Altera Corporation
	A collection of fixed point
	Basic Functions/Arithmeti
	https://documentation.alte
es	https://documentation.alte

Generate Completed

```
Progress: Adding CORDIC_0 [altera_CORDIC 20.1]
Progress: Parameterizing module CORDIC_0
Progress: Building connections
Progress: Parameterizing connections
Progress: Validating
Progress: Done reading input file
Info: kipro_sin_cos: Generating kipro_sin_cos "kipro_sin_cos" for QUARTUS_SYNTH
Info: CORDIC_0: C:/intelfpga_lite/20.1/quartus/dspba/backend/windows64/cmdPolyEval -target CycloneIVI
Error: CORDIC_0: Could not generate name
Error: CORDIC_0: Failed to generate HDL for current parameters
Warning: CORDIC_0: Cannot Achieve Frequency at any possible latency
Warning: CORDIC_0: Unable to estimate resource usage
Error: add_fileset_file: No such file C:/Users/KIPRAS        AppData/Local/Temp/alt8519_71:
Info: CORDIC_0: "kipro_sin_cos" instantiated altera_CORDIC "CORDIC_0"
Info: kipro_sin_cos: Done "kipro_sin_cos" with 2 modules, 3 files
Error: qsys-generate failed with exit code 1: 3 Errors, 2 Warnings
Info: Finished: Create HDL design files for synthesis
```

Generate: completed with errors and warnings.

Type	Path	Message

0 Errors, 0 Warnings

ALTERA_CORDIC IP Core User Guide

- Use the ALTERA_CORDIC IP core to implement a set of fixed-point functions with the CORDIC algorithm.
- ALTERA_CORDIC IP Core Features on page 3
- DSP IP Core Device Family Support on page 3
- ALTERA_CORDIC IP Core Functional Description on page 4
- ALTERA_CORDIC IP Core Parameters on page 7
- ALTERA_CORDIC IP Core Signals on page 9

ALTERA_CORDIC IP Core Features

- Supports fixed-point implementations.
- Supports both latency and frequency driven IP cores.
- Supports both VHDL and Verilog HDL code generation.
- Produces fully unrolled implementations.
- Produces faithfully rounded results to either of the two closest representable numbers in the output.

DSP IP Core Device Family Support

Intel offers the following device support levels for Intel FPGA IP cores:

- Advance support—the IP core is available for simulation and compilation for this device family. FPGA programming file (.pof) support is not available for Quartus Prime Pro Stratix 10 Edition Beta software and as such IP timing closure cannot be guaranteed. Timing models include initial engineering estimates of delays based on early post-layout information. The timing models are subject to change as silicon testing improves the correlation between the actual silicon and the timing models. You can use this IP core for system architecture and resource utilization studies, simulation, pinout, system latency assessments, basic timing assessments (pipeline budgeting), and I/O transfer strategy (data-path width, burst depth, I/O standards tradeoffs).

- Preliminary support—Intel verifies the IP core with preliminary timing models for this device family. The IP core meets all functional requirements, but might still be undergoing timing analysis for the device family. You can use it in production designs with caution.
- Final support—Intelverifies the IP core with final timing models for this device family. The IP core meets all functional and timing requirements for the device family. You can use it in production designs.

DSP IP Core Device Family Support

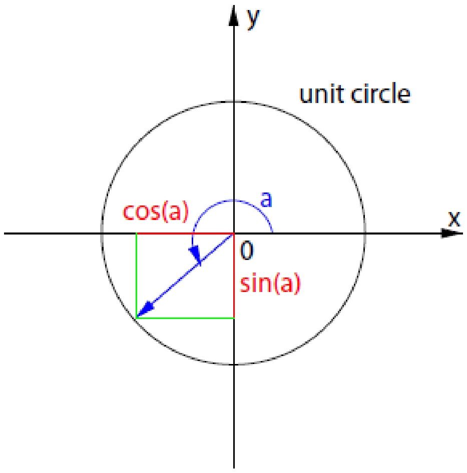
Device Family	**Support**
Arria® II GX	Final
Arria II GZ	Final
Arria V	Final
Intel® Arria 10	Final
Cyclone® IV	Final
Cyclone V	Final
Intel MAX® 10 FPGA	Final
Stratix® IV GT	Final
Stratix IV GX/E	Final
Stratix V	Final
Intel Stratix 10	Advance
Other device families	No support

ALTERA_CORDIC IP Core Functional Description

- SinCos Function on page 4
- Atan2 Function on page 5
- Vector Translate Function on page 5
- Vector Rotate Function on page 6

SinCos Function
Computes the sine and cosine of angle a.

SinCos Function

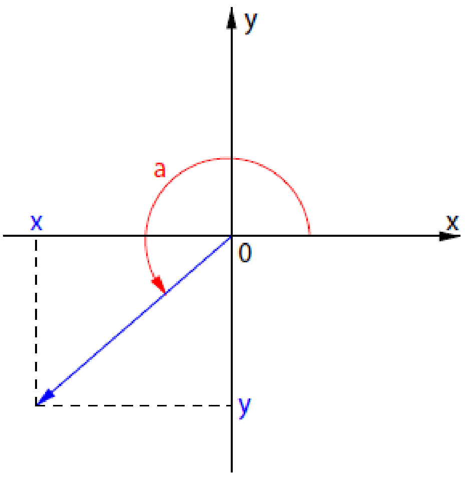


The function supports two configurations, depending on the sign attribute of a:

- If a is signed, the allowed input range is [-π,+π] and the output range for the sine and cosine is ∈[−1,1].
- If a is unsigned, the IP core restricts the input to [0,+π/2] and restricts the output range to [0,1].

Atan2 Function
Computes the function atan2(y, x) from inputs y and x.

Atan2 Function



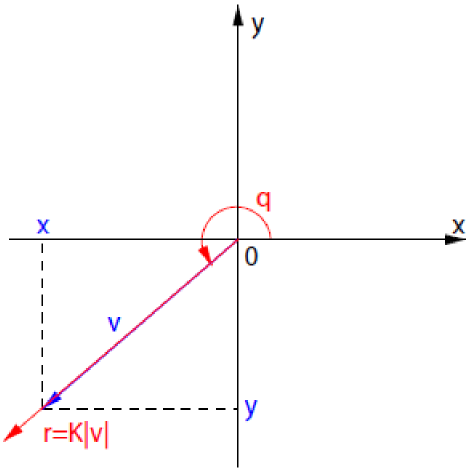
- If x and y are signed, the IP core determines the input range from the fixed-point formats.

- The output range is [-π,+π].

Vector Translate Function

The vector translate function is an extension of the atan2 function. It outputs the magnitude of the input vector and the angle a=atan2(y,x).

Vector Translate Function



The function takes inputs x and y and outputs a=atan2(y, x) and M = K(x2+y2)0.5. M is the magnitude of the input vector v=(x,y)T, scaled by a CORDIC specific constant that converges to 1.646760258121, which is transcendental, hence has no fixed value. The functions supports two configurations, depending on the sign attribute of x and y:

- If the inputs are signed, the formats give the allowed input range. In this configuration the output range for a is ∈ [−π,+π]. The output range for M depends on the input range of x and y, according with the magnitude formula.
- If the inputs are unsigned, the IP core restricts the output value for a [0,+π/2]. The magnitude value still depends on the formula.

Vector Rotate Function

The vector rotate function takes a vector v= (x,y)T given by the two coordinates x and y and an angle a. The function produces a similarity rotation of vector v by the angle a to produce the vector v0=(x0,y0)T.

Vector Rotate Function

The rotation is a similarity rotation because the magnitude of the produced vector v0 is scaled up by the CORDIC specific constant K(˜1.646760258121). The equations of the coordinates for vector v0 are:

- x0 = K(xcos(a)−ysin(a))
- y0 = K(xsin(a)+ ycos(a))

If you set the sign attribute to true for the x,y inputs for the function, the IP core restricts their range to [−1,1]. You provide the number of fractional bits. The input angle a is allowed in the range [−π,+π], and has the same number of fractional bits as the other inputs. You provide the output fractional bits and the total width of the output is w=wF+3, signed. For unsigned inputs x,y, the IP core restricts the range to [0,1], the angle a to [0,π].

ALTERA_CORDIC IP Core Parameters

SinCos Parameters

Parameter	Values	Description
Input data widths		
Fraction F	1 to 64	Number of fraction bits.
Width w	Derived	Width of fixed-point data.
Sign	signed or unsigned	The sign of the fixed-point data.
Output data widths		
Fraction	1 to 64, where *F*OUT ≤ *F*IN	Number of fraction bits.
Width	Derived	Width of fixed-point data.
Sign	Derived	The sign of the fixed-point data.
Generate enable port	On or off	Turn on for enable signal.

Atan2 Parameters

Parameter	Values	Description
Input data widths		
Fraction	1 to 64	Number of fraction bits.
Width	3 to 64	Width of fixed-point data.
Sign	signed or unsigned	The sign of the fixed-point data.
Output data widths		
Fraction		Number of fraction bits.
Width	Derived	Width of fixed-point data.
Sign	Derived	The sign of the fixed-point data.
Generate enable port	On or off	Turn on for enable signal.
LUT Size Optimization		Turn on to move some of the typical CORDIC operations into look up tables to reduce implementation cost.
Manually Specify LUT Size		Turn on to input the LUT size. Larger values (9-11) enable mapping some computations to memory blocks Only when **LUT Size Optimization** is on..

Vector Translate Parameters

Parameter	**Values**	**Description**
Input data widths		
Fraction	1 to 64	Number of fraction bits.
Width	Signed: 4 to 64; unsigned: F to 65	Width of fixed-point data.
Sign	signed or unsigned	The sign of the fixed-point data
Output data widths		
Fraction	1 to 64	Number of fraction bits.
Width	Derived	Width of fixed-point data.
Sgn	Derived	The sign of the fixed-point data
Generate enable port	On or off	Turn on for enable signal.
Scale factor compensation	On or off	For vector translate, a CORDIC specific constant that converges to 1.6467602 … scales the magnitude of the vector (x2+y2)0.5 so that the value for the magnitude, *M*, is M = K(x2+y2)0.5. The format of the output depends on the input format. The largest output value occurs when both the inputs are equal to the maximum representable input value, *j*. In this context: *M* = *K*(*j*2+*j*2)0.5 = *K*(2*j*2)0.5 = *K*20.5(*j*2)0.5 =*K* 20.5*j* ~2.32*j* Therefore, two extra bits left of the MSB of *j* are required to ensure *M* is representable. If scale factor compensation is selected, *M* becomes: *M* = j0.5 ~ 1.41 j One extra bit is sufficient for representing the range of *M*. Scale factor compensation affects the total width of the output.

continued…

Vector Rotate Parameters

Parameter	Values	Description
Input data widths		
X,Y inputs		
Fraction	1 to 64	Number of fraction bits.
Width	Derived	Width of fixed-point data.
Sign	signed or unsigned	The sign of the fixed-point data.
Angle input		
Fraction	Derived	–
Width	Derived	–
Sign	Derived	–
Output data widths		
Fraction	1 to 64	Number of fraction bits.
Width	Derived	Width of fixed-point data.

Parameter	Values	Description
Sign	Derived	The sign of the fixed-point data
Generate enable port	On or off	Turn on for enable signal.
Scale factor compensation		Turn on to compensate the CORDIC-specific constant on the magnitude output . For both signed and unsigned inputs, turning on decreases by 1 the weight of the magnitude for x0 and y0. The outputs belong to the interval [-20.5, +20.5]K. Under default settings, the output interval will therefore be [-20.5K , +20.5K] (with K~1.6467602…), or ~[-2.32, +2.32]. Representing the values in this interval requires 3 bits left of the binary point, one of which is for the sign. When you turn on **Scale factor compensation**, the output interval becomes [-20.5, +20.5] or ~[-1.41, 1.41], which requires two bits left of the binary point, one of which is for the sign. Scale factor compensation affects the total width of the output.

continued…

ALTERA_CORDIC IP Core Signals

Common Signals

Name	Type	Description
clk	Input	Clock.
en	Input	Enable. Only available when you turn on **Generate an enable port**.
areset	Input	Reset.

Sin Cos Function Signals

Name	Type	Configuration	Range	Description
a	Input	Signed input	$[-\pi,+\pi]$	Specifies the number of fractional bits (F_{IN}). The total width of this input is F_{IN}+3.Two extra bits are for the range (representing π) and one bit for the sign. Provide the input in two’s complement form.
		Unsigned input	$[0,+\pi/2]$	Specifies the number of fractional bits (F_{IN}). The total width of this input is $w_{IN}=F_{IN}+1$. The one extra bit accounts for the range (required to represent π/2).
s, c	Output	Signed input	[−1,1]	Computes sin(a) and cos(a) on a user-specified output fraction width(F). The output has width $w_{OUT}= F_{OUT}+2$ and is signed.
		Unsigned input	[0,1]	Computes sin(a) and cos(a) on a user-specified output fraction width(F_{OUT}). The output has the width $w_{OUT}= F_{OUT}+1$ and is unsigned.

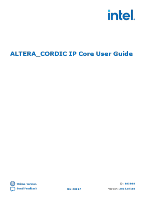
Atan2 Function Signals

Name	Type	Configur ati on	Range	Details
x, y	Input	Signed in put	Given by *w*, *F*	Specifies the total width (*w*) and number fractional bits (*F*) of the input. Provide the inputs in two's complement form.
		Unsigned input		Specifies the total width (*w*) and number fractional bits (*F*) of the input.
a	Ouput	Signed in put	$[-\pi,+\pi]$	Computes atan2(y,x) on a user-specified output fraction width (*F*). The output has the width $w_{OUT}= F_{OUT}+2$ and is signed.
		Unsigned input	$[0,+\pi/2]$	Computes atan2(y,x) on output fraction width (F_{OUT}). The output format has the width $w_{OUT} = F_{OUT}+2$ and is signed. However, the output value is unsigned.

Name	**Direction**	**Configuration**	**Range**	**Details**
x, *y*	Input	Signed input	Given by *w*, *F*	Specifies the total width (*w*) and number fractional bits (*F*) of the input. Provide the inputs in two’s complement form.
q	Output		[−*π*,+*π*]	Computes atan2(y,x) on a user-specified output fraction width *F*q. The output has the width *w*q=*F*q+3 and is signed.
r			Given by *w*, *F*	Computes *K*(x2+y2)0.5. The total width of the output is *wr*=*F*q+3, or *wr*=*F*q+2 with scale factor compensation.
				The number of meaningful bits depends on the number of iterations which depends on *F*q. The format of the output depends on the input format.
				MSB(M_{OUT})=MSB_{IN}+2, or MSB(M_{OUT})=MSB_{IN}+1 with scale factor compensation
x, y	Input	Unsigned input	Given by *w*,*F*	Specifies the total width (*w*) and number fractional bits (*F*) of the input.
q	Output		[0,+*π*/2]	Computes atan2(y,x) on an output fraction width *F*q. The output has the width *w*q=*F*q+2 and is signed.
r			Given by *w*,*F*	Computes *K(*x2+y2)0.5. The total width of the output is *wr*=*F*q+3, or *wr*=*F*q+2 with scale factor compensation.
				MSB(M_{OUT})=MSB_{IN}+2, or MSB(M_{OUT})=MSB_{IN}+1 with scale factor compensation.

Name	**Direction**	**Configuration**	**Range**	**Details**
x, y	Input	Signed input	[−1,1]	Specifies the fraction width (*F*), total number of bits is $w = F+2$. Provide the inputs in two’s complement form.
		Unsigned input	[0,1]	Specifies the fraction width (*F*), total number of bits is $w = F+1$.
a	Input	Signed input	[−*π*,+*π*]	Number of fractional bits is *F* (provided previously for x and y), total width is $w_a = F+3$.
		Unsigned input	[0,+*π*]	Number of fractional bits is *F* (provided previously for x and y), total width is $w_a = F+2$.
x0, y0	Output	Signed input	[−20.5,+20. 5]*K*	Number of fractional bits F_{OUT}, where $w_{OUT} = F_{OUT}+3$ or $w_{OUT} =$ $F_{OUT}+2$ with scale factor reduction.
		Unsigned input		

Documents / Resources

 intel ALTERA_CORDIC IP Core User Guide Online Version Send Feedback UG-20017 ID: 683808 Version: 2017.05.08	**intel ALTERA_CORDIC IP Core** [pdf] User Guide ALTERA_CORDIC IP Core, ALTERA_, CORDIC IP Core, IP Core

References

- intel **1. ALTERA_CORDIC IP Core User Guide**
- intel **Intel ISO 9001:2015 Registrations**